



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

PRŮMYSL 4.0 A DIGITÁLNÍ DVOJČE

INDUSTRY 4.0 AND THE DIGITAL TWIN

HABILITAČNÍ PRÁCE

HABILITATION THESIS

AUTOR PRÁCE

AUTHOR

Ing. Václav Kaczmarczyk, Ph.D.

BRNO 2023

Abstrakt

Tato práce si klade za cíl poskytnout v ucelené formě informace, týkající čtvrté průmyslové revoluce se zaměřením na problematiku digitálních dvojčat. Práce je dělena do tří oblastí. První, kterou je možné nazvat Digitálně podporovaná výroba popisuje problematiku teorie životního cyklu produktů ve výrobním podniku a možnosti současných softwarových systémů pro řízení životního cyklu. Data z těchto systémů tvoří základ kompletní digitální reprezentace produktů, výrobních strojů i celých výrobních linek a v dnešní době slouží jako podklad k vytváření různých typů modelů a následným simulacím. Pokud jsou tyto modely v průběhu životního cyklu aktualizovány s využitím skutečně měřených dat a jejich výstupy jsou využity pro změnu parametrů produktů, stávají se digitálními dvojčaty a pracují s fyzickými dvojčaty v rámci kyber-fyzického systému. Koncept řízení životního cyklu produktů a kompletního digitálního popisu je jedním z kroků vedoucích k transformaci, označované jako Průmysl 4.0. V následující části jsou představeny základní koncepty Průmyslu 4.0 a jsou popsány rozdíly mezi tradiční průmyslovou výrobou a moderními koncepty po transformaci z hlediska podnikové hierarchie, horizontální integrace napříč hodnotovým řetězcem a integrace inženýrských procesů. Jedním z důležitých konceptů Průmyslu 4.0 je Asset Administration Shell, který je definován v pasivní nebo aktivní formě. Obě tyto formy jsou detailně popsány spolu s modelovacími nástroji pro jejich vytvoření a nejčastěji používanými komunikačními protokoly pro jejich nasazení. Vzhledem k tomu, že aktivní AAS ve výrobních zařízeních je úzce spjato s problematikou plánování výroby, je třetí část této práce věnována popisu metod a algoritmů pro sestavení a výpočet výrobních plánů, a to jak v tradiční centralizované podobě, tak i v konceptu propojených kyber-fyzických systémů. V této části popsáno rozšíření vyjednávacího protokolu CNP o heuristiku umožňující lepší plnění požadavků Just-in-time výroby. Všechny tři oblasti, do kterých míří tato práce, sjednocuje testbed Průmyslu 4.0, který se nachází v laboratořích Ústavu automatizace a měřicí techniky Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně. V poslední části práce jsou detailně popsány možnosti využití tohoto testbedu při aplikaci výsledků vědeckého výzkumu do praxe a ve výuce témat souvisejících s Průmyslem 4.0.

Abstract

This thesis aims to provide information in a comprehensive form concerning the fourth industrial revolution with a focus on the issue of digital twins. The work is divided into three areas. The first, which can be called Digitally Supported Manufacturing, describes the issues of product life cycle theory in a manufacturing company and the capabilities of current software systems for life cycle management. The data from these systems form the basis of a complete digital representation of products, production machines and entire production lines and nowadays serve as the basis for the creation of various types of models and subsequent simulations. When these models are updated during the life cycle using actual measured data and their outputs are used to change product parameters, they become digital twins and work with physical twins within the cyber-physical system. The concept of product lifecycle management and complete digital description is one of the steps leading to the transformation referred to as Industry 4.0. In the following section, the basic concepts of Industry 4.0 are introduced and the differences between traditional industrial manufacturing and modern post-transformation concepts are described in terms of corporate hierarchy, horizontal integration across the value chain and engineering process integration. One of the important concepts of Industry 4.0 is the Asset Administration Shell, which is defined in passive or active form. Both of these forms are described in detail, along with the modeling tools for creating them and the most commonly used communication protocols for deploying them. Since active AAS in manufacturing facilities is closely related to the issue of production scheduling, the third part of this thesis is devoted to the description of methods and algorithms for the construction and calculation of production schedules, both in the traditional centralized form and in the concept of interconnected cyber-physical systems. This section describes the extension of the CNP negotiation protocol with heuristics to enable better fulfillment of just-in-time production requirements. All three areas addressed in this thesis are unified by the Industry 4.0 testbed located in the laboratories of the Institute of Automation and Measurement Technology, Faculty of Electrical Engineering and Communication Technologies, Brno University of Technology. The last part of the thesis describes in detail the possibilities of using this testbed in the application of the results of scientific research into practice and in the teaching of topics related to Industry 4.0.

Bibliografická citace

KACZMARCZYK, Václav. *Průmysl 4.0 a digitální dvojče*. Brno, 2023. Habilitační práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky.

Prohlášení autora

„Prohlašuji, že jsem svou habilitační práci na téma *Průmysl 4.0 a digitální dvojče* vypracoval samostatně 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 habilitační práce dále prohlašuji, že v souvislosti s vytvořením této habilitační 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 trestně právních důsledků vyplývajících z ustanovení druhé části, hlavy VI díl 4 trestního zákoníku č. 40/2009 Sb.“

V Brně dne:

Podpis:

Obsah

1	Průmyslové revoluce	2
1.1	Od vzniku manufaktur k internetu	2
1.2	(R)evoluce číslo čtyři	5
1.3	Současný stav	7
1.4	Holistický pohled na průmysl	8
1.4.1	Ohlédnutí pamětníka	9
1.4.2	Motivační příklad 1 - Aditivní výroba	10
1.4.3	Motivační příklad 2 - Pokročilé výrobní stroje	11
1.5	Struktura této práce	12
2	Digitálně podporovaná výroba	16
2.1	Životní cyklus výrobku	16
2.1.1	Fáze vývoje	17
2.1.2	Fáze zavádění	18
2.1.3	Fáze růstu zájmu	19
2.1.4	Fáze zralosti	19
2.1.5	Fáze útlumu zájmu	20
2.1.6	Výjimky při modelování S-křivkou	20
2.2	Správa životního cyklu výrobku	20
2.2.1	Možnosti ovlivnění životního cyklu	21
2.2.2	Softwarové systémy pro správu životního cyklu	22
2.2.3	Složitost výrobku a PLM	23
2.3	Software pro správu životního cyklu	24
2.3.1	Archiv pro uložení modelů	25
2.3.2	Způsob ukládání podkladů v Teamcenteru	26
2.3.3	Workflow procesy	28
2.4	Od fyzického dvojčete po to digitální	33
2.4.1	Nízká úroveň integrace – digitální model	34
2.4.2	Digitální stín	34
2.4.3	Digitální dvojče	35
2.5	Digitální dvojče výrobku	37
2.5.1	Případová studie - Maserati	38
2.6	Digitální dvojče výrobního stroje	39
2.6.1	Tvorba 3D modelu	40

2.6.2	Aplikace dynamických vlastností	40
2.6.3	Implementace fyzikální simulace	41
2.6.4	Případová studie - Škoda Machine Tool	42
2.7	Digitální dvojče výrobní linky	43
2.7.1	Virtuální obraz reálné výroby	47
2.7.2	Plánování výrobního procesu	47
2.7.3	Detailní simulace výrobního procesu	49
2.7.4	Případová studie - Škoda Auto	51
2.8	Digitální dvojče výrobního procesu	52
2.9	Digitální dvojče pro jiné typy výroby	54
2.10	Digitální dvojče mimo průmysl	56
2.11	Shrnutí kapitoly	59
3	Průmysl 4.0?	60
3.1	Hlavní aspekty průmyslové integrace	63
3.1.1	Vertikální propojení 3.0	64
3.1.2	Horizontální propojení 3.0	68
3.1.3	Propojení inženýrských procesů 3.0	69
3.2	Přechod k Průmyslu 4.0	69
3.2.1	Spouštěče Průmyslu 4.0	70
3.2.2	Kyber-fyzické systémy	74
3.2.3	Vertikální integrace v Průmyslu 4.0	77
3.2.4	Horizontální integrace 4.0	79
3.2.5	Integrace inženýrských procesů 4.0	80
3.3	Shrnutí kapitoly	81
4	Digitální dvojče z pohledu průmyslu 4.0	82
4.1	Model RAMI	82
4.1.1	Integrační osa RAMI 4.0	84
4.1.2	Osa hodnotového toku RAMI 4.0	85
4.1.3	Osa hierarchie RAMI 4.0	86
4.2	Model <i>Industry 4.0 component</i>	87
4.2.1	Transformace objektů na I4.0 komponenty	89
4.2.2	Způsoby integrace I4.0 komponenty	90
4.2.3	Speciální případy mapování dat	92
4.2.4	Pasivní AAS	93
4.2.5	Aktivní AAS	93
4.3	Základní struktura AAS	95
4.3.1	Meta Model AAS	97
4.3.2	Identifikace komponent	100
4.4	Možnosti implementace AAS	102
4.4.1	AASX Package Explorer	102
4.4.2	Další nástroje pro modelování AAS	103
4.5	Jazyk aktivních komponent	104

4.6	Prostředky komunikace mezi komponentami	105
4.6.1	MQTT	106
4.6.2	OPC	108
4.6.3	Srovnání MQTT a OPC UA	114
4.7	Shrnutí kapitoly	115
5	Plánování výroby	116
5.1	Úrovně plánování v podniku	116
5.2	Potřeby výrobních firem v oblasti plánování	120
5.3	Metody pro plánování výrobních kapacit	121
5.4	Matematické modely plánovacích problémů	124
5.5	Deterministické metody rozvrhování ve výrobě	131
5.5.1	Lineární programování	132
5.5.2	Lineární modely pro plánování výroby	132
5.5.3	Řešení metodami Branch and Bound	135
5.6	Stochastické metody pro plánování výroby	136
5.6.1	Konstruktivní metody	136
5.6.2	Lokální prohledávání	139
5.6.3	Meta-heuristické algoritmy	140
5.7	Kritéria výběru vhodné matematické metody	140
5.8	Shrnutí kapitoly	142
6	Inteligentní výrobní systémy	143
6.1	Typy řídicích architektur výrobních systémů	143
6.1.1	Centralizovaná architektura	144
6.1.2	Hierarchická architektura	145
6.1.3	Upravená hierarchická architektura	145
6.1.4	Heterarchická architektura	146
6.2	Definice inteligentního výrobního systému	147
6.3	IMS založené na agentním přístupu	148
6.4	Potenciál a využitelnost MAS v průmyslovém prostředí	152
6.5	Související technologie přístupu Průmyslu 4.0	154
6.6	Referenční architektura CPPS	155
6.7	Shrnutí kapitoly	157
7	Algoritmy pro distribuovanou organizaci výroby	158
7.1	Model rozvrhování úloh v CPPS	159
7.1.1	Robustnost rozvrhování	159
7.1.2	Interakce mezi agenty	160
7.1.3	Operativní plánování	167
7.2	Návrh algoritmu organizace výroby CPPS	171
7.2.1	Model výrobního zdroje (Resource agent)	172
7.2.2	Plánování operací	173
7.2.3	Heuristika vkládání úloh do rozvrhů RA	176
7.2.4	Možná rozšíření heuristiky vkládání	178

7.3	Simulace MAS řízení výroby	178
7.3.1	Validace simulace řídicí logiky	180
7.4	Shrnutí kapitoly	182
8	Testbed Průmyslu 4.0	183
8.1	Konstrukce testbedu	185
8.2	Softwarové vybavení testbedu	187
8.3	Typ výroby v chytré továrně	187
8.4	Centralizovaně řízená výroba	188
8.4.1	Distribuovaně realizovaná výroba	190
8.4.2	Virtuální zprovoznění	192
8.4.3	Rozšířená realita	193
8.5	Shrnutí kapitoly	194
9	Závěr	196
A	Popis implementace simulačního prostředí	199
	Literatura	204

Předmluva

Proslýchá se, že žijeme v období 4. průmyslové revoluce. Někteří z nás se s tímto pojmem setkávají takřka denně, ale většina lidí vůbec netuší, že se nějaká průmyslová revoluce odehrává. Není to divné? Neměla by se revoluce, která ze své podstaty mění náhle a převratně celou společnost, dotknout každého z nás?

Tato práce si klade za cíl poskytnout přehledně informace týkající se témat souvisejících právě s poslední průmyslovou revolucí. Nejedná se určitě o kompletní, hluboký a vyčerpávající popis všech rysů, konceptů či technologií. Ten by byl násobně obsáhlejší a vyžadoval by spolupráci týmu autorů. Práce se zaměřuje na popis hlavních aspektů *Digitálních dvojčat* napříč celým životním cyklem vyráběných produktů, či výrobních zařízení.

Motivací pro napsání této práce byla neexistence uceleného materiálu, který by bylo možné použít jako zdroj informací, potřebný při výuce předmětu *Průmysl 4.0* na Ústavu automatizace a měřicí techniky, Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně. Částečně je na ni možno nahlížet jako na výukový materiál, ale většinou obsahuje zejména informace obecnější povahy, které je pro účely výuky nutno doplňovat řadou příkladů a praktických ukázek. Tyto doplňkové materiály jsou v současné době studentům prezentovány na přednáškách, praktický výcvik je prováděn v počítačových cvičeních v uvedeném předmětu.

Na tomto místě chci poděkovat svému učiteli a později kolegovi, Ing. Janu Páskovi, CSc., bez jehož podpory by tato práce nevznikla. Velký dík za nekonečnou trpělivost patří také mé rodině - manželce Zuzaně, dětem Jakubovi a Adéle.

Kapitola 1

Průmyslové revoluce

1.1 Od vzniku manufaktur k internetu

Po mnoho staletí lidé vykonávali různá řemesla individuálně ve svých domovech nebo v jejich blízkosti. Vzhledem k tomu, že některé zeměpisné oblasti jsou vhodnější pro vykonávání určitých řemesel, než jiné (např. sklářství v zalesněných oblastech s dostatkem materiálu k vytápění a místech bohatých na vstupní surovinu), začali se konkrétní řemeslníci v těchto oblastech koncentrovat. Aby byli schopni uspět v relativně velké konkurenci, byli navázáni na obchodníky, kteří jejich zboží vykupovali a dodávali mnohem širšímu okruhu zákazníků, než by výrobci sami dokázali.

Postupně začaly vznikat manufaktury - předchůdci továren, ve kterých byli řemeslníci zaměstnáváni a pracovali společně. Způsob práce každého jednotlivce v manufaktuře však zůstal stejný. Jednalo se o *instrumentální*¹ ruční práci bez asistence strojů.

Na sklonku 18. století vypukla v Británii první průmyslová revoluce. Ruční práci v textilním průmyslu poprvé začaly ve větším měřítku přebírat stroje, které vznikly na základě převratných vynálezů i pomalého kontinuálního vylepšování mechanických zařízení. Jedním z nich byl tkalcovský stav. Úkoly, které předtím doma namáhavě zpracovávaly ručními nástroji stovky tkalců, se přenesly do jedné přádelny vybavené těmito stavy a zrodila se továrna.

Masivní nástup *mechanizace*² do průmyslu a zemědělství vedl k prudkému růstu produktivity. Ta byla následována nevídaným růstem populace, který přispěl k rozsáhlým společenským, politickým i kulturním změnám. Významem je tento převrat srovnatelný s neolitickou revolucí, která způsobila proměnu společnosti od lovců a sběračů ke společnosti zemědělské.

První průmyslová revoluce je úzce spjata s masivním rozšířením parního pohonu. Přestože v jednoduché formě sílu páry objevil již kolem roku 50 n.l. Herón Alexandrijský

¹instrumentace - pracovní proces je vybaven ručními nástroji

²mechanizace - fyzická lidská práce je nahrazována činností strojů



Obrázek 1.1: Montážní linka v závodech H. Forda (1913) [28]

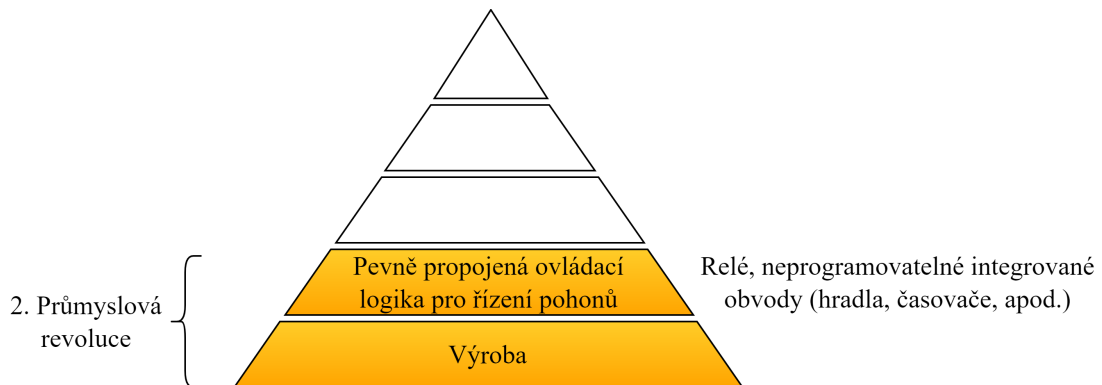
(Aeolopile), trvalo dalších více než 1700 let, než tehdejší parní pohon dokázal roku 1769 James Watt zkombinovat s jednoduchým odstředivým zařízením - regulátorem a otevřela se cesta k průmyslovému využití parní energie.

Slovo *revoluce* na jednu stranu podněcuje zvědavost, na druhou stranu vzbuzuje emoce a šokuje. Proto se i každá další významná změna, ke které v průmyslu došlo, začala postupně nazývat revolucí. Druhá, tzv. vědecko-technická revoluce, která přišla asi o 100 let později, přinesla elektrifikaci, motorizaci a rozšíření nových materiálů do výroby. Klíčovou roli sehrály vynálezy Thomase Alvy Edisona (žárovka), Nicolý Tesly (elektrický transformátor) a Gottlieba Daimlera (spalovací motor).

Nezbytným činitelem se stal také začátek těžby a zpracování ropy a zemního plynu. Od počátku 20. století, kdy Henry Ford vyvinul pohyblivou montážní linku s dělbou práce, hovoříme o věku masové výroby. Díky prvním dvěma průmyslovým revolucím lidé zbohatli a začali se mnohem více koncentrovat ve městech.

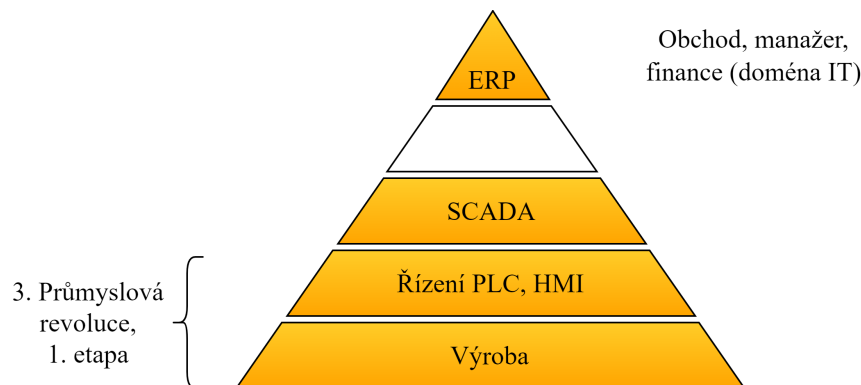
Roztočená kola pohonů výrobních linek, stejně tak jako vynálezy mechanických spínacích prvků a regulátorů vlastně znamenaly elektrotechnickou revoluci. Vezměme si známou pyramidu popisující hierarchicky jednotlivé vrstvy zařízení a systémů ve výrobních podnicích. Pokud bychom chtěli technologie druhé průmyslové revoluce někam začlenit, bylo by to do spodních dvou úrovní této pyramidy (obr. 1.2). Zbývající

části pyramidy zatím zůstaly neobsazeny.



Obrázek 1.2: Elektrotechnika v období druhé průmyslové revoluce

Třetí průmyslová revoluce souvisí s nástupem výpočetní techniky, která umožnila *automatizaci*³ řady odvětví lidské činnosti a urychlila technický vývoj. Za její přesný počátek bývá uváděn rok 1969, kdy byl vyroben první programovatelný logický automat, avšak její datování je sporné. Stejně tak, jako byl přechod od uhlí a páry k elektřině poměrně spojitý a logický, i přechod od mechanismů k automatům byl spíše výsledkem přirozeného vývoje (*evoluce*) než pravou revolucí [26].



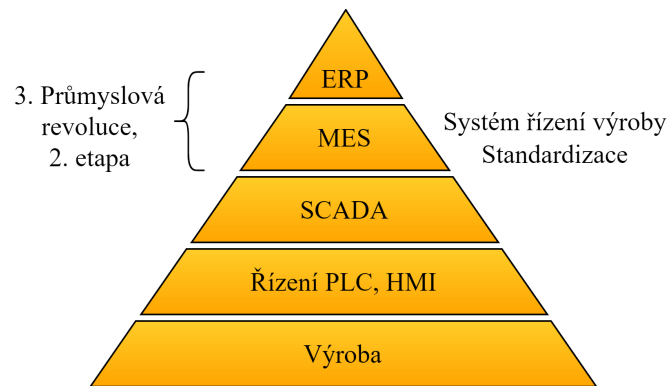
Obrázek 1.3: Informační a řídicí technologie 1. etapy 3. průmyslové revoluce

Pohledem na automatizační pyramidu na obr. 1.3 zjistíme, že třetí průmyslová revoluce již měla mnohem širší záběr. Díky pozdějšímu rychlému vývoji v oblasti informačních technologií mohly do výrobních podniků pronikat průmyslové deriváty (především programovatelné logické automaty). Dělo se to však velmi pomalu a vývoj byl vždy oproti PC opožděn. Tato éra se do technické historie zapsala jako *věk informací* a následně *věk internetu*.

Na začátku 2. tisíciletí se objevily *systemy řízení výroby* (Manufacturing Execution Systems, MES). Tyto systémy přinesly možnost realizovat skutečně automatickou výměnu informací mezi manažerskou a výrobní úrovní v podnicích. Jednotlivé moduly těchto systémů spravují všechny aspekty výrobních podniků současnosti a umožňují

³automatizace - duševní a řídicí práce je nahrazována činností strojů

skutečně automatizovat průmyslovou výrobu (Obr. 1.4). Z dnešního pohledu můžeme říci, že proběhla 2. etapa třetí průmyslové revoluce .



Obrázek 1.4: Informační a řídicí technologie dnešního výrobního podniku

Současně s tímto převratným propojením pyramidy se začal připojovat tlak globalizace. Množství informací ve výrobní vrstvě enormně rostlo a stále roste. S tím přicházejí i nové sofistikované technologie na zpracování a využívání dat a informací. Nejedná se však o žádné převratné inovace, neboť tyto technologie byly víceméně odkoukané od IT z obchodní podnikové domény.

1.2 (R)evoluce číslo čtyři

První krok ke čtvrté průmyslové revoluci byl proveden již v roce 1962, kdy vznikl projekt počítačového výzkumu agentury ARPA. Agentura dostala v souvislosti se studenou válkou v USA zadání, aby vyvinula komunikační síť pro počítače s decentralizovaným řízením. V roce 1969, kdy byl tento projekt dokončen, uvedla agentura první experimentální síť ARPANET. V roce 1987 se rozšířil pojem Internet a během 90. let docházelo k jeho rozšiřování a komercializaci. Od přelomu tisíciletí pak sledujeme extrémní nárůst uživatelů internetu, který v dnešní době již dosahuje řádu miliard. Tím to ale nekončí. K síti se připojují kromě lidí také stroje a věci obecně. Reálné a virtuální světy se začínají prolínat a do hry vstupují tzv. kyber-fyzické systémy.

O čtvrté průmyslové revoluci se začalo hovořit až teprve mnohem později - na počátku minulé dekády a měla trvat minimálně dalších 10 – 30 let⁴. Čtvrtá průmyslová revoluce je charakterizována právě masovým rozšířením internetu a jeho průnikem do doslova všech oblastí lidské činnosti. Stejně jako u předchozích tří, se opět nejedná o revoluci ve smyslu *zítra to bude úplně jinak*. Je to revoluce postupnou evolucí, kdy se nejen v průmyslu, ale i ve službách postupně přechází od užívání izolovaných počítačů a strojů k integrovaným řešením.

Tato změna přináší nové možnosti a změny v charakteru tradičních výrobních i spotřebních procesů. Přestože čtvrtá průmyslová revoluce není pouze o digitalizaci,

⁴Bude však jistojistě *formálně* mnohem rychleji nahrazena dalšími revolucemi, které budou dále rozvíjet nové koncepty a možnosti.

právě digitalizace je základní podmínkou pro realizaci systémových kybernetických principů a realizaci inteligentních výrobních systémů a služeb. Nebude se však jednat o prostou digitalizaci, tedy připojení všeho do internetu! Budou také využívány moderní metody kybernetiky, umělé inteligence, dojde k propojení virtuálních zařízení, která budou spolupracovat s fyzickým světem.

Největší změna se bude týkat produkce výrobků v rámci továren, která bude transformována ze samostatných automatizovaných výrobních jednotek na plně integrovaná (samozřejmě automatizovaná) a průběžně optimalizovaná výrobní pracoviště. Propojením fyzických výrobních zařízení a datově řízených modelů (digitálních dvojčat těchto zařízení) do kyber-fyzických systémů vzniknou sítě s mnoha novými možnostmi. Tyto sítě, schopné autonomní výměny informací, okamžitých reakcí na změny momentálních podmínek či vzájemné diagnostiky, budou základním stavebním prvkem inteligentních továren.

V inteligentních továrnách budou výrobní stroje, senzory, informační systémy, ale i produkty samotné vzájemně propojeny s nadřazenými systémy v rámci továrny. Výrobní proces bude trvale optimalizován a bude schopen reagovat na nečekané změny způsobené poruchou některého výrobního zařízení, nebo dokonce zvýšením pravděpodobností vzniku poruchy v nejbližší době. Všechny systémy budou tedy moci v reálném čase pružně reagovat na okamžitou a měnící se poptávku po produktech, tedy na individuální požadavky zákazníků a umožní vyrábět produkty spolehlivě a efektivně. Fyzické prototypy budou z velké části nahrazeny virtuálními návrhy výrobků, výrobních prostředků a výrobních procesů, tedy digitálními dvojčaty. Uvedení výrobních procesů do provozu bude předcházet jejich tak zvané virtuální zprovoznění, tedy celková simulace, ověření činnosti a optimalizace.

Další propojení vzniknou v horizontálním směru napříč celým dodavatelským řetězcem. Postupně vzniknou také nové obchodní modely spolupráce mezi firmami. Díky efektivnímu propojení obchodních řetězců, založeném na nově vznikajících standardech, bude možné rychle nahradit chybějící či nefunkční článek již v okamžiku hrozícího problému.

Inteligentní produkty, které budou v takových továrnách vyráběny, nejenže budou během výroby jednoznačně identifikovatelné, lokalizovatelné, a tedy i dohledatelné, ale budou také udržovat svůj aktuální stav a celou svou výrobní historii. Na základě měnící se dostupnosti vstupních surovin či polotovarů u dodavatelů a měnícího se zájmu zákazníků bude možné v reálném čase pružně reagovat změnou množství a charakteru vyráběných produktů. Bude možné reagovat na individuální potřeby zákazníků, a tedy vyrobit pro každého z nich požadovaný produkt, přitom ale efektivně a se zachováním co nejvyšší konkurenceschopnosti. Vzhledem k tomu, že proces výroby bude sledován a optimalizován, bude možné reagovat na nečekané změny způsobené např. poruchou některého z výrobních strojů. Taková interakce bude možná právě díky propojení prvků fyzického světa prostřednictvím tzv. internetu věcí.

Kyber-fyzické systémy budou prostřednictvím komunikačních rozhraní vzájemně

interagovat a analyzovat data, čímž budou předvídat případné chyby, či poruchy, a v reálném čase se pomocí rekonfigurace svých vlastností přizpůsobovat měnícím se podmínkám. Stále větší roli bude hrát družicová navigace, a to především v oblasti autonomní dopravy a logistiky.

Po ukončení výrobního procesu konkrétního výrobku bude formou pokročilých služeb zajišťován po celou dobu jeho životnosti servis, případně činnosti, které souvisejí s vylepšováním jeho parametrů a užitečných vlastností. Tyto služby bude možné poskytovat u komplikovanějších a technicky náročnějších celků díky monitoringu pomocí vzdáleného přístupu k zařízení. Další formou zprostředkování servisních služeb bude aktivní sebediagnostika produktu a automatická komunikace se servisním střediskem. Běžná údržba bude probíhat autonomně v režimu tzv. prediktivní údržby, kdy na základě sledování konkrétních aktuálních vlastností produktu bude možné provádět automatické korekce parametrů tak, aby bylo možné napravit nežádoucí chování výrobku bez zásahu člověka.

1.3 Současný stav

Pomyslná kola nové průmyslové revoluce roztočila v roce 2011 Německá iniciativa Industrie 4.0. Vzhledem k provázanosti české ekonomiky s tou německou je nutné, aby také Česká republika z této iniciativy vytěžila maximum. V rámci České republiky proto v roce 2016 vznikla iniciativa *Průmysl 4.0* [72]. Vlastní program s názvem *Industrie du Futur* spustila v květnu 2015 i francouzská vláda. Průmysl 4.0 je tedy čistě evropskou iniciativou. Neznamená to však, že by ve zbytku světa neprobíhaly paralelně totožné změny. V USA mají svou iniciativu *Smart Manufacturing Leadership Coalition*, jejíž hlavní rolí je transformace průmyslového sektoru ve vzájemně propojené, informacemi řízené prostředí, které umožňuje optimalizaci vlastních výrobních procesů i celého hodnotového řetězce. Čínská vláda kvůli postupnému mizení výhody levné místní pracovní síly spustila vlastní program pro zvýšení konkurenceschopnosti svého průmyslu s názvem *Made-in-China 2025*. Ten se z velké části inspirovuje německým Industrie 4.0 a pokrývá problémy standardů, procesů, energetické optimality či rozvoje lidských zdrojů v několika prioritních průmyslových segmentech. Podobné strategie prosazují také země jako Jižní Korea či Japonsko.

Vraťme se ale zpět do České republiky. Jak se zde Průmysl 4.0 týká každého z nás? Mnohokrát již bylo řečeno, že *roboti seberou lidem práci* podobně jako v minulosti *stroje braly lidem práci*. Ano, přesně tak, jako měly transformační procesy první, druhé i třetí průmyslové revoluce obrovský vliv na celou společnost⁵, očekává se, že i procesy Průmyslu 4.0 překročí úzké mantinely samotného průmyslu. Proto se také začíná objevovat vážná diskuse o důsledcích Průmyslu 4.0 pro zaměstnanost. Existují pesimistické studie předpovídající uvolňování 40 - 50 % současného počtu zaměstnanců v

⁵Tedy nejen na vlastní průmyslovou výrobu, ale v návaznosti také na vědu, zemědělství, transport, veřejnou správu, a další oblasti.

horizontu příštích 15 let. Lze předpokládat, že na trhu práce dojde k útlumu požadavků po některých profesích, některé dokonce zaniknou.

Proti uvedeným studiím ovšem zase stojí jiné odhady růstu produktivity práce, či zvýšení odbytu, jež se budou projevat v růstu obrátu firem, a tedy v poptávce po nových zaměstnancích. Vzhledem k charakteru změn přirozeně cítíme, že přibude práce zejména pro kvalifikované multidisciplinární pracovníky, ale vzkvétat budou i další obory a vzniknou nové pozice pro méně kvalifikované pracovní síly.

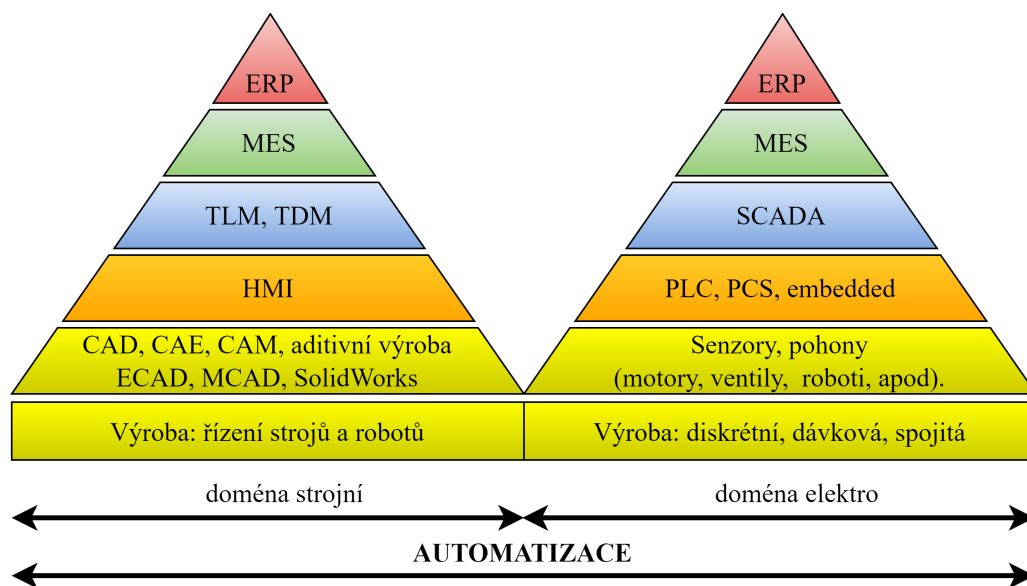
Lidé v produktivním věku se musí připravit na změnu. Ti, kteří svou kariéru začínali v době nástupu automatizace do průmyslu, kdy se z reléového řízení postupně přecházelo na PLC, jsou v dnešní době schopni přijmout další velkou změnu jen s těžší. Pro mladší generace je však nadcházející transformace zcela přirozená. Mladí lidé se zcela přirozeně nechtějí nechat svazovat a omezovat zažitými pořádky a dokáží své vnímání otevřít novým myšlenkám a konceptům, a ty postupně prosazovat a uvádět v život.

1.4 Holistický pohled na průmysl

Iniciativa Průmysl 4.0 přinesla kromě jiného spoustu nových termínů. Některé je nyní na místě objasnit. Základním pojmem problematiky je INDUSTRY – PRŮMYSL. Každý demokraticky zvolený prezident po volbách říká, že bude prezidentem všech, tedy i těch, co ho nevolili. Stejně tak i iniciátoři myšlenek Průmysl 4.0, chtějí-li uspět, musí získat důvěru a přízeň co největšího počtu průmyslových sektorů. Není to snadný úkol, protože průmysl má mnoho typů a odvětví a ty jsou natolik rozdílné, že vyžadují zcela odlišné přístupy k jejich řízení. Kdo se již trochu seznámil s původem a ideou vize Průmyslu 4.0, ví, že její kořeny jsou úzce spjaty s automobilovou výrobou, potažmo obecně s diskrétní výrobou. Existují ale ještě další sektory průmyslové výroby:

1. *diskrétní výroba,*
2. *dávková výroba,*
3. *spojitá výroba.*

Pro každý ze sektorů existují charakteristické typy výrobních procesů. Ty se mezi sebou rozlišují podle různého *objemu vyráběného množství či výrobků, nestejné složitosti,* nebo *unikátnosti vyráběných produktů,* apod. Spojité procesy se vyznačují stálou produkcí proudu výstupní suroviny s pečlivě udržovanými parametry - např. pitné vody či elektrické energie. Dávkové výrobní procesy na svém výstupu produkují ucelené dávky materiálů. Jako příklad je možné jmenovat pasterizaci vstupní suroviny při výrobě mléčných výrobků. Tyto procesy se již řadí do skupiny nespojitých, stejně jako diskrétní procesy, které produkují vzájemně oddělené jednotlivé výrobky, jejichž parametry se často kus od kusu podstatně liší.



Obrázek 1.5: Začlenění strojní domény (vlevo) do průmyslové automatizace (vpravo)

1.4.1 Ohlédnutí pamětníka

Celá problematika výrobních procesů a jejich řízení je na *Ústavu automatizace a měřicí techniky Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně* historicky hluboce zakořeněna, avšak s jednou výjimkou, kterou představuje číslicové řízení obráběcích strojů.⁶

„Výroba strojních součástí, jako jsou hřídele, ozubená kola a další, pro kterou jsou tyto obráběcí stroje nezbytné, tvoří základ diskretních výrobních procesů. Přesto je možné dle pamětníků vysledovat odklon od problematiky prakticky zaměřeného číslicového řízení obráběcích strojů již v 70. letech minulého století. Tehdy se značná pozornost začala soustřeďovat na zpětnovazební kybernetiku. Dnes již není zcela jasné, proč se to stalo. Jeden z možných důvodů je, že zabývat se řídicími algoritmy, které jen nastavují rozměry, nebo jejich přírůstky a ovládají pohony ve třech osách tenkrát nepředstavovalo zajímavé téma. Číslicové řízení strojů však zůstalo na fakultě strojního inženýrství a dnes je situace taková, že v tomto oboru jsou míle před námi. Byla to tenkrát chyba, která se projevila až po mnoha letech.“ [85].

Položme si nyní otázku, odkud myšlenka Průmyslu 4.0 přišla? Jako nejpravděpodobnější se jeví inspirace systémy CAD (*computer aided design*), CAE (*computer aided engineering*) a od číslicového řízení obráběcích a tvářecích strojů.

Byla to právě iniciativa Průmysl 4.0, která způsobila, že jsme „nahlédli přes rameno“ kolegům ze strojní fakulty a viděli, že i oni mají automatizační pyramidu, a že jejich pyramida má strukturu velmi podobnou té naší. Obě jsou uvedeny na obr. 1.5. Vrstvy, které si v obou pyramidách odpovídají jsou značeny stejnou barvou.

⁶Spoluautorem myšlenek prezentovaných v této kapitole je Ing. Jan Pásek, CSc., který na autora dlouhé roky působil coby pedagog a později kolega.

Jak je vidět z tohoto obrázku, metody SW algoritmů pro řízení CNC strojů a posléze robotů vychází z počítačové podpory návrhu (CAD), inženýringu (CAE), výroby (CAM) a dalších. HMI plní stejnou funkci jako u *elektro-automatizace*. Funkci SCADA, tj. rozsáhlejší serverová vizualizační uživatelská rozhraní u *elektro-automatizace*, zde přebírají systémy *Tool Lifecycle Management* a *Tool Date Management*.

Tato *pyramidální dvojčata* jsou jednou z výzev Průmyslu 4.0. Je to výzva, která říká: „přestaňme do nekonečna nastavovat PID regulátory a pojd' me dělat průmyslovou revoluci“. Neudělejme ale stejnou chybu dvakrát, jen tentokrát v opačném směru. Pojd' me přemýšlet nad Průmyslem 4.0 konzistentně pro všechny typy a sektory průmyslu!

Pojem Průmysl 4.0 se mnohdy váže s termínem *iniciativa* a je možné říct, že se jedná o slovo přiléhavé. Průmysl 4.0 je iniciativa, která má fungovat jako spouštěč k běhu na dlouhou trať. Iniciativa, která zahrnuje téměř vše, čeho bylo v současnosti v oblasti průmyslové automatizace dosaženo. Dotýká se všech typů průmyslových výrob, všech průmyslových sektorů, zařízení, systémů, služeb i pracovníků, kteří se v průmyslu vyskytují, ale také sociálních a environmentálních aspektů. Něco tak heterogenního pod jedním názvem snad odbornou průmyslovou veřejnost ještě nikdy nevyzvalo ke společné cestě. Je to skutečně spíše cesta, než cíl. Ten je sice zatím vágní a částečně i neznámý a otevřený. Je to vize, pro někoho až fantastická, ale nebyly ty největší dosažené technické úspěchy na začátku také pouze science fiction?

1.4.2 Motivační příklad 1 - Aditivní výroba

Trendem současné doby je vymýšlení a výroba stále složitějších výrobků v co nejkratší době, aby byl výrobek vyvinut co nejlevněji, ale také aby se vlastnosti výrobku maximálně přiblížily přání zákazníka. Pro splnění těchto cílů je nutné, aby výrobní podnik pracoval efektivně a organizovaně, a to jak uvnitř, tak i při spolupráci s dodavateli a odběrateli. To znamená, aby měl pod kontrolou jednak celý proces výroby, nakládání se surovinami, ale i zpracování výrobních a dalších dat.

S efektivitou výroby, a koneckonců i s CNC výrobou je velice úzce spjata tzv. *aditivní výroba*, tedy 3D tisk jako další pilíř a zdroj iniciativy Průmysl 4.0. V čem aditivní výroba spočívá? Představme si knihu, která je složena ze stránek. Kniha je výrobek, který můžeme uchopit. Jak ji ale dostaneme do digitální elektronické podoby, abychom s ní mohli manipulovat v počítačovém virtuálním světě? Jednoduše tak, že jednotlivé stránky naskenujeme a uložíme je dohromady do jednoho souboru. Jiný podobný příklad je počítačová tomografie, magnetická rezonance a jiné lékařské zobrazovací metody, kdy z lidského těla uděláme obrazovou knihu, ve které můžeme listovat. Posledním příkladem může být využití 3D skeneru pro pořízení prostorových dat nějakého objektu zájmu.

Co to zkusit naopak? Vztít stránky, listy materiálu a lepit je na sebe a tím vytvořit kompaktní výrobek. Ani to není nic nového. Vyrábí se tak mnoho výrobků od lyží, sportovních kánoí až po trupy letadel. Pro ně společné, že mají souvislé plochy, nemají

dutiny a nejsou tvarově složité. Vrátime-li se k příkladu knihy, víme, že i tam je uvedený postup polovičatý, že jsme sice zdigitalizovali stránky, ale ne jednotlivá písmenka. Když text napíšeme a vytiskneme, vytváříme každé písmeno z určitého počtu pixelů. Naskenovaná kniha je zdigitalizovaný výrobek. Princip aditivní výroby tkví v ukládání malého množství hmoty na přesné místo dané algoritmem návrhu. Vrstvu po vrstvě, až je výrobek kompletní.

Aditivní výroba, tedy 3D tisk, je proces spojování materiálu dle 3D digitálních dat, nejčastěji vrstva po vrstvě. 3D tisk se již osvědčil ve výrobě plastů. Nyní nastoupil svoji cestu ve zpracování kovů. To nejsou zrovna malé průmyslové sektory. Plasty a kovy ve zpracovatelském sektoru představují 25 % celkové produkce. Vezměme si nyní reálný příklad. V leteckém průmyslu patří palivové trysky k nejsložitějším částem výroby, a to z jednoho důvodu: jsou sestaveny z 20 různých součástí, které musí být samostatně vyrobeny, a pak pracně smontovány. Letecký průmysl teď používá 3D tisk, což jim umožní změnit těch 20 rozdílných součástí v jeden díl. Výsledek? Dramatické zjednodušení návrhu bez nutnosti použití spojovacího materiálu, snížení hmotnosti výrobku a nárůst produktivity 40 %, tedy 40 % více výrobků v tomto konkrétním případě.

3D tisk také nabízí větší variabilitu výrobků tím, že umožňuje vyrobit díly v customizovaných verzích výrobků. Technologie se využívá např. při výrobě protetických pomůcek zákazníkovi doslova na míru. 3D tisk také plní nezastupitelnou úlohu při vývoji zařízení, kdy umožňuje v rádech hodin po dokončení návrhu získat prototypové kusy výrobku a tím obrovsky urychlit vývoj zařízení.

Z hlediska obsluhy jsou navíc technologie 3D tisku nenáročné - časová dotace přípravy výroby je relativně nízká, příprava dat poloautomatická a samotná výroba je v podstatě autonomní.

1.4.3 Motivační příklad 2 - Pokročilé výrobní stroje

Pokročilé výrobní roboty můžeme dnes vidět v řadě průmyslových sektorů. Nejčastěji (jak jinak) opět v automobilovém průmyslu. Velikostně odpovídají člověku, spolupracují s člověkem a mohou být programováni tak, aby vykonávali složité nerutinní, neopakující se úlohy. V době startu 4. průmyslové revoluce bylo v továrnách automatizováno pouze 8 % úloh, přičemž šlo o méně složité a opakující se práce. Do roku 2025 předpokládá navýšení tohoto podílu na 25 %. Tzn., že v roce 2025 pokročilí roboti doplní pracovníky tak, že společně přinesou 20 % zvýšení produktivity, vyrobí se o 20 % více, aby se dosáhlo zvýšení růstu o 20 %.

Ovšem cílem nové průmyslové revoluce ve skutečnosti není produktivita, ale lepší a chytřejší produkty. Jde o zákaznickou škálovatelnost. Zákazník si volí např. velikost, tvar a funkce. Představte si svět, ve kterém si každý může koupit přesně takový výrobek, jaký chce, s funkcí, kterou potřebuje, s designem, který se mu líbí, za stejnou cenu a ve stejném dodacím čase, jako v případě výrobku vyráběného hromadně. Ať už se

bude jednat o auto, oblečení nebo telefon. Nová výrobní revoluce to umožní. Pokročilí roboti mohou být naprogramováni tak, aby vyrobili jakoukoliv konfiguraci výrobku, bez potřeby dodatečného času na přestavení nebo na náběh. 3D tiskárny okamžitě vyrobí jakýkoliv zákaznický design. **Nyní jsme schopni vyrobit dávku jednoho výrobku** za stejných podmínek, jako u dřívější hromadné výroby.

1.5 Struktura této práce

Pokud je možné pomocí 3D tisku digitálně vyrobit složitou součástku, proč by nebylo možné digitálně vyrobit celé auto? Hledejme odpověď v otázce: „Co muselo být splněno, aby se složitá letecká tryska, zmíněná v předchozí kapitole, dala vůbec vyrobit?“ Bylo to možné, protože každá její část je digitálně popsána v návrhu, tedy produkt je zcela digitální. Takže ano, je možné digitálním způsobem vyrobit auto, za předpokladu, že každá výrobní operace bude digitálně popsána. Můžeme také říci, že „životní cyklus produktu bude zdigitalizován“. *Od designu, přes objednávku, výrobu dílů, montáž a distribuci* k zákazníkovi, a už to tak zůstane do jeho *recyklace* či *likvidace*. Životní cyklus produktu se tedy skládá z několika dobře identifikovatelných fází. V kapitole **2.1** jsou uvedeny bližší informace týkající se životního cyklu produktu a podrobněji popsány právě jeho jednotlivé fáze.

V průběhu návrhu výrobku, jeho výroby, ale i po jejím ukončení vzniká velké množství (dnes již) elektronických dat. Tato data je nutné používat k výrobě, archivovat, třídit, sdílet a provádět s nimi mnoho dalších činností, samozřejmě ve víceuživatelském prostředí, aby mohly sloužit jako podklady k dalšímu vývoji. Celý životní cyklus výrobku je nutno řídit a usměrňovat, zejména v jeho raných fázích, což by bez nástrojů pro *řízení životního cyklu výrobků* v současné době nebylo vůbec možné. Těmto nástrojům, základním pojmům a myšlenkám se hlouběji věnuje kapitola **2.2**.

Uvážíme-li, že máme digitálně k dispozici kompletní data životního cyklu výrobku, máme vlastně digitální model výrobku. Tento model, bez ohledu na to, jak užitečný může být v různých fázích životního cyklu, zůstává statický, a tím pádem se jeho využitelnost v rámci života výrobku snižuje. Pokud přidáme možnost kontinuálně pořizovat data i v průběhu jeho dalších fází životního cyklu a udržovat tato data společně s modelem, získáme digitální stín, s jehož využitím lze provádět např. případové studie vlivu změny různých parametrů na stav výrobního systému. Pokud se výsledky simulací kontinuálně a automatizovaně promítají do změny parametrů systému, mluvíme o *digitálním dvojčeti*. Problematika digitálního dvojčete je detailně popsána v kapitole **2.4**, ale digitální dvojče se vyskytuje napříč celou prací.

Můžeme-li vytvořit kompletní digitální obraz (dvojče) produktu, lze jistě vytvořit digitální obraz samotného výrobního zařízení (které je také produktem, jen jiného výrobce). Přes vytváření modelu, resp. digitálního dvojčete výrobního stroje v kapitole **2.6**, se v kapitole **2.7** dostaneme k myšlence modelovat celou výrobní linku, potažmo ve **2.8** celou továrnu. Kvůli omezeným zdrojům (ať už lidským, či výpočetního výkonu)

je vždy účelné vytvářet modely jen s takovou mírou detailu, jaká je nezbytně potřebná pro zamýšlené využití. Zapojení digitálních dvojčat není věcí objevující se výlučně v průmyslu, a proto je na konci **8.4.2.** kapitoly popsáno několik příkladů využití mimo tuto oblast.

Práce dále pokračuje kapitolou **3**, zabývající se popisem transformace tradičních konceptů výroby k Průmyslu 4.0. Zmíněna je mimo jiné konvergence informačních a operačních technologií, která firmám přinesla nové, dříve netušené možnosti, ale také mnoho výzev a rizik. Jak proces konvergence postupoval, pronikaly postupně čisté informační technologie i do systémů a zařízení využívaných ve výrobní oblasti a v dnešní době jsou tyto technologie přítomny napříč celým výrobním procesem. Toto rozšíření se, mimo jiné, stalo jedním, ale nikoli jediným ze spouštěčů popisované transformace. Další spouštěče jsou popsány v kapitole **3.2.1**. V souvislosti s průmyslem 4.0 se objevil také nový pojem - kyber-fyzický výrobní systém - integrace fyzického a kybernetického systému schopna komunikace s okolím. Tyto systémy, popsané v kapitole **3.2.2** formují základ moderní průmyslové výroby.

Celou průmyslovou integraci je možné chápat ve třech různých rovinách (směrech). Jedná se o *vertikální* propojení, *horizontální* propojení a propojení *inženýrských procesů*. V kapitolách **3.1.1** a **3.2.3** je charakterizováno vertikální propojení, a to před transformací k Průmyslu 4.0 a po ní. Kapitoly **3.1.2**, resp. **3.2.4** popisují stejný způsobem horizontální propojení, a konečně kapitoly **3.1.3** a **3.2.5** uvádějí do stejného kontextu propojení inženýrských procesů v rámci celého životního cyklu produktu.

Digitalizace průmyslu znamená zcela nové způsoby komunikace a spolupráce mezi entitami na různých úrovních - od výrobních zařízení, až po celé podniky. Důležitou podmínkou pro možnost vzájemného porozumění a sdílení dat a myšlenek je seskupení rozdílných aspektů do společného modelu. Tím je tříosý model *RAMI*, popsaný v kapitole **4.1** reprezentující všechny charakteristické prostředky a aspekty výroby. Z tohoto obecného modelu vychází další, specifitější model Industry 4.0 komponenty (kapitola **4.2**). Ten formalizuje popis vlastností a schopností kyber-fyzických systémů tím, že definuje obálku - Asset Administration Shell, kterou může být každý potřebný fyzický objekt (stroj) obalen. Celý koncept obálky AAS je navržen jako univerzální, neboť obálka může zapouzdřit nejen stroj, ale také pouze jednu jeho část. Situace může být také opačná, a jedna administrativní obálka může obalit i více strojů. Všechny tyto koncepty jsou popsány dále v kapitolách **4.2.1** - **4.2.3**. Existují dva typy administrativní obálky - pasivní a aktivní. Pasivní slouží především k uchování dat souvisejících s produktem, výrobním zařízením či jakoukoli jinou entitou, kterou obalují. Informace jsou v obálce organizovány do tzv. submodelů tím způsobem, že informace pro jednu ucelenou doménu jsou vždy sdruženy do jednoho submodelu. Tuto organizaci popisuje kapitola **4.3** a způsob formálního vytváření obálky a submodelů je pak shrnut v kapitole **4.3.1**. V současné době existují modelovací nástroje, prostřednictvím nichž je možné popsanou pasivní část AAS vytvořit, udržovat a připravit k distribuci. Popis vybraných nástrojů má své místo v kapitole **4.4**.

Pro druhý typ obálky - aktivní AAS - je situace poněkud odlišná a dodnes není

přesně ustanovená forma její tvorby. Existují již pilotní projekty, které se o implementaci pokusily (kapitoly 4.5 a 4.6). Aktivní AAS je předmětem výzkumu a naše pracoviště bylo jedním z prvních, které teoretické myšlenky začalo testovat a dále rozvíjet v praxi s využitím unikátního testbedu a to mimo jiné i v rámci česko-německého projektu RACAS⁷.

Celý koncept Industry 4.0 komponenty s administrativní obálkou fyzického zařízení je zajímavý, avšak bez možnosti přistupovat k datům této obálky by byl k ničemu. V současnosti existují dva hlavní způsoby, kterými se taková komunikace mezi zařízeními realizuje. Těmi jsou *MQTT* (kapitola 4.6.1) a *OPC UA* (kapitola 4.6.2). Nelze si snad představit rozdílnější způsoby komunikace - od relativně jednoduchého MQTT, jehož specifikace se vešla do méně, než 100 stran, po nesmírně komplexní OPC UA specifikovaného na více, než 1200 stranách. Už z této skutečnosti je zřejmé, že tyto komunikační protokoly budou nasazovány pro odlišné případy využití. Základní rozdíly obou protokolů jsou shrnuty v kapitole 4.6.3.

Vraťme se ale k aktivní administrativní obálce. Slovo *aktivní* znamená, že obálka implementuje chování, které by se dalo označit jako *autonomní*, tedy umožňující proaktivně iniciovat komunikaci s jiným AAS za účelem splnění určitého cíle. Pokud je taková aktivní obálka vytvořena pro konkrétní kus (instanci) vyráběného produktu, může tímto cílem být právě organizace výroby fyzické instance tohoto produktu. Aktivní obálka, resp. algoritmus zabudovaný v ní, podle předem určeného mechanismu osloví jiné AAS, které obalují fyzické výrobní stroje a postupně vyjedná realizaci jednotlivých kroků výrobního procesu. Takové vyjednávání však není bez úskalí, neboť ve stejnou chvíli může existovat mnoho podobných aktivních AAS, zastupujících budoucí fyzické produkty, jejichž záměrem je využít výrobní stroje stejným způsobem. Problematika rozhodnutí, který výrobek bude zpracováván jako první a na kterém stroji je ve skutečnosti velice komplexní.

Právě proto je, počínaje kapitolou 5 velká část této práce věnována plánování výroby v podniku. V úvodu jsou objasněny některé základní pojmy z této problematiky: jsou jmenovány tři základní úrovně hierarchie plánování (kapitola 5.1), dále jsou popsány různé typy výroby dle tzv. bodu rozpojení objednávky (5.2) a vysvětleny rozdíly mezi plánovacími metodami od úplných začátků tohoto oboru, až po současnost.

Pro účely plánování výroby nestačí jednoduché rozlišení tří typů výroby - spojitě, dávkové a diskrétní (kusové), ale uplatňuje se jiný pohled - v kapitole 5.4 je specifikováno několik charakteristických typů, které se rozlišují podle způsobu zpracování zakázky ve stroji, či skupině strojů. Pro každý z těchto typů je definován matematický model. Nakonec je také nastíněn příklad jednoduchého algoritmického řešení vybraného typu. Ne všechny typy však lze řešit triviálně, některé (jmenovitě typ *Job shop*) vedou ke konstrukci matematického problému s vysokou komplexností, který je buď nutné řešit speciálními výpočetními metodami, nebo je potřeba řešení získat jiným způsobem.

Výsledek - plán výroby - může být *vypočten* buď aplikací matematické metody, či

⁷Digital Representation of Assets with Configurable AAS for CPP-Systems

odhadem řešení v řídicím členu (např. plánovacím modulu systému MES), nebo může být *dohodnut* za běhu systému jako důsledek vzájemné interakce administrativních obálek jednotlivých strojů a produktů. První zmíněné jsou označeny jako *centralizované metody řešení* a dále v textu se dělí na *deterministické*, popsané v kapitole 5.5 a *stochastické*, jejichž popis je součástí kapitoly 5.6. Druhý způsob získání řešení se dá označit za *distribované rozhodování*.

Obecně existuje několik typů architektur (průmyslových) systémů - od *centralizované a hierarchické* architektury, až po architektury *heterarchické*, tedy bez jasně dané hierarchie. Každý z výše uvedených typů rozhodování je vhodný pro aplikaci v jiném typu architektury systému - pro centralizovanou architekturu je možné získat řešení použitím centralizovaných metod v ústředním členu, pro heterarchickou architekturu, která centrální člen zcela postrádá, bude potřebné nasazení jiných algoritmů. Algoritmy pro architektury bez vertikální hierarchie jsou obecně postaveny na bázi agentních systémů. Ty jsou popsány v kapitole 6. Každý agent (jinak též entita podobná aktivní administrativní obálce) existuje v agentním systému a autonomně, případně v interakci s ostatními agenty sleduje splnění svých vlastních cílů. Mezi jednotlivými agenty může probíhat buď spolupráce, nebo nějaká forma soutěže. Při spolupráci jsou agenti rozděleni na dvě skupiny - agent manažer (produktový agent - administrativní obálka produktu) a agent řešitel (administrativní obálka stroje) a tyto dvě skupiny navzájem spolupracují prostřednictvím interakcí definovaných v protokolu Contract Net Protocol (kapitola 7.1), či v některé z jeho rozšířených variant. V rámci kapitoly 7.2 je popsána heuristická metoda, která slouží pro optimalizaci vkládání úloh do rozvrhů agentů strojů při využití právě CNP protokolu. Pomocí této heuristiky by se výsledný rozvrh a jeho realizace měla přiblížit způsobu plánování Just-in-Time. Kapitola 7.3 pak popisuje simulační prostředí, které bylo vyvinuto pro účely testování CNP protokolu a heuristického algoritmu.

Poslední, 8. kapitola této práce je věnována problematice výuky témat souvisejících s průmyslovou transformací. Vzhledem k tomu, že mnoho důležitých oblastí má přesah do příbuzných, avšak přesto velmi odlišných oborů - zejména strojního inženýrství a informatiky, bývají tyto oblasti pro studenty automatizace obvykle těžko uchopitelné. Pro usnadnění pochopení byl vybudován testbed - fyzické zařízení vyrábějící reálné produkty, prostřednictvím kterého je možné jednotlivé aspekty Průmyslu 4.0 demonstrovat. V rámci kapitoly 8.1 je popsána konstrukce testbedu po mechanické stránce, stejně tak jako hardwarové vybavení jednotlivých výrobních jednotek. Pro testbed byl adaptací komerčního řešení vytvořen ERP/ MES systém (popsán v kapitole 8.2) a vznikla aplikace rozšířené reality, jejíž popis je předmětem kapitoly 8.4.3. Tato kapitola také zároveň ukončuje celou tuto práci.

Kapitola 2

Digitálně podporovaná výroba

Životní cyklus výrobku lze přirovnat k životnímu cyklu člověka. Na jeho průběh, jak u člověka, tak u výrobku má výrazný vliv jeho vývoj a zrození. Zatímco člověka formují geny, u výrobku jsou to um a dovednost lidí, kteří se na jeho vzniku podíleli od prvotního nápadu, přes vývoj, návrh a realizaci. Poznamená ho také prostředí, ve kterém byl vyroben a nástroje, včetně těch softwarových, které byly při jeho výrobě použity. Celý (životní) cyklus je velmi komplexní a vyvinout, vyrobit a úspěšně prodávat nějaký produkt je výsledkem nejen úsilí jeho tvůrců, ale také výsledkem mnoha rozhodnutí, která musí být ve výrobním podniku podložena fakty a skutečnostmi. V této kapitole si proto přiblížíme nejen životní cyklus výrobku, ale také možnosti, jak lze životní cyklus ve výrobním podniku ovlivňovat a řídit, včetně softwarových platforem - systémů, které jsou k tomuto účelu dnes určeny, a umožňují udržovat kompletní databáze nejen všech konstrukčních modelů, ale také dalších souvisejících dokumentů. Takto je digitalizován celý životní cyklus produktu – od designu, přes objednávku, výrobu dílů, montáž a distribuci k zákazníkovi a zůstane to tak až do jeho likvidace.

Pod pojmy *digitalizace výroby* či *digitální výroba* si můžeme představit využití dostupných digitálních dat k vytvoření virtuálního obrazu reálné výroby. S informacemi, které se nacházejí v systémech pro řízení výroby¹ a konektivitou těchto systémů do výrobního procesu je možné s pomocí specializovaných nástrojů vytvořit virtuální entitu *digitální dvojče*, a dokonce celé virtuální prostředí, které je nazýváno *Digitální továrna* nebo *Digitální dvojče továrny*.

2.1 Životní cyklus výrobku

Pod pojmem *výrobek* si lze snadno představit nějaký hmotný produkt, který prošel výrobním procesem. Avšak marketingové teorie výrobek chápou mnohem širším způsobem, a to jako objekt, který lze na trhu prodat nebo koupit, a zároveň slouží k uspokojení lidských potřeb. Může jít tedy o hmotný či nehmotný statek [5].

¹MES - Manufacturing Execution Systems

Z hlediska marketingu je každý výrobek kombinací tří základních složek - *jádra užítka, vlastního produktu a rozšířeného výrobku*. Jádro užítka zajišťuje základní funkce výrobku, tedy důvod, proč zákazníci daný výrobek kupují. *Vlastní produkt* pak dotváří společně s jeho fyzickou realizací také styl (design), značka, kvalita, varianty vyhotovení (pokud jsou) a obal. Design výrobku určuje, jak bude vypadat a jak jej zákazník rozezná od konkurenčního produktu. Kvalita zosobňuje schopnosti výrobku plnit své parametry, jako jsou trvanlivost, spolehlivost či přesnost. Výrobek bývá často realizován v několika variantách, čímž může zaujmout širší škálu zákazníků. Obal pak plní jednak logistickou funkci (tedy chrání výrobek a umožňuje s ním manipulovat), a také nese informace o výrobku a slouží k jeho propagaci. Jako rozšířený výrobek se dají chápat doplňkové služby nehmataelné povahy, jako je např. instalace, záruka, doprava zdarma apod.

Každý výrobek na trhu si prochází svým vývojem, tedy životním cyklem. Intuitivně můžeme na životní cyklus výrobku pohlížet jako na život produktu - začínající procesem těžby surovin, výrobou výrobku, jeho prodejem, používáním a naposledy odstraněním jako nepotřebného odpadu. Tento pohled odpovídá spíše životnímu cyklu *konkrétního kusu výrobku*. Zamyslíme-li se nad životním cyklem z hlediska druhu výrobku, uvidíme jej jako *život v podniku od fáze návrhu a vývoje, přes zavedení na trh až po útlum zájmu zákazníků a stažení výrobku z trhu*. Dále v textu již bude popsán právě význam druhý. Teorie životního cyklu výrobku pracuje s pěti fázemi, přičemž jejich délka trvání je u různých druhů výrobků odlišná. Jednotlivé fáze můžeme pojmenovat jako

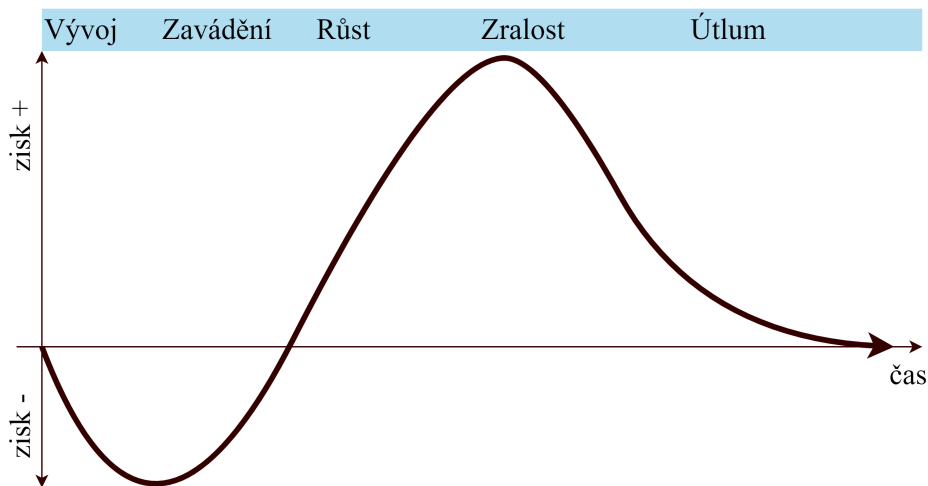
- vývoj,
- zavádění,
- růst zájmu o výrobek,
- kulminace zájmu a
- útlum zájmu.

Tyto fáze je možné znázornit grafem, tzv. S-křivkou. Graf uvedený na obrázku 2.1 popisuje vývoj zisku v čase a je typický tím, že se od začátku dostává do záporných hodnot. Často se také modeluje jako vývoj počtu prodaných kusů nebo tržeb v čase. V takových případech graf neprotíná svislou osu [5].

2.1.1 Fáze vývoje

Životní cyklus výrobku začíná v okamžiku prvotní zaznamenané myšlenky o výrobku. Tato fáze dále pokračuje přes výzkum, vývoj a realizaci výroby produktu. Prodeje výrobku jsou v této fázi velice malé, či spíše žádné a náklady vložené do výrobku se neustále zvyšují. Firma tedy může přecházet do ztráty.

Může se stát, že kdykoli v této fázi firma dospěje k rozhodnutí, že dále nemá smysl na vývoji výrobku pracovat, a vývoj utlumí. Pokud naopak firma započne marketingové



Obrázek 2.1: Typický průběh S-křivky

aktivity již ve vývojové fázi, může tím výrobku zajistit značný úspěch již v začátcích prodeje (takové rozhodnutí udělala například automobilka Tesla, která cíleným marketingem získala již v této fázi vývoje svého modelu X značné množství objednávek) [98].

Někdy bývá v literatuře tato fáze vynechávána, nebo je přidružena (zřejmě jako málo zajímavá z pohledu objemu prodeje) k fázi zavádění produktu na trh. Pro obor automatizace je však tato fáze nejzajímavější, neboť se během ní odehrává prakticky vše, co je pro nás podstatné.

2.1.2 Fáze zavádění

Fáze zavádění popisuje umístování výrobku na trh. Během ní probíhá nastavování výroby a logistiky produktu, vytváření prodejních kanálů a také určení marketingového mixu (ten popisuje pomocí několika parametrů strategickou pozici výrobku na trhu).

Fáze začíná nulovými prodeji a příjmy a většinou se vyznačuje pomalým růstem prodeje výrobku. Ten bývá způsoben kombinací různých vlivů – od neznalosti či neochoty zákazníků přijmout výrobek, až po možné technické problémy vznikající při výrobě a nutnost jejich ladění.

Pro fázi zavádění existuje několik strategií, které může firma upřednostnit. Nejsou zde konkrétně jmenovány, jejich výčet lze získat např. z [5], jen podotkněme, že volba strategie závisí nejen na typu výrobku, ale také na tom, jakým způsobem a jaké skupině zákazníků firma zamýšlí výrobek prodávat. Ne každá firma si může dovolit zvolit libovolnou strategii pro svůj výrobek, neboť záleží také na její proslavenosti.

Dobrou volbu strategie prokázala například firma Apple při zavádění svého prvního modelu iPhone. V té době sice existovaly dotykové telefony, avšak kvůli své neohrabanosti a technické nevytříbenosti nemohly být a také nebyly nijak masově rozšířené. Je pravdou, že Apple přišel skutečně s revoluční novinkou, kterou podpořil obří reklamní kampaní. Jelikož už tenkrát měla firma dobré jméno, mohla si dovolit nasadit relativně

vysokou cenu, a přesto o svůj výrobek okamžitě vzbudila ohromný zájem. Tento přístup se označuje jako *strategie rychlého sbírání* [81].

2.1.3 Fáze růstu zájmu

Po zavedení výrobku na trh se objem prodeje výrobku značně zvyšuje. Zákazníci již nakupují a další se k nim přidávají, zejména pokud jsou ohlasy veřejnosti na nově zavedený produkt pozitivní. Příjmy jsou vyšší než výdaje, podnik se dostává do zisku. Podnik musí nyní zaujmout růstovou strategii – má následující možnosti:

- Vytvoření modifikace výrobku
- Vstup na nové trhy
- Rozšíření distribuční sítě
- Změna typu propagace výrobku
- Snížení ceny výrobku

Zamysleme se mezi všemi uvedenými možnostmi zejména nad bodem *vytvoření modifikace výrobku*. Firma, která svůj produkt nemodifikuje (lépe řečeno neinovuje), se vystavuje riziku, že zájem zákazníků začne opadávat. V tuto chvíli bude produkt generovat již jen minimální zisky a firma se bude muset spolehnout na jiné produkty ze svého portfolia (pokud takové existují). Na scénu nyní také vstupuje konkurence, která je motivována vidinou možného zisku a snaží se přidat již zavedenému produktu nové vlastnosti, a tím původní produkt překonat. Konkurenční firma také může při vývoji svého výrobku těžit (a dnes také často těží) z technického řešení již existujícího produktu. Vhodným a výhodným postupem v této fázi je proto začít přemýšlet nad modifikací – vylepšením produktu.

2.1.4 Fáze zralosti

Rychlý růst prodeje charakteristický pro fázi růstu střídá nyní jeho zpomalení. Produkt je na trhu zavedený, poptávka i prodeje jsou víceméně stabilní. Povědomí o výrobku v cílové skupině zákazníků je na maximu. Tlak konkurence se zvyšuje. Firma na výrobku stabilně vydělává, musí však investovat do jeho nástupce, případně do marketingu v reakci na růst konkurence. Jedná se o nejdelší fázi životního cyklu výrobku. Délka jejího trvání značně závisí na odvětví, ve kterém se firma pohybuje. Zatímco ve světě počítačů či mobilních telefonů to jsou řádově měsíce, u automobilů je to déle než rok a v jiných odvětvích to může být ještě mnohem déle.

Fázi zralosti lze prodloužit inovací produktu. Inovace může být v podobě změny výroby produktu, zvýšení jakosti či zlepšení funkcí a vzhledu. Zákazníci modifikace

hodnotí zpravidla pozitivně a často jsou ochotni za zlepšení kvality zaplatit vyšší cenu. Známým příkladem inovovaných produktů jsou facelifty automobilů. Zhruba v polovině životního cyklu modelu (cca po třech letech) přicházejí automobilky s faceliftem, který za minimální náklady omladí starý model.

Životní cyklus výrobku lze také prodloužit tím, že jej firma zavede i na zahraniční trhy, kde může žít ještě několik dalších let. Takovým výrobkem byla svého času Škoda Octavia 1. generace, jejíž výroba se těsně před ukončením této fáze cyklu v ČR přestěhovala do Číny [133].

2.1.5 Fáze útlumu zájmu

Jakmile je trh nasycen, začnou prodeje výrobku klesat. Pokud se jedná o produkt, který je založen na skutečných potřebách zákazníků, bývá pokles prodejů postupný. Byl-li produkt zákazníkům vnucen masivní reklamou, je útlum zájmu rychlý. Ve fázi útlumu se firma musí rozhodnout, jakým způsobem ukončí výrobu a prodej – buď může čekat, dokud zisky z prodejů budou pokrývat výrobní náklady nebo může ukončit výrobu a prodeje rovnou. Neprodané kusy mohou sloužit jako náhradní díly při servisních opravách již prodaných výrobků.

2.1.6 Výjimky při modelování S-křivkou

Popsaný cyklus platí pro mnoho skupin výrobků. Existují však výjimky. Pomocí S-křivky nelze modelovat výrobky a služby, které splňují základní životní potřeby, jako jsou jídlo či energie. Dále jsou to unikátní či nenahraditelné produkty, které se vyrábí v jednom kusu či relativně malých sériích, například pro oblasti medicíny či strojírenství.

Mezi výjimky řadíme také komponenty všech větších funkčních celků. Pokud například firma vyrábí LCD panely do monitorů, je pravděpodobné, že je dodává více výrobcům monitorů, a tak počty prodaných kusů panelů úzce souvisejí s počty prodejů monitorů.

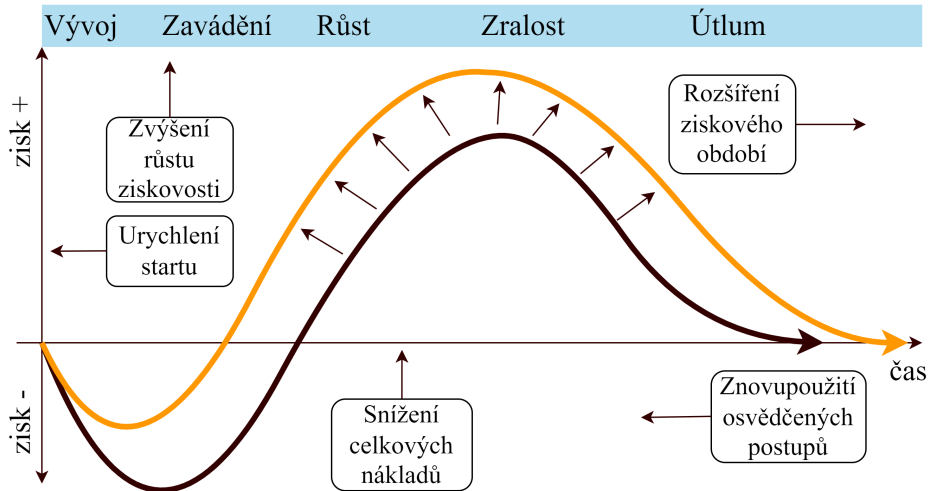
Další skupinou jsou výrobky, jejichž životní cyklus je příliš dlouhý, například konkrétní značka a druh piva. Z časového průběhu prodejů lze pouze obtížně vyčíst relevantní informace pro kontext životního cyklu.

2.2 Správa životního cyklu výrobku

Správou životního cyklu výrobku se rozumí proces řízení celého životního cyklu napříč všemi popsanými fázemi, tedy od prvotního nápadu až po ukončení servisu. Správa životního cyklu výrobku (PLM z anglického Product Lifecycle Management) sdružuje podnikové procesy, systémy, data a pracovníky a pro výrobní podnik i jeho partnery vytváří páteř informací o výrobku [7].

Pro správu (také se ale používá pojem řízení) životního cyklu výrobku jsou vyvíjena softwarová řešení, která zahrnují spolupracující infrastrukturu počítačových aplikací používaných k práci se všemi uvedenými entitami v průběhu celého cyklu.

Dříve, než se pustíme do popisu těchto softwarových řešení, pro jejichž označení se vžila zkratka PLM systémy, pojdme se ještě jednou podívat na typický průběh S-křivky. Možnosti změny tvaru charakteristiky jsou znázorněny na obrázku 2.2.



Obrázek 2.2: S-křivka popisující životní cyklus produktu

2.2.1 Možnosti ovlivnění životního cyklu

Vývoj a výroba produktu sestávají z dlouhé a časově náročné řady procesů, které mohou snadno být zpožděny různými vlivy, jako je např. zpoždění dodavatelů materiálu či poskytovatelů služeb. Ve fázi vývoje také často dochází ke zpožděním kvůli podcenění časové náročnosti jednotlivých dílčích částí.

Rozdělením vývoje pomocí PLM systémů do paralelně běžících procesů se celý vývoj zefektivní. Udržením konzistentních, přesných a synchronizovaných informací o produktech, procesech a zdrojích od počátku životního cyklu se *minimalizují náklady spojené se změnami*, ať už přicházejí v kterékoli části cyklu. S pomocí PLM je možné snížit *počet prototypů*, efektivně spravovat návrh výrobku a spolupráci v rámci dodavatelského řetězce. PLM také otevírá cestu k využití virtuálního dvojčete produktů a výrobních zařízení, které umožní zkrácení času nastavení výroby [46].

Díky PLM systémům je možné podchytit a zaznamenávat *preferenci cílových zákazníků* a trhu, které se týkají konkrétního výrobku. Následně je možné sladit tyto požadavky s konkrétními inovačními prvky s ohledem na další aspekty jejich realizovatelnosti. PLM systémy umožňují efektivní *správu variant výrobků*. Informační báze PLM systému zvyšuje rychlost pracovních skupin a omezuje množství chyb při návrhu výrobku tím, že všechny osoby, které se ho účastní, mají trvalý přístup k nejnovějším informacím o produktech a mohou provádět změny v reálném čase.

Díky koncentrací informací v PLM systému je možné během fáze zralosti produktu *efektivně určit směr jeho dalšího vývoje*. Dále je možné *identifikovat vhodnost vytvoření speciálních doplňků či nabídek*. PLM také umožňuje vznik tzv. produktových platforem, které lze opakovaně využívat napříč dalšími produkty (například podvozková platforma MQB firmy VW).

Jakmile jsou informace o produktu uloženy do PLM systému, zůstávají v něm uloženy a mohou být kdykoli opětovně využity. Ať se jedná o data původních výrobků, výrobní postupy, procesy, intelektuální kapitál, statistiky prodejů, servisu apod.

Tedy abychom to shrnuli. Základní mantrou, která provází myšlenku řízení životního cyklu, je snížení celkových nákladů napříč všemi fázemi životního cyklu výrobku. Jak je uvedeno výše, lze toho dosáhnout zefektivněním vývojových a výrobních procesů, možností dělat cílená a kvalifikovaná rozhodnutí a přecházením potřeby kaskádového přepracování navržených výrobků.

2.2.2 Softwarové systémy pro správu životního cyklu

Vznik PLM systémů se datuje do doby, kdy 2D kreslení začalo být nahrazováno 3D modelováním a při vývoji výrobků se začalo úzce spolupracovat v projektových týmech. Při této spolupráci bylo potřebné zajistit sdílení aktuálních verzí návrhů. Stále dokonalejší konstrukční CAD/CAE systémy totiž umožňovaly konstruktérům, technologům a dalším pracovníkům v předvýrobních etapách vytvářet mnohem složitější návrhy, varianty i komplexní řešení nových produktů či jejich inovací. Výrobci CAD systémů proto začali vyvíjet vlastní systémy pro správu dat (zpočátku pojmenované PDM) integrované do CAD řešení.

Na systémy PDM vyhrazené pro správu produktových dat se postupem času nabalily další domény, kterými byly:

- Systémový inženýring, který sbírá všechny požadavky na výrobky (primárně požadavky zákazníků, ale také legislativní požadavky např. na bezpečnost, kvalitu atd.), a koordinuje práce na vývoji nových a inovacích stávajících výrobků.
- Správa výrobního portfolia, která slouží zejména k alokaci výroby podle plánu vývoje produktu a dostupných výrobních zdrojů.
- Systémy pro vývoj, konstrukci a přípravu výroby, které slouží k počítačové podpoře konstrukce mechanické i elektrické části výrobku, pro technickou přípravu výroby a simulace výrobních procesů.
- Systémy pro správu výrobních procesů, které umožňují navrhovat a optimalizovat vývojové a výrobní procesy obsahující podporu změnových řízení, apod.

Systém pro řízení výrobního cyklu výrobku je tedy informační platforma, která v sobě sdružuje technické, výrobní i marketingové údaje o portfoliu výrobků firmy.

Nejedná se zde pouze o statické informace, ale dynamicky se vyvíjející datové množiny, rozhodovací procesy, poznámky, komunikaci mezi členy týmů a další. Všechny změny provedené v návrzích výrobků jsou dokumentovány systémem PLM umožňuje i přístup k historickým údajům.

2.2.3 Složitost výrobku a PLM

Složitost současných výrobků trvale narůstá – jejich návrh je složitý, výroba je složitá a složitá je i podpora. Tento trend pozorujeme v posledních deseti letech ve všech výrobních odvětvích. Ať už je náročný samotný výrobek nebo výrobní proces, dosáhnout ziskovosti vývoje a výroby v jednadvacátém století je stále náročnější.

- **Mechanická složitost:** Dokonce i *jednoduché* mechanické výrobky jsou dnes složitější. Modernější konstruktérství může produkovat výrobky s více funkcemi a současně je zmenšit a zlevnit. Díky lepším možnostem návrhu a ověřování posunuli návrháři hranice a dnes vytvářejí výrobky, které byly dříve nedosažitelné. Díky novým materiálům, například kompozitním materiálům a nanomateriálům, se zlepšila flexibilita a inovace, avšak objevily se i nové problémy. Zejména ztížená modularizace a opakovatelnost.
- **Mechatronika:** Kromě vyšší mechanické složitosti jsou moderní výrobky také *inteligentnější*. Téměř každý současný výrobek, od letadla po domácí spotřebič, již obsahuje elektroniku a elektronické řídicí prvky. Například moderní automobily mohou být vybaveny 50 i více mikroprocesory, které řídí všechny funkce od brzdění přes časování motoru až po zábavu. Tato kombinace mechanických, elektrických a softwarových součástí neboli mechatronika činí výrobky stále složitějšími.
- **Zkracování vývojového cyklu:** Vývojový cyklus výrobků se neustále zkracuje. Ať už díky rychlému technologickému pokroku a tlaku zákazníka na stále modernější produkty, nebo vlivem konkurenčního boje. Například Škoda Octavia první generace byla vyvíjena 4 roky paralelně s výrobou Felicie a výroba Octavií pak trvala dalších 10 let. V dnešní době Škoda uvádí novou variantu jednoho z devíti modelů průměrně každých 6 měsíců.
- **Výroba množství variant:** Vzhledem k tomu, že výrobci soutěží na globálních trzích, musí produkt vyrábět v mnoha různých variantách. Kromě jazykové lokalizace software či návodů také platí (obrazně), že jedna velikost se nehodí pro všechny. Dalším problémem mnoha variant a distribuovaného prodeje je servis, neboť výrobky musí být opravitelné takřka *kdekoli* na světě.
- **Distribuce vývoje i výroby:** Globalizace ovlivňuje také vývoj i výrobu produktů. Firmy chtějí těžit ze schopností místních odborníků a často i levné pracovní síly, a proto stále častěji využívají globální dodavatelské řetězce. Díky

globální výrobě se nejen usnadňuje lokalizace výrobků, ale také se zkracuje doba uvedení na trh a optimalizují se náklady.

- **Zaznamenání požadavků zákazníků:** Firmy již dnes pochopily, že tím nejcennějším, co mají, jsou informace. Proto se snaží (a v budoucnu budou čím dál více) získat co nejvíce informací od zákazníků užívajících jejich produkty, aby na jejich základě mohly vyvíjet lepší produkty, či mohly např. optimalizovat kvalitu a servis těch stávajících. Sofistikované produkty budou stále více ovlivňovány jejich firmwarem, který bude možné u mnoha typů produktů aktualizovat vzdáleně.
- **Udržení ziskovosti po celou dobu životního cyklu:** Předchozí aspekty se týkaly zatím pouze návrhu výrobku a procesu jeho výroby. Výrobci se však musí zaměřit také na další fáze cyklu, které jsou pro celkovou ziskovost zásadní. Každý se můžeme zamyslet nad tím, co od konkrétního výrobku, pokud si jej koupíme, očekáváme. Máme zájem, aby vydržel na věky? Jistě ne, už vzhledem k současnému tempu technického vývoje výrobek morálně zastará za několik let. Do té doby si však jako zákazníci přejeme (a výrobce koneckonců také), aby výrobek fungoval bez poruch. Dnešní konkurenční trhy neodpouštějí chyby ve výrobě, špatnou kvalitu a vysoké výrobní náklady. Servis během cyklu výrobku, hraje významnou roli. V některých případech je servis nastaven tak, aby bylo dosaženo vyšších marží na servisních pracích, protože různé faktory stlačily směrem dolů ceny samotných výrobků. Platí to hlavně tam, kde servis je jen sporadický, či preventivní – například u automobilů. Takový typ servisu se pak promítá i do záručních lhůt, které bývají pěti i víceleté.

Pro komplexní příklad kombinace všech uvedených aspektů je možné vybrat firmu Škoda Auto. Asi není nutné ji moc představovat - jedná se o jednoho z nejstarších světových výrobců automobilů, který vznikl už v roce 1895[15]. V roce 1991 došlo ke spojení Škody s koncernem Volkswagen, který po celém světě zaměstnává téměř 700 000 lidí [117]. Pro firmu se stal důležitým milníkem rok 2014, kdy Škoda ve svých závodech po celém světě poprvé vyrobila více než milion vozů. Aktuální roční celosvětová produkce se pohybuje kolem 750 000 vozů [134]. Aktuálně Škoda vyrábí 11 modelových řad, které prodává na více než 100 trzích světa [104, 134].

Nenapadá mě žádný předmět *denní potřeby* složitější, než je osobní automobil. Při kvalitě, v jaké jsou vyráběny, objemu produkce a snad nekonečnému počtu nabízených variant je automobilka schopna dosáhnout ziskovosti právě jen a pouze díky komplexním systémům pro správu životního cyklu produktů! Škoda auto využívá softwarový systém, který je popsán v následující kapitole.

2.3 Software pro správu životního cyklu

V této kapitole se odrazíme od potřeb a požadavků specifikovaných výše a popíšeme softwarový produkt, který v sobě shrnuje funkcionalitu, podle výše uvedené vize.

Tím produktem (lépe řečeno sjednocujícím prvkem celého produktového portfolia) je software Siemens Teamcenter². Zde uvedeme stručný výčet jeho charakteristik [41] vždy spolu s krátkým komentářem, jak ta která vlastnost souvisí s řízením životního cyklu.

2.3.1 Archiv pro uložení modelů

Teamcenter slouží jako archiv, kde jsou uloženy 3D modely všech konstruktérů.

- **Přístupy uživatelů:** Do tohoto archivu mohou vstupovat všichni uživatelé, kteří mají udělen přístup. Některé části archivu mohou vyžadovat speciální oprávnění. Teamcenter automaticky ukládá jména autorů modelu, čas založení a pak dále jména osob, které model upravovaly či schvalovaly a opět datumy těchto operací. Eviduje se také, kdo s kterým modelem momentálně pracuje.
- **Verzování a revize:** Zaznamenává se celá historie vývoje modelu včetně provedených změn. Díky používání revizí a jejich schvalování je možné ukládat klíčová provedení modelu, která jsou použita ve výrobě nebo v dalších projektech. Díky integraci s CAD systémem se ukládají rovněž verze, tedy jednotlivá uložení změn modelu.
- **Podklady modelu:** Pro každý z modelů a každou revizi je možné připojit jeden nebo více podkladů – souborů libovolného typu (dokumentů, e-mailových příloh, výkresů, fotografií apod.). V rámci tzv. relací je možné řízeně vytvářet vazby mezi dokumenty a modely, které se následně uplatní při vyhledávání dat. Každý dokument je rovněž možné opatřit klíčovými slovy, pomocí kterých lze opět prohledávat.
- **Klíčové informace:** Pro model je možné definovat sadu důležitých – klíčových údajů. Tyto údaje jsou zapisovány do modelu při jeho tvorbě v prostředí CAD. Díky integraci jsou tyto údaje synchronizovány do TC a obráceně. Údaje mohou být rovněž automaticky přenášeny do dalších dokumentů (např. manuálu ve Wordu, kalkulací v Excelu) nebo systémů (export dat pro výrobu).
- **Stav zpracování:** Teamcenter umí pracovat s tzv. stavem modelu. Tento stav je využíván při řízení procesů, které nad modelem mohou probíhat. V každém okamžiku jsou tedy uživatelům přístupné informace o tom, zda nad modelem proběhl nějaký řídicí proces, je možné zobrazit jeho aktuální stav nebo jeho historický průběh. Rovněž je možné vidět jeho budoucí kroky.
- **Probíhající procesy:** Teamcenter umožňuje definovat Workflow procesní toky, pomocí kterých řídí a sleduje podnikové procesy. Těmi mohou být například

²Přestože je plnění výše uvedených potřeb a požadavků demonstrováno v této práci výhradně na něm, není tento produkt samozřejmě jediný. Mezi další významné výrobce PLM patří například společnost Solidworks Corporation se svým Solidworks PDM, Autodesk, Inc. s produktem Autodesk Vault či PLM Windchill společnosti PTC. Existuje také celá řada výrobců malých PLM systémů, které bývají nejčastěji ve formě webových aplikací.

- Poptávka – proces zahrnuje a udržuje všechny podklady, vyjádření k termínu a ceně, vytvořené materiály poptávky (kalkulace, model, výkres, harmonogram, projekt, nabídka, smlouva).
- Změna – proces zahrnuje a udržuje podklady změny, dotčené modely a výkresy, sleduje průběh změny, vyjádření ostatních útvarů, návrh na změnu a schválení změny.
- Výroba – proces zajišťuje provedení všech potřebných činností – vytvoření nového NC kódu, vytvoření technologického postupu, promítnutí změny do následných dokumentů a systémů (aktualizovaný výkres, NC kód, technologický postup, návody), generování a export dat pro systém řízení a plánování výroby.

Protože správa procesů je jednou z klíčových funkcí PLM, bude v další části této kapitoly uvedeno ještě několik příkladů spolu s detailnějším popisem.

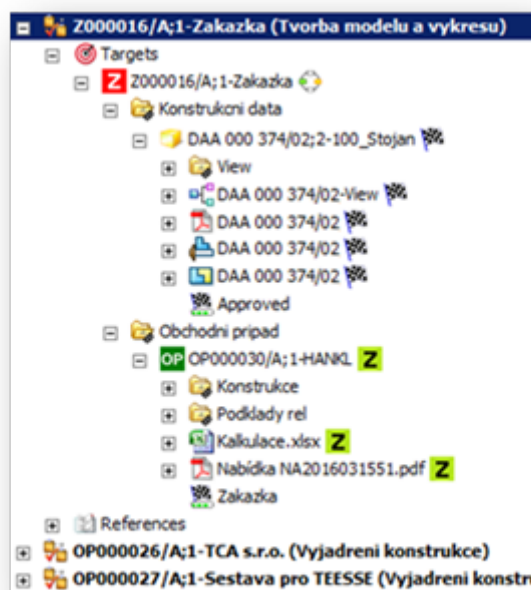
- **Efektivní prohledávání zdrojů dat:** Model není potřeba hledat pouze podle názvu souboru, ale využít je možné všechny dostupné údaje (autor, datum, klíčové údaje, stav, klíčová slova atd.) Je možné připravit vlastní vyhledávací formuláře či filtry. Lze rovněž hledat dle kusovníkového rozpadu (tedy části nějakého celku či naopak, kde je použita ta která část). Jak již bylo uvedeno, pro model jsou okamžitě nalezeny všechny klíčové dokumenty provázané pomocí relace.

2.3.2 Způsob ukládání podkladů v Teamcenteru

Soubory jsou v Teamcenteru uloženy pouze jednou, nevytváří se žádné duplicity. Pokud je potřeba konkrétní dokument přiložit (propojit) k dalším entitám, vytvoří se pouze odkaz. Připojený soubor je ve všech umístěních vždy stejný, a tedy provedená změna souboru se automaticky zobrazí ve všech odkazech. Tím je zaručena jednoznačnost dat [41].

V Teamcenteru jsou všechna data ukládána prostřednictvím tzv. položek. Založení položky se provede jednoduchým výběrem z menu aplikace. Položka je dále používána jako nositel skupiny příbuzných informací. Uživatel při zakládání vybírá typ položky, který chce založit. Systém Teamcenter umožňuje nastavení automatického číslování přidávání položek. Mezi položky se řadí například:

- **Konstrukční položka** – zastřešuje kompletní informace o vyráběném dílci. Položka v sobě ukládá mimo samotného modelu také kusovník, tedy seznam všech podsestav, dílů a výchozích materiálů, ze kterých je sestavená. Položka dále obsahuje 2D výkres a PDF export tohoto výkresu (který se automaticky přegeneruje vždy po změně výkresu). Součástí struktury položky je také informace o jejím stavu (řádek approved - schváleno) a relační adresář View, ve kterém jsou vloženy odkazy na nižší položky použité v kusovníku.



Obrázek 2.3: Příklad uložení položek a dokumentů v cílech procesu [41]

- **Obchodní případ** – slouží k seskupení podkladů a informací o konkrétní poptávce. V relačním adresáři Konstrukce jsou uloženy podklady od konstruktérů, vzorové konstrukční položky. V relačním adresáři Podklady jsou uloženy všechny ostatní podklady (převážně od obchodníka). Obchodník na základě podkladů vytváří soubory Kalkulace a Nabídka.
- **Zakázka** – slouží k seskupení podkladů a informací o konkrétní zakázce. V relačním adresáři Konstrukční data jsou uloženy požadované revize konstrukčních položek zakázky (ukládají se pouze vrcholy, tedy poslední platné). V relačním adresáři Obchodní případ je vložena odpovídající položka typu Obchodní případ pro snadné zobrazení již poskytnutých podkladů (po rozkliknutí položky).

Teamcenter umožňuje definovat, jaké vlastnosti bude uživatel zadávat při zakládání položek (včetně deklarace, že údaj je povinný, typu hodnoty, umožnění výběru jedné z n hodnot atd.). Umožňuje také specifikovat strukturu pro konkrétní typ položky.

Pokud není uživatel omezen nastavením přístupových práv nebo jiným nastavením systému, vždy má přístup k veškerým souborům a informacím vloženým do archivu. Pro práci se soubory se využívá metoda check-in/check-out. V případě, že uživatel chce editovat nějaký soubor, provede napřed jeho rezervaci (check-in). V tu chvíli mohou ostatní uživatelé tento soubor jen prohlížet. Zároveň ale vidí, kdo má soubor zarezervován a pracuje na jeho editaci. Po dokončení úprav uživatel soubor uvolní (check-out), a tím jej opět zpřístupní pro editaci jinému uživateli. Toto platí pro všechny soubory vložené do archívu (nejenom 3D), i soubor typu MS Office tedy otevře k editaci ve stejný čas pouze jeden uživatel. Kontrastem s tímto řízeným přístupem k prostředkům

je možnost současné úpravy jednoho souboru více uživateli. Také tuto, v poslední době populární možnost Teamcenter podporuje.

2.3.3 Workflow procesy

Workflow, neboli pracovní či technologický postup, je schéma provádění nějaké komplexní činnosti rozepsané jako série jednoduchých kroků a vazeb. Workflow procesy tedy koordinují velkou většinu úkolů, které jdou nad rámec individuálního rozhodnutí jediného uživatele. Právě pomocí workflow je možné dosáhnout efektivní práce. V oblasti výrobních podniků lze workflow procesy využít na celou řadu operací, jmenovitě např. pro

- **Vyjádření** různých útvarů podniku k modelu či výkresové dokumentaci
- **Schválení konstrukční dokumentace**, které zahrnuje schválení modelu, výkresu i dalších dokumentů.
- **Přípravu a schválení návrhu na změnu**, tedy shromáždění potřebných podkladů a vyjádření k chystané změně. Změnové řízení je podrobněji popsáno v dalším textu.
- **Provedení samotné změny**, tedy provedení změny v konstrukci, koordinace úkolů na vytvoření opravy NC kódu a technologického postupu a promítnutí změny do podnikového ERP.
- **Export schválené dokumentace**, kdy systém vyexportuje údaje o položkách, výkresy a kusovníky pro celý rozpad vybrané položky.
- **Vytvoření nabídky**, při které systém shromáždí podklady a vyjádření napříč útvary, dále pak kalkulaci a poté proběhne odsouhlasení termínů.
- **Realizace zakázky**, která sestává z definice obsahu, připojení podkladů, rozpadu na jednotlivé položky a úkoly, sledování stavu přípravy výroby a předání dat do podnikového ERP.
- **Řízení projektu**, tedy definice týmu, připojení podkladů, rozpad na položky a úkoly, sledování stavu projektu, řízené ukládání dokumentů, a v poslední řadě vyhodnocení projektu.

Konkrétní pracovní úkol, jako nejmenší část workflow procesu, obsahuje vždy zadání (targets) a odkazy (references) na relevantní dokumenty a položky, které se ho týkají. Při plnění úkolu má tedy uživatel přístup ke všem potřebným informacím a nemusí nic dále vyhledávat. Uživatelské rozhraní úkolu dále obsahuje prostor, ve kterém se uživatel může k úkolu vyjádřit – slovním komentářem, a/nebo výběrem jedné z připravených hodnot. Uživatel úkol uzavírá vydáním rozhodnutí o schválení, případně o nutnosti

přepřerování úkolu. Systém následně automaticky spustí následující úkol pro dalšího konkrétního uživatele.

Pracovní úkol workflow procesu je často přiřazen ke splnění nikoli fyzickému uživateli, ale samotnému informačnímu systému. Takový úkol pak proběhne, jsou-li splněny podmínky k jeho spuštění a poté pokračuje proces automaticky dále. Sem je možné řadit úkoly typu

- **Kontrola plnění** uživatelských úkolů a relevantních dat. V případě, kdy uživatelský úkol není splněn do požadovaného času, je uživatel znovu upozorněn prostřednictvím e-mailu nebo může být úkol přiřazen automaticky jinému uživateli.
- **Rozhodnutí**, kterým směrem bude pokračovat proces.
- **Výběr uživatele** pro následující uživatelský úkol.
- **Změna stavu** položky.
- **Vygenerování a odeslání e-mailové zprávy**.
- **Vygenerování PDF**, office souborů dle šablony.
- **Export** výkresů, položek, kusovníků do ERP.

Příklad workflow procesu – poptávkové řízení [41]

Obchodník – založení obchodního případu: V tomto příkladu je obchodník prvním uživatelem, který dostane přiřazený úkol - založení nové položky typu Poptávka do systému. Po založení obchodník schválí přidělený úkol, čímž potvrdí jeho zpracování.

Obchodník – příprava Formuláře poptávky: Jeho dalším úkolem je vytvořit klíčový dokument Formulář poptávky (který může generovat ručně, nebo dle šablony). Do něj zapíše dostupné údaje (ručně, případně mohou být synchronizovány) a soubor vloží do archivu Teamcenter na konkrétní určené místo (většinou do konkrétní revize položky Poptávka). Po dokončení operace obchodník schválí přidělený úkol, čímž potvrdí jeho zpracování.

Obchodník – vložení podkladů: Následně obchodník do archivu Teamcenter vloží na určité místo všechny získané dokumenty – podklady. Tyto podklady se umísťují většinou do relačního adresáře Podklady konkrétní revize položky Poptávka. Může se jednat o e-mailové zprávy, MS Office dokumenty, PDF, fotografie, 2D, 3D modely, videa apod. Je vhodné si uvědomit, že každý dokument i model je uložen v archivu Teamcenteru pouze jednou. Opakované vkládání do relačních adresářů různých položek již představuje pouze vytváření odkazů na původní dokument. Po dokončení operace obchodník schválí přidělený úkol, čímž potvrdí jeho zpracování.

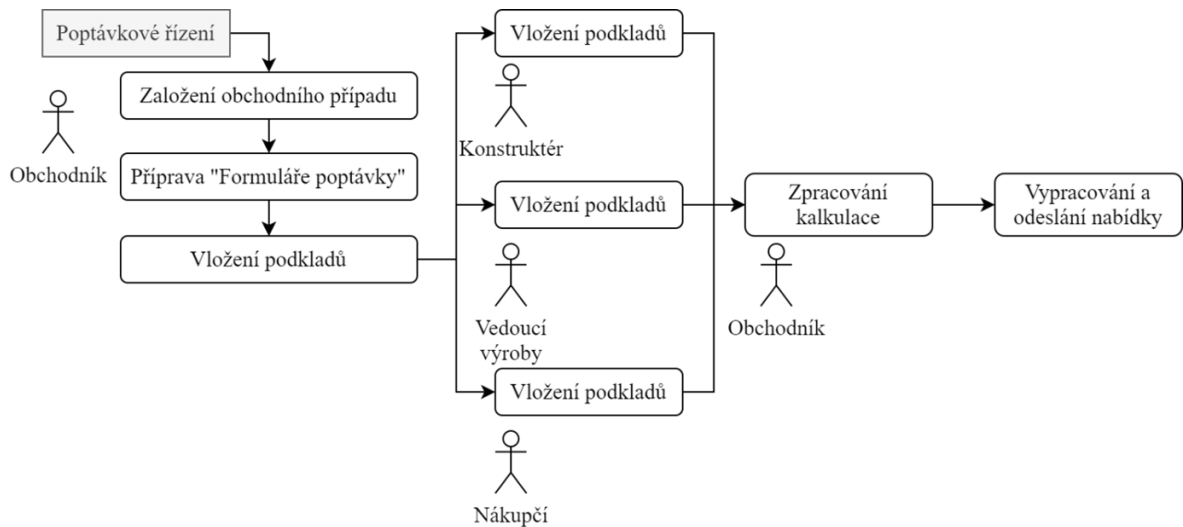
System Teamcenter – hlídání procesu: Po splnění předchozího úkolu systém vygeneruje nové paralelní úkoly pro tři uživatele zároveň (konstruktéra, vedoucího výroby a nákupčího).

Konstruktér – vložení podkladů: Konstruktor se seznámí se zadáním a příloženými podklady, vyplní jemu určená pole dokumentu Formulář poptávky – zapíše své vyjádření. Zároveň může doplnit další soubory – podklady. Pokud chce, může konstruktor přiložit i odpovídající 3D modely. Po dokončení práce opět schválí přidělený úkol.

Vedoucí výroby – vložení podkladů: Postupuje stejně jako konstruktor.

Nákupčí – vyplnění vložení podkladů: Postupuje stejně jako konstruktor a vedoucí výroby.

System Teamcenter – hlídání procesu: Během plnění předchozích úkolů kontroluje přístup k Formuláři poptávky a jiným dokumentům. I když mohou uživatelé plnit úkoly zároveň, nemohou editovat žádný z dokumentů najednou. Teamcenter pomocí zamykání dokumentů zabezpečí, aby si soubory uživatelé vzájemně nepřepisovali. Teamcenter dále pomocí synchronizace vlastností převede klíčové informace ze souboru Formulář poptávky do vlastností položky Poptávka. Teprve po splnění všech tří paralelních úkolů systém spustí nový úkol, nyní opět pro obchodníka.



Obrázek 2.4: Workflow procesu vytvoření poptávky, dle [41]

Obchodník – zpracování kalkulace: Obchodník zároveň s úkolem obdrží veškeré dříve zpracované informace a podklady. V tomto případě vidí příloženou konkrétní revizi položky Poptávka. Ta má ve svých vlastnostech popsány základní údaje. Vidí také příložený soubor Formulář poptávky – ihned si ho může zobrazit a prohlédnout. Vidí i relační adresář Podklady a v něm uložené veškeré podklady týkající se tohoto konkrétního případu poptávky. Je-li připojen, může vidět i odpovídající konkrétní revizi 3D modelu.

Po seznámení se se zadáním a příloženými podklady a informacemi vytvoří Ob-

chodník soubor Kalkulace, může se jednat např. o soubor XLS vytvořený pomocí šablony z údajů položky Poptávka. Obchodník soubor dále dle potřeby upraví. Nakonec odsouhlasí splnění úkolu.

Obchodník – vypracování a odeslání nabídky: Obchodník v rámci tohoto úkolu vytvoří soubor Nabídka, může to být dokument typu Word vytvořený pomocí šablony z údajů položky Poptávka. Obchodník soubor dále dle potřeby upraví. Nabídka bude opět uložena u položky Poptávka. Vytvoří e-mailovou zprávu, do které vloží soubor Nabídka, případně přiloží další vybrané podklady či Kalkulaci a soubor odešle. K položce Poptávka přiloží i odeslaný e-mail. Nakonec odsouhlasí splnění úkolu.

Systém Teamcenter – uzamčení procesu: Po splnění posledního úkolu provede Teamcenter uzamčení všech souborů připojených k této konkrétní položce Poptávka včetně souborů v relačním adresáři Podklady. Výjimkou jsou přiložené 3D modely – ty podléhají vlastnímu procesnímu toku Schválení modelu. Poté dojde i k uzamčení vlastní revize položky Poptávka.

Veškeré podklady a vytvořené dokumenty jsou uzamčeny tak, aby nemohlo dojít k jejich úpravě nebo doplnění po zpracování a odeslání nabídky. Kromě těchto informací je k dispozici i zápis průběhu procesního toku Workflow, který zobrazuje časově kdy a komu byly úkoly postupně přidělovány, jak byly plněny a obsahují i vyjádření uživatelů.

Další možnosti pro workflow proces

Požadavek na úpravy nabídky: Pokud vznikne požadavek na úpravu již schválené a odeslané nabídky, může být založena nová revize této konkrétní položky Poptávka a může být znovu spuštěn procesní tok na úpravu nabídky. Nové dokumenty budou ukládány do nové revize položky *Poptávka*, přitom však budou, díky propojení s původní revizí stále k dispozici i původní dokumenty. Na závěr bude vypracována nová kalkulace, vypracována a odeslána nová nabídka a opět dojde k uzamčení všech dokumentů včetně uzamčení této nové revize položky *Poptávka*. Pomocí revizí *Poptávky* je pak snadno dohledatelné, na základě kterých nových podkladů či informací došlo ke změně kalkulace a nabídky.

Zaznamenání rozhodnutí zákazníka: Popisovaný proces skončil zasláním nabídky, protože se jednalo o krátkou ukázkou. Pro potřeby sledování dalšího vývoje by proces poptávky obsahoval i úkol *Rozhodnutí zákazníka*, kde by po obdržení vyjádření zákazníka *Obchodník* připojil dokument odpovědi od zákazníka, případně dokument *objednávky* a hlavně by nastavil příznak úspěchu poptávky. Podle příznaku by pak systém nastavil stav položky (nabídka přijatá, zamítnutá).

Spuštění následných procesů – příprava a provedení zakázky: V případě úspěchu nabídky na základě stavu položky může Teamcenter automaticky spustit následné procesy. O přijetí zakázky jsou tak automaticky informováni potřební uživatelé a jsou zahájeny potřebné činnosti. Položka *Poptávky* je pak s novým procesem automaticky spojena a uživatelé tak mají snadno k dispozici všechny potřebné

údaje a dokumenty i během procesu *Zpracování zakázky*.

Přizpůsobení workflow procesů

Přestože pro většinu často využívaných workflow procesů (zejména těch, které byly popsány výše) existují optimalizované ustálené sekvence jednotlivých úkolů, zvyklosti v každé firmě se liší, proto se mohou lišit i workflow procesy. Konfigurace procesů a jejich jednotlivých kroků (úkolů) probíhá při integraci PLM systému do firmy. Integrátor (firma) na základě dodaných podkladů vytvoří nové procesy, případně provede přizpůsobení existující množiny procesů. Takovým přizpůsobením je dosaženo stavu, kdy automatizovaný proces nejlépe odpovídá původní zažité sekvenci kroků a uživatelé se v něm přirozeně dokážou orientovat.

Změnové řízení

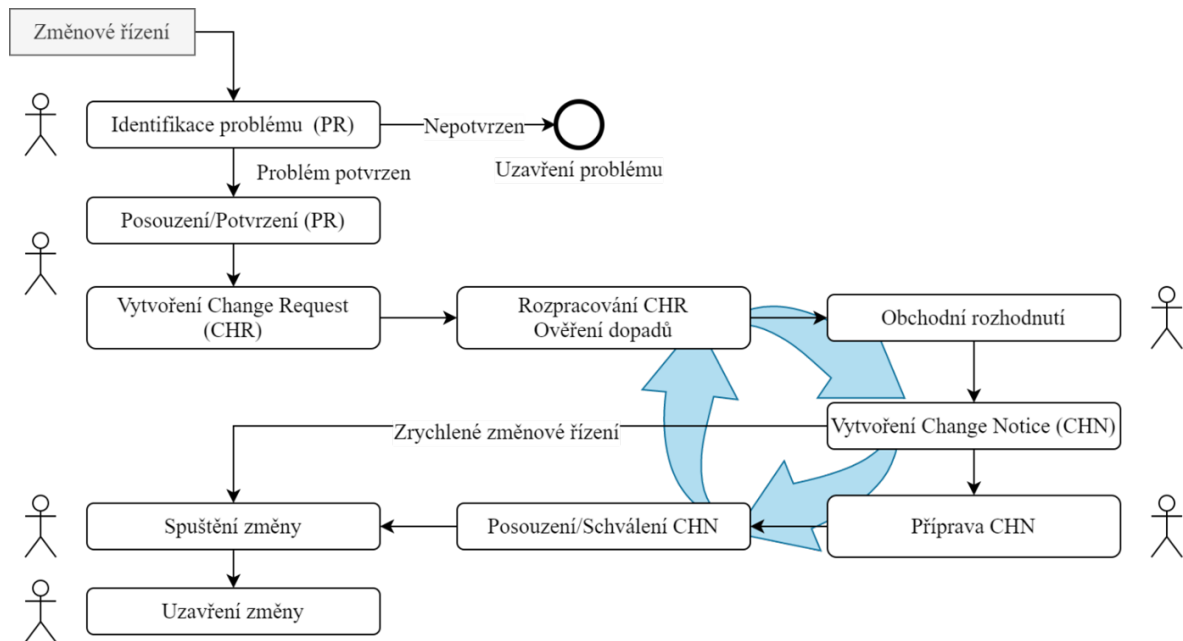
Jedním z velmi důležitých typů workflow procesů je změnové řízení. Změna v různých fázích vývoje a výroby je každodenním jevem ve většině průmyslových odvětví. Správně zavedené změnové řízení je tudíž zásadní pro schopnost výrobce pružně reagovat na zákaznické požadavky, a také na změny vzniklé úpravami, vylepšeními nebo odstraňováním nedostatků. Náklady na změnu v průběhu vývoje jsou výrazně nižší, než v pozdějších fázích (platí pravidlo 1:10:100).

Každý výrobce se tak pochopitelně snaží vyřešit většinu změn v konstrukční fázi. V dalším textu budou prezentovány možnosti, jak odhalit potřebu změny již během vývojové fáze, dokonce ještě před výrobou prvního fyzického prototypu výrobku.

Nyní se však zaměříme na jeden z nástrojů Teamcenter – modul Změnové řízení. Tento modul zajišťuje komplexní propojení všech dat souvisejících se změnou a realizací změny pomocí workflow procesů. Změnové řízení obsahuje nejen všechny potřebné informace pro provedení změny (aktuální 2D, 3D data, kusovníky), ale nabízí také možnost rychlé analýzy dopadu změny. Ta obsahuje informace o tom, jaké položky budou změnou ovlivněny, a umožní tak učinit správné rozhodnutí o dalším postupu. Změnové řízení v Teamcenteru je předdefinováno podle praxe z podniků, které dlouhodobě vykazují úspěchy na poli *managementu změn*, zároveň je také v souladu se zavedenými standardy.

Standardní průběh změnového řízení je možné popsat workflow procesem na obrázku 2.5. Jednotlivé úkoly jsou přiřazeny různým uživatelům. Pro jednoduchost zde nejsou uvedeny jejich role, jejich zastoupení v jednotlivých krocích je pouze naznačeno.

- Problem Report (PR): slouží k popisu a základní identifikaci zjištěného problému.
- Change Request (CHR): slouží ke sběru dat o provedených rozhodnutích v rámci změnového řízení nad připraveným návrhem realizace změny.



Obrázek 2.5: Workflow procesu změnového řízení (dle [41])

- Change Notice (CHN): implementační plán - poskytuje detailní informaci o implementaci jednoho nebo více změnových požadavků.

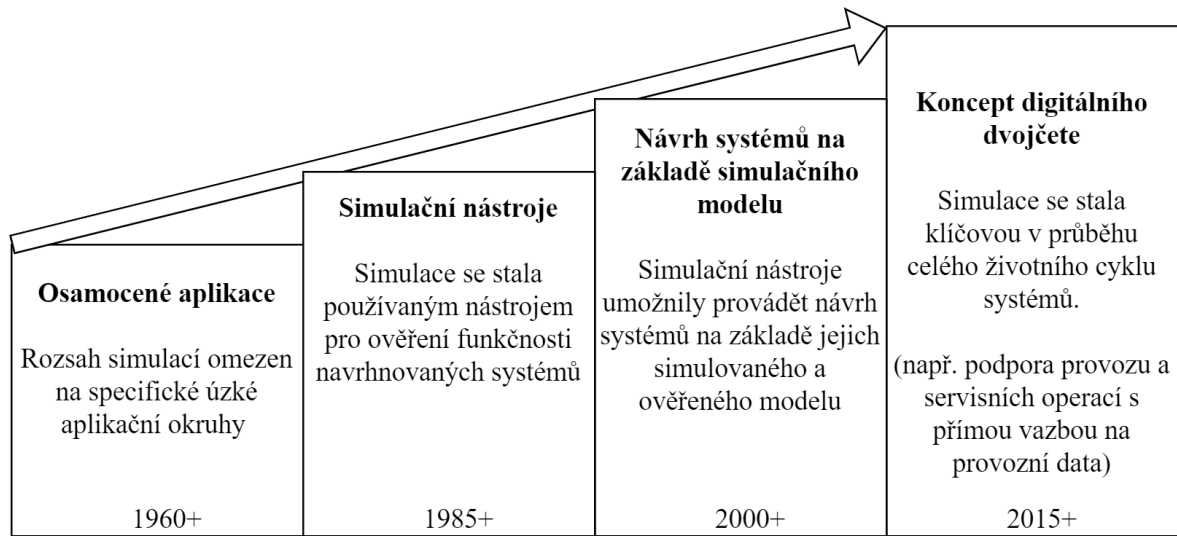
Popsaný proces se samozřejmě upravuje podle zvyklostí ve firmě a zpravidla také podle typu výroby a velikosti firmy.

2.4 Od fyzického dvojčete po to digitální

Myšlenka využití *dvojčat* – identických kopií zařízení pro vývoj a testování sahá do sedmdesátých let, kdy NASA pracovala na projektu Apollo. Během mise Apollo 13 došlo dne 14. 4. 1970 k explozi kyslíkové nádrže, která vážně poškodila servisní modul, a tím ohrozila nejen misi samotnou, ale i životy celé posádky. Tým inženýrů na Zemi musel promptně najít efektivní řešení, jak minimalizovat negativní dopad nečekaného poškození. Jelikož měli na zemi k dispozici věrnou repliku kosmické lodi i s technickými detaily, mohli důvěryhodně nasimulovat nebezpečnou situaci a otestovat možná řešení. Právě možnost fyzicky vyzkoušet hypotetické postupy posádku zachránila.

V roce 2003 publikovala NASA spolu s US Air-Force článek o současných problémech při vývoji a provozu nové generace vyvíjených dopravních prostředků. Namísto použití dosavadních rozšířených postupů, jako byly matematické pravděpodobnostní metody vycházející z vlastností využitých materiálů, heuristické přístupy ke konstrukci, testování fyzických prototypů a zanedbávání rozdílů testovacích a skutečných provozních podmínek navrhuje článek využití nového paradigma – digitálního dvojčete. Toto paradigma zahrnovalo sestavení věrného modelu a spuštění simulace, která společně s palubním systémem vozidla, poskytujícím informace o aktuálním stavu a historii vo-

zidla, otevřela možnosti simulovat chování vozidla v reakci na různé podněty, a tím předvídat chování v budoucnu. To vše s cílem dosáhnout nových milníků v bezpečnosti a spolehlivosti těchto vozidel.



Obrázek 2.6: Historie simulačních konceptů, dle [88]

2.4.1 Nízká úroveň integrace – digitální model

Základem digitálního dvojčete je digitální model, který můžeme chápat jako digitalizovanou verzi již existujícího nebo plánovaného fyzického objektu či systému. Takto si můžeme představit například parametrický trojrozměrný konstrukční návrh nějakého výrobku, výrobní linky, stavební plány budovy, ale také fyzikální model technologického procesu. Digitální model nezahrnuje žádné automatizované datové toky mezi digitálním a fyzickým světem, je víceméně statický (pokud není ručně aktualizován) a existuje izolovaně. Proto digitální model, jakmile je jednou sestaven, *zůstává po celou dobu svého života neměnný* (samozřejmě v případě požadavků na revizi/facelift výrobků dochází k jeho ručním změnám). To znamená, že *změní-li se parametry modelovaného systému, model přestane odpovídat realitě a využívat jej k simulaci systému postrádá smysl*.

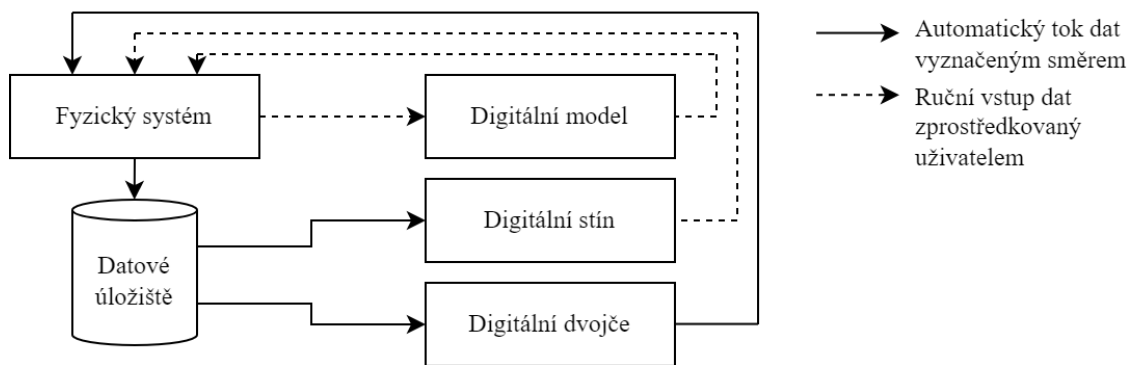
Je potřeba si rovněž uvědomit, že pokud mluvíme o kvalitním digitálním modelu pro simulace, nemusíme tím nutně chápat absolutně přesný trojrozměrný konstrukční návrh (dokonce to ani nemusí být konstrukční model). Systém vždy modelujeme takovým způsobem, aby byly zdůrazněny ty aspekty, které jsou pro konkrétní využití modelu důležité a ostatní bývají pro zjednodušení modelování, zkrácení potřebného času (a také pro zlepšení výpočetního výkonu) zanedbány.

2.4.2 Digitální stín

Digitální stín reprezentuje další krok od digitálního modelu založeného čistě na historických datech a znalostech. Stav systémů, nejen z hlediska aktuální činnosti, ale také

např. z hlediska opotřebení různých dílů, se v průběhu jejich životního cyklu neustále mění. Proto potřebujeme získávat data systémů nejen jednorázově během konstrukce digitálního modelu, ale neustále během doby provozu. Digitální stín tedy představuje koncept, ve kterém probíhá **jednocestná automatická výměna dat** směrem od fyzického systému k digitálnímu modelu. Změna stavu fyzického systému tedy vede ke změně parametrů digitálního stínu.

V digitálním stínu jsou shromažďována veškerá podstatná data - hodnoty snímačů, rozhodování řídicích členů, informace o zásazích operátorů, o objednávkách či produktech - která fyzický systém produkuje spolu s jejich historií. Změněné parametry digitálního stínu však nemohou být automaticky promítnuty do parametrů systému. Zde je potřebný manuální zásah uživatele (operátora).



Obrázek 2.7: Rozdíl modelu, stínu a dvojčete

2.4.3 Digitální dvojče

NASA ve svém článku definovala digitální dvojče jako “integrovanou multifyzikální víceměřítkovou pravděpodobnostní simulaci systému, která využívá nejlepšího dostupného fyzikálního modelu, dat ze senzorů, historie vozidla (či celé flotily) a dalších dostupných dat za účelem zrcadlit život systému – fyzického dvojčete” [88].

Pro naše účely můžeme použít pozdější definici, která říká, že *Digitální dvojče je virtuální interaktivní dynamická reprezentace fyzického systému, která umožňuje simulovat chování tohoto systému od jeho nejmenších částí, až po jeho celek* [40]. Existují dva obecné typy digitálních dvojčat: dvojče *typu systému* a dvojče *instance systému*. Zatímco digitální dvojče typu systému (výrobku) obsahuje data potřebná zejména k jeho výrobě (vytváření instancí), jako jsou 3D modely, kusovníky, apod., digitální dvojče *instance výrobku* modeluje konkrétní instanci a je založeno na propojení s fyzickým výrobkem po celou dobu životního cyklu a kromě výše uvedeného zahrnuje také informace o konkrétní instanci - výsledky zkoušek, testů, ale také aktuální či historická *runtime* data měřená snímači a případně predikované budoucí hodnoty. Sjednocením dat pořizovaných z řady instancních digitálních dvojčat vzniká *instancní agregované dvojče*, jehož pomocí je možné navíc vyhodnocovat parametry, jako *střední dobu mezi poruchami*.

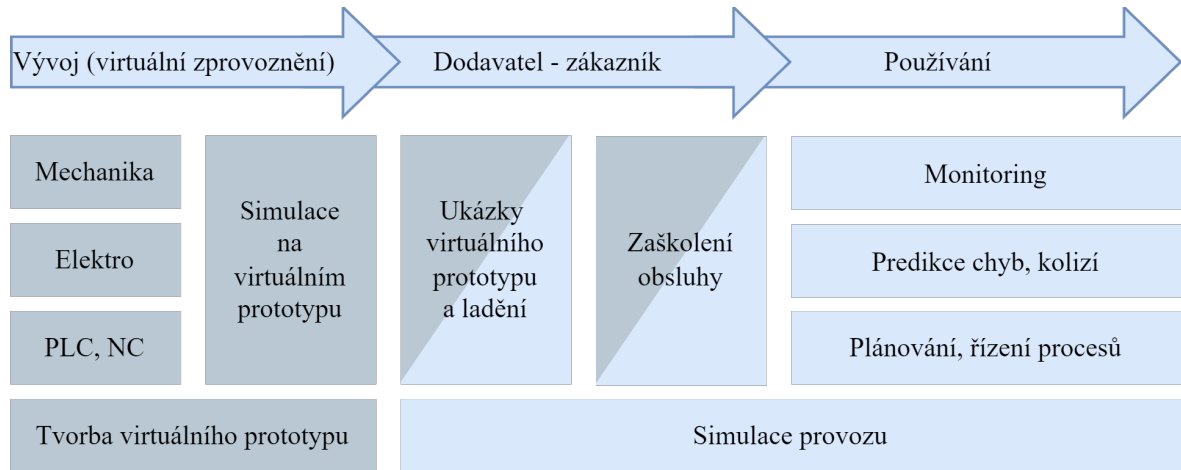
Na rozdíl od digitálního stínu u dvojčete v této podobně existuje obousměrná automatická komunikace mezi fyzickým systémem a modelem [88]. Jakákoli změna ve fyzickém systému se v projeví v modelu a obráceně. Na základě chování systému tak mohou být v reálném čase modifikovány parametry modelu a obráceně na základě využití modelu je možné provést cílenou změnu fyzického systému. Digitální dvojče tak může ovlivňovat přímo fyzický svět. Rozeznáváme dvě hlavní skupiny digitálních dvojčat (DT): *prediktivní* a *interogativní*. Zatímco první skupina slouží pro predikci budoucího chování a výkonnosti systému a využívá se jak v případě DT typu, tak pro DT instancí, druhá skupina se využívá pro správu instancí v průběhu jejich životního cyklu a získávání agregovaných údajů [40]. Princip funkce konceptu digitálního dvojčete lze popsat následujícími šesti kroky [70]:

1. Vytváření: Dochází ke sběru dat pro digitální dvojče pomocí senzorů na stroji a v jeho okolí. Mohou být doplněny i informace např. ze systémů plánování výroby apod.
2. Komunikace: Dochází k obousměrné komunikaci mezi fyzickým a virtuálním strojem v reálném čase. K tomuto účelu může být využita síťová komunikace.
3. Shromáždění: Jsou shromážděna a předzpracována data obdržena z fyzického stroje. Je možné využít cloudové platformy.
4. Analýza: Předzpracovaná získaná data jsou analyzována za účelem lepšího pochopení stavu fyzického stroje. Na základě toho je možné se rozhodnout pro určitá opatření.
5. Porozumění: Vizualizace výsledků analýz a rozdílů mezi fyzickým a virtuálním strojem.
6. Řízení: Jsou provedeny zásahy do stroje v návaznosti na předchozí kroky (servis, korekce apod.)

Ve fázi vývoje a uvádění do provozu, kde se digitální dvojče v současnosti nejvíce uplatňuje, slouží jako virtuální prototyp, který je možné virtuálně zprovoznit, a tím ušetří čas od návrhu po skutečné uvedení fyzického stroje do provozu. Díky možnosti paralelní práce více stran (konstruktéři mechaniky, elektrokonstruktéři, PLC i NC programátoři), které mohou současně vyvíjet každý svou část stroje, lze dosáhnout časové úspory až 30 % [105].

Digitální dvojče má své uplatnění i ve vztahu dodavatele (např. výrobce stroje či linky) a zákazníka (uživatel stroje, či operátora linky), neboť umožňuje zaškolení obsluhy ještě před dodáním technologie, a to i na dálku. Na monitoru počítače vidí operátor digitální dvojče a na softwarovém ovládacím panelu se pak naučí požadované úkony. Velkou výhodou je i fakt, že výrobce může zákazníkovi prezentovat stroj či linku ještě před jeho realizací, a zákazník tak může jednotlivé vývojové fáze odsouhlasit, případně snadno upřesnit své požadavky. Ty mohou být snadno implementovány do virtuálního prototypu a po odsouhlasení i do fyzického stroje [105].

Ve fázi používání zařízení zákazníkem má být digitální dvojče neustále spojeno se svým fyzickým dvojčetem. V reálném čase má reprezentovat jeho chování, zkoumat jeho stav a predikovat chyby a kolize. Na základě informací z digitálního dvojčete mohou být plánovány, případně také řízeny celé výrobní procesy. Využití digitálního dvojčete v rámci životního cyklu je znázorněno na obr. 2.8. Šedá barva znamená benefity pro výrobce, světle modrá pak benefity pro uživatele.



Obrázek 2.8: Využití digitálního dvojčete během životního cyklu entity, dle [105]

2.5 Digitální dvojče výroby

V rámci následujících kapitol budou představeny nástroje a možnosti, které vycházejí právě z PLM systémů a umožňují realizovat dvojče - předdigitální či digitální - záleží na tom, ke kterému pojmu se přikloníme. Ve výrobním prostředí lze digitální dvojče realizovat ve více doménách a s různými úrovněmi integrace a komplexity. Jednou z nich je digitální dvojče výrobku, jehož prostřednictvím je možné provádět řadu ověřování a testů ještě před výrobou prvního prototypu, přes dvojče výrobní linky až po dvojče logistických procesů. Každá aplikace se zcela liší svým využitím, rozsahem a úrovní simulace. Pro každou jsou také vhodné jiné softwarové nástroje [11, 12].

Termín **digitální dvojče výrobku** se dostal do širšího povědomí v roce 2002, kdy ho Michael Grieves definoval v souvislosti s řízením životního cyklu výrobku. Koncept digitálního dvojčete tehdy zavedl jako virtuální reprezentaci vyrobeného produktu a měl sloužit k porovnání výrobku s jeho inženýrským návrhem. Pro vytvoření digitálního dvojčete výrobku je možné využít soubor modelů informací z PLM systému [40].

S využitím digitálního dvojčete výrobku je možné provádět simulace, jejichž výsledky mohou vést ke změně parametrů designu produktu ještě před vyhotovením prvního prototypu. Následně, pokud již návrh produktu vyhovuje požadavkům, je po realizaci prototypu možné provést fyzické testy. Během sériové výroby každý z produktů podléhá ověřování kvality. Jakékoli odchylky od požadované kvality jsou automaticky v reálném čase porovnávány s digitálním dvojčetem za účelem odhalení jejich příčin

a jejich eliminace. Samozřejmě stále platí, že ve fázi inženýringu je (ten správný) model produktu využíván k vytvoření programů pro obráběcí stroje či 3D tiskárny a mnoha dalším úkonům. Při správném použití zajistí uvedení vysoce optimalizovaného produktu s minimem konstrukčních chyb.

Jak si tedy lze digitální dvojče představit? Nejčastěji jako digitální model jeho konstrukce, vytvořený v softwarovém nástroji, či nástrojích. Ten by však bylo možné využít jako jeden díl skládačky - není zdaleka jedinou možností reprezentace fyzického produktu. Existují i další:

- Numerický model mechanické konstrukce s ohledem na její parametry pružnosti a pevnosti.
- Model určený pro simulaci obtékání vzduchu kolem výrobku.
- Elektronické schéma pro modelování chování analogových elektronických obvodů výrobku.
- Stavový automat pro modelování chování výrobku.
- Stavový model výrobního procesu ve výrobní buňce používaný pro řízení operací a plánování toku výroby, apod.

2.5.1 Případová studie - Maserati

Firma Maserati historicky vkládá do svých vozů velký díl ruční práce, jejíž cena samozřejmě s časem narůstá. Značka však nechce z prestižních důvodů tuto ruční práci nahradit neosobní velkovýrobou. Nicméně výrobní náklady samozřejmě nesmí přesáhnout náklady na výrobu v průmyslovém měřítku, a tedy musí být zabezpečena konkurenceschopnost značky. To koneckonců platí pro každý podnik [8, 99].

Jednou z možností, jak výše zmíněného dosáhnout, je při vývoji nového vozu použít digitální dvojče. Právě takové dvojče sehrálo klíčovou roli i během vývoje modelu Maserati Ghibli (obr. 2.9). Pro jeho vývoj byla využita právě softwarová platforma PLM Tecnomatix. Díky digitalizaci mohla být vytvořena virtuální kopie souběžně s fyzickým vývojem vozu, která byla až do posledního šroubku stoprocentně věrnou podobou originálu. Během vývojové fáze byla pro optimalizaci procesů souběžně využívána data jak reálných, tak i virtuálních modelů.

To přineslo snížení nákladů a kratší dobu potřebnou na vývoj. Jako příklad může posloužit použití aerodynamického tunelu, jenž se používá k optimalizaci aerodynamických vlastností karoserie. Při použití virtuálního dvojčete posloužila naměřená data z malého počtu skutečných testů k rychlému a levnému dalšímu vývoji ve virtuálním prostředí. Neustálými drobnými úpravami digitální kopie vozu mohli vývojáři nacházet nové způsoby optimalizace tvaru a komponentů nově vznikajícího auta.



Obrázek 2.9: Výroba Maserati Ghibli [8]

Dalším příkladem jsou akustické vlastnosti (zvuk motoru). Vývojáři proto optimalizovali zvuk uvnitř vozu tak, že do interiéru prototypu umístili figuríny s mikrofony a data z nahrávek využili pro další virtuální testy.

Dále je možné zmínit testování prototypů na silnicích i zkušebních polygonech za současného sběru dat. Díky tomu bylo možno testovací jízdy opakovat libovolně často virtuálně za upravených podmínek a nové vozy optimalizovat. Digitální dvojče tedy firmě pomohlo minimalizovat počet drahých prototypů.

2.6 Digitální dvojče výrobního stroje

Celý koncept digitálního dvojčete jako virtuální reprezentace výrobku se později rozšířil i mimo původní záměr. Představme si výrobní zařízení - jednoúčelový manipulátor, či univerzální obráběcí stroj (frézka) v tovární hale. Digitální dvojče tohoto stroje obsahuje informaci o celé jeho konstrukci, fyzikálních vlastnostech, kinematickém a dynamickém chování. Digitální model je obousměrně propojen s fyzickým dvojčetem a přejímá v reálném čase informace o jeho aktuálním stavu a událostech na něm. Tyto informace jsou analyzovány a zobrazovány a jejich zpracováním je možné predikovat chyby, poruchy a kolize. V návaznosti na této predikci je možné ovládat fyzický stroj.

Základem pro vytvoření digitálního dvojčete výrobního stroje a jeho zprovoznování je tzv. *vázaná mechanická soustava* (Multibody Dynamics System, MBS), což je 3D model stroje, který v sobě zahrnuje dynamické vlastnosti. Simulacemi na tomto modelu lze zjistit časové průběhy sil a momentů v konkrétních vazbách stroje (ložiska, pohybové

maticy, vedení). Podle těchto průběhů pak lze dimenzovat části stroje a např. pohony. Praktické řešení probíhá tak, že v simulačním software jsou definována na geometrickém 3D modelu *tuhá* (v některých případech i poddajná) tělesa a jejich vzájemné vazby. Tělesům jsou přiřazeny materiálové charakteristiky a na základě těchto údajů software dopočítá hmotnosti a momenty setrvačnosti. Jedním ze software pro tyto simulace je např. **Mechatronics Concept Designer**, modul návrhového software **NX** firmy Siemens.

Při virtuálním zprovoznění a predikci chování skutečného stroje je nutné pro jeho virtuální interpretaci pomocí digitálního dvojčete průběžně provádět počítačové simulace. K tomu je potřebné využít tzv. co-simulační techniky - propojení více simulačních nástrojů ve prospěch jedné konkrétní simulace. Příkladem může být propojení Mechatronics Concept Designeru se skutečným, či virtuálním řídicím systémem, nebo jeho propojení se software **Matlab/Simulink** či **SIMIT**, které umožní realizovat fyzikální výpočty různých parametrů, případně **ANSYS**, který počítá analýzy metodou konečných prvků [70]. Pokud je digitální dvojče připojeno na skutečný řídicí systém, probíhají na něm simulace metodou HiL (Hardware in the Loop). To jsou simulace, ve kterých je digitální model systému řízen skutečným hardware a software. Pro skutečný hardware musí být emulovány senzory a akční členy z virtuálního prostředí tak, aby mohlo dojít k plnohodnotné simulaci.

2.6.1 Tvorba 3D modelu

Společnost Siemens nabízí software NX - CAD/CAE/CAM program, ve kterém konstruktéři strojů či linek vytvoří jejich 3D modely. Samotný NX umožňuje provést potřebné simulace a výpočty. Prvním krokem při vytváření digitálního dvojčete je tvorba 3D digitálního modelu (např.) v NX. Cílem je sestavit model, který bude co nejlépe reprezentovat reálný stroj. V ideálním případě je možné získat kompletní model od výrobce stroje. Často však výrobce dodává pouze sadu modelů jednotlivých dílů, kterým je nutné následně přiřadit standardní geometrické vazby.

2.6.2 Aplikace dynamických vlastností

Pomocí nástavby MCD pro NX je možné jednotlivým dílům vytvořeného 3D modelu přiřadit materiálové charakteristiky. Díky nim software dopočítá hmotnosti dílů a jejich momenty setrvačnosti. Tím jsou modelu přiřazeny dynamické vlastnosti. Jeden, či více dílů³ pak může být definován jako *tuhé těleso*. V dalším kroku se vytvoří kinematické vazby mezi jednotlivými tělesy tak, aby se vůči sobě pohybovaly stejně jako v reálném stroji. Pro styčné plochy (např. vedení lineárních os) je možné přidat koeficienty tření.

MCD umožňuje vytvořit také signály, které umožní zaznamenat stav *virtuálních senzorů* aplikovaných na stroj, jako jsou např. koncové spínače, a virtuální *akční členy*,

³Více dílů kvůli optimalizaci simulací, neboť výpočetní možnosti hardware a software jsou omezené

kteře jej umožní rozpořybovat. Prostřednictvím těchto signálů převedených do tzv. *sdílené paměti* (Shared Memory, SHM) je možné zajistit i komunikaci s dalšími nástroji - např. SIMIT, který zajišťuje detailní fyzikální simulace stroje, či Matlab, ANSYS, atd.

Geometrie, kinematické vazby a dynamické vlastnosti modelu obráběcího stroje virtuálně reprezentují konstrukční prvky a vazby reálného stroje. Nadeřinované senzory a akční členy reprezentují hardware skutečných akčních členů a sensoriky. Sdílená paměť, přes kterou jsou posílány signály, se dá označit za reprezentaci kabeláže na fyzickém stroji [105].

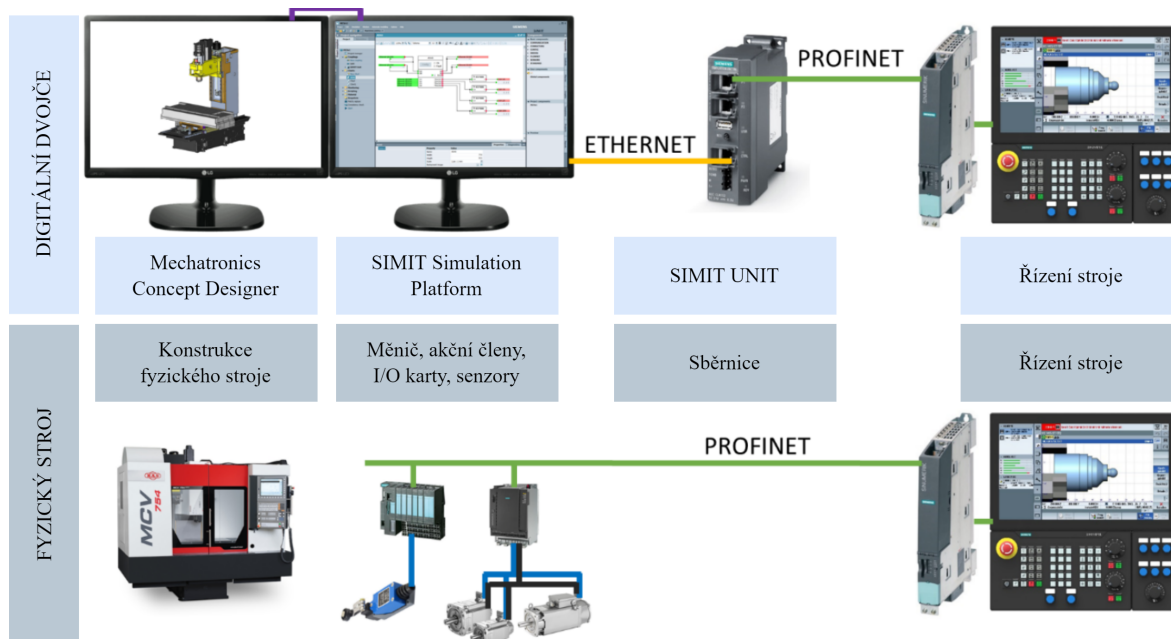
2.6.3 Implementace fyzikální simulace

Automatizační aplikace je při vývoji potřebné kompletně testovat a virtuálně zprovoznit. K tomuto účelu slouží platforma SIMIT Simulation Platform (SIMIT SP). Simulace vytvořená na této platformě je tak virtuální reprezentací logiky výrobního stroje, technologického procesu, či jiného systému a umožňuje testovat procesní program s využitím emulovaného hardware (SiL), případně s využitím reálného PLC hardware (HiL).

Díky integraci s dalšími vývojovými nástroji firmy je možné při vývoji digitálního dvojčete využít existující data. Simulace s využitím SIMIT tedy může přinést úsporu nákladů a zvýšit kvalitu automatizačních projektů. Díky svému širokému spektru záběru lze SIMIT využít napřič celým životním cyklem - nabízí všechny možnosti, které byly detailně popsány výše v 2.5 (ověření návrhu, simulace změn výroby, zaškolení obsluhy, apod.). S využitím platformy je možné provádět simulaci na třech úrovních:

1. *Signálová úroveň* - Po specifikaci a propojení signálů je možné provádět funkční testy na základě proměnných exportovaných např. z nástroje SIMATIC Manager. Rozpracovaný stav testu, případně celý testovací scénář lze zachytit a uložit pro pozdější analýzu.
2. *Úroveň zařízení* - SIMIT SP obsahuje množství již připravených komponent (senzorů, akčních členů, jako jsou ventily, či motory), případně je možné vytvořit vlastní zařízení. Tato zařízení SIMIT SP emuluje na základě stavu signálů z nižší úrovně. Takto lze testovat chování jednotlivých komponent (např. otevírání ventilu, aktivaci koncového spínače).
3. *Procesní úroveň* - Na této úrovni SIMIT SP umožňuje simulovat fyzikální chování technologie. Pomocí dostupných knihoven je možné vytvářet digitální modely/dvojčata chemických procesů, proudění kapalin, ale i strojů a zařízení, či celých výrobních linek. Platforma přináší pro každou z uvedených oblastí mnoho připravených modelů, které je možné dále modifikovat, ale také vytvářet vlastní [102].

Jak již bylo uvedeno výše, SIMIT SP nabízí propojení s dalšími nástroji. Jedním ze



Obrázek 2.10: Role nástrojů v systému digitálního dvojčete, dle [105]

způsobů je sdílená paměť (SHM) a propojení s nástrojem MCD. Při přístupu do sdílené paměti je nutné nastavit v obou nástrojích totožné mapování proměnných (názvy, adresy, datové typy vstupů/výstupů). Jiná varianta zase umožňuje přímé propojení se zařízením *SIMIT Unit*⁴. Všechny definované signály lze následně v SIMIT SP upravovat a propojovat jak mezi sebou, tak s vytvořenou emulací technologického procesu [105, 118].

2.6.4 Případová studie - Škoda Machine Tool

Společnost ŠMT pro potřeby prezentace na odborných veletrzích sestavila zmenšený model horizontální vyvrtávačky ŠMT - HCW 4S (což je největší horizontální vyvrtávačka, která se na světě vůbec vyrábí, viz obr. 2.11). ŠMT ve spolupráci s firmou Siemens implementovala digitální dvojčte tohoto stroje. Originál stroje má pojezd v ose X přes 25 m v ose Y je 10 m a jeho součástí je i nezávislá plošina. Na modelu byla, na rozdíl od skutečnosti, z důvodu zjednodušení zvolena varianta pevného spojení s vřeteníkem. Stroj je pracovištěm, které umožňuje frézovací, vyvrtávací i karuselovací operace. Cílem bylo ukázat, že panelem s reálným řídicím systémem je možné, stejně kvalitně jako stroj, řídit i jeho digitální dvojčte a na něm dokonale předem otestovat, jak se stroj bude chovat, až bude skutečně vyrobený. Na základě této myšlenky vznikly dva identické ovládací panely, přičemž jeden z nich je určen pro řízení modelu a druhý pro virtuální dvojčte.

⁴Jedná se o HW zařízení, umožňující komunikaci mezi řídicím systémem a SIMIT SP - zajišťuje úpravu I/O signálů tak, aby je byl schopen SIMIT přijmout. Pro řídicí systém simuluje signály ze senzorů a akčních členů tak, aby byly řídicím systémem přijaty jako signály ze skutečné technologie. Díky SIMIT Unit je možné provést HiL simulaci nezbytnou pro vytvoření digitálního dvojčete.

Přestože je model primárně určen pro veletrhy, po přidání originálního ovládacího panelu je možné na něm školit obsluhu v programování, testovat provoz a sledovat z nadhledu, jak funguje celek. Model naznačuje obrábění v šesti osách, čtyři osy stroje plus pojezd stolu a otáčení stolu. Na realizaci se podílely dva týmy - jeden tým vyřešil rozpohybování modelu v rámci softwaru NX Mechatronic Concept Designer. Druhý tým dal dohromady část SIMIT, čili simulace logického chování stroje.

Prostřednictvím digitálního dvojčete se dá odladit vše. PLC program, vlastní technologické programy, nasimulovat veškeré kolize, bezpečnost stroje, jestli jsou správně nadimenzovány pohony atd. Bylo rozhodnuto, že v budoucnu při vývoji nového zařízení – stroje nebo části stroje, firma půjde opět touto cestou [47].



Obrázek 2.11: Horizontální vyvrtávačka typu ŠMT HCW

2.7 Digitální dvojče výrobní linky

Budeme-li problematiku dvojčat posouvat v hierarchii fyzických zařízení podniku ještě výše, můžeme definovat dvojče celé výrobní linky a později celé haly. Na této úrovni již nebudeme provádět detailní fyzikální simulace přechodových dějů v elektrotechnice, ani počítat mechanické parametry jednotlivých částí výrobních strojů. Zde spíše bude potřeba vědět, jak jednotlivá výrobní i podpůrná zařízení spolupracují a podílejí se na výrobě produktu, nebo případně, jak by tato zařízení spolupracovat měla. K získání těchto znalostí mohou pomoci nástroje tzv. digitální továrny.

Digitální továrna je zastřešující pojem pro síť digitálních modelů strojů, robotů, dalšího vybavení, procesů a metod, které jsou integrovány v rámci průběžného datového managementu. Cílem je prostřednictvím propojení všech zmíněných prvků vytvořit prostředí pro komplexní plánování, projektování, ověřování a průběžné zlepšování

všech důležitých struktur, procesů a zdrojů reálné továrny v souvislosti s jejími výrobky” [78].

Digitální výroba prostřednictvím nástrojů digitální továrny je využívána v různých průmyslových sektorech a umožňuje inženýrům vytvořit úplný popis výrobního procesu ve virtuálním prostředí, mimo jiné pro obráběcí stroje, montážní linky, pracovní centra, návrhy zařízení, posouzení, ergonomického hlediska, apod.



Obrázek 2.12: Vyobrazení digitální továrny - 3D model výrobní linky

Vzniklé virtuální výrobní prostředí pak umožňuje:

1. Propojit produkty, procesy, provozy a informace o zdrojích, zobrazit je v kontextu a projít změnovými procesy s kontinuálním a komplexním přístupem k návrhu výroby.
2. Zobrazovat procesy ve 2D či 3D virtuálním grafickém prostředí, což může být využito pro
 - návrh a ověřování nových výrobních procesů (svařovacích, lakovacích, manipulačních, kompletačních, atd.), a to včetně automatického návrhu programů robotizace pro konkrétní roboty a automatizace;
 - optimalizaci nových i stávajících procesů, a to jak z hlediska časového, tak např. z hlediska ergonomie operátora. Je možné využívat znalosti získané ze simulací, již ve fázích plánování a vývoje produktu;
 - ulehčení rozhodování, neboť simulace usnadňuje přímou identifikaci následků uvažovaných změn;
 - ověřování provozu systému v různých mezních či kritických situacích. Je tedy možné testovat provoz zařízení tzv. „na hraně limitů“ a analyzovat jeho reakce, případně testovat různé typy selhání.

3. Optimalizaci procesu výroby dílů. Dovoluje vytvářet flexibilní pracovní postupy schopné zobrazit 2D/3D informaci o dílu spolu s pokyny pro jeho obrábění a opracování.
4. Zjednodušení sdílení údajů o kvalitě napříč podnikem vytvořením kompletních, ověřitelných řídicích programů založených na CAD.
5. Provádění výrobních procesů s přístupem k datům životního cyklu v reálném čase. Získávání zpětné vazby z aktuálních výrobních operací tak, aby získaná data mohla být analyzována a znalosti začleněny do procesu navrhování výrobku⁵, případně zvýšení přesnosti výroby.
6. Podporu six-sigma⁶ a štíhlých iniciativ poskytnutím grafického prostředí pro analyzování požadovaných změn.
7. Zvolit optimální strategii údržby díky přístupnosti historických dat o technickém stavu zařízení, jeho výkonu nebo opotřebování v reálném čase. V dnešní době se začínají prosazovat systémy, které za pomoci strojové inteligence analyzují sesbíraná data za účelem hledání kauzálních vztahů mezi mírou opotřebení a poruchovostí zařízení, realizují tedy tzv. prediktivní údržbu.

Při správném využití nástrojů digitální továrny dochází k odstranění kritických míst a optimalizaci procesu. Je možné se také rychleji a snadněji věnovat komplexním projektům, které by v minulosti mohly být považovány za příliš riskantní nebo nevýhodné. Schopnost simulace digitální výroby pomáhá snížit náklady na uvedení do provozu virtuálním ověřením programů robotizace a automatizace.

V předchozích kapitolách bylo diskutováno změnové řízení. S využitím simulačních nástrojů lze navrženou změnu ještě v průběhu změnového řízení promítnout do simulovaného výrobního prostředí a tam si ověřit, zda nová verze produktu či stroje odpovídá požadavkům. Takové ověření může být rovněž předepsáno jako jeden z úkolů workflow procesu změnového řízení.

Představme si chvíli, že neexistuje uvedená možnost ověřit funkčnost výrobního zařízení, tedy je potřeba postupovat při realizaci linky od mechanického návrhu, montáže, návrhu a osazení hardware komponent, návrhu, vývoje a testování software. Lehce se pak stane, že případná chyba nebude odhalena včas, ale bude se šířit do dalších vývojových, či provozních fází životního cyklu:

- Pokud je oprava chyby provedena na začátku, ve *fázi návrhu konceptu*, mohou být náklady (například) deset euro (oprava zabere krátký čas práce designera).

⁵To jinými slovy znamená využití provozních znalostí během plánovací fáze. Ve stejném čase se tak integrují činnosti skupin **výroby** a **projekce**

⁶Six sigma je strategie řízení používaná v různých odvětvích průmyslu. Six Sigma si klade za cíl identifikovat a odstranit příčiny defektů a chyb v procesech výroby a obchodu. Stupňování: One Sigma - efektivita 31 %, až Six Sigma - efektivita 99,9997 %.

- Pokud je oprava provedena až ve fázi vývoje, stojí již desetinásobek - stovky eur.
- Pokud je oprava provedena až ve chvíli, kdy probíhá nasazování strojů a řídicích programů, může se jednat o stonásobek původní hodnoty - tisíce eur.
- Pokud se na chybu přijde až když je nutno zastavit linku, na které probíhá sériová výroba, lehce se dostáváme na tisícinásobky původní hodnoty, tedy desetitisíce eur.

Současné iniciativy ve vývoji nástrojů digitální výroby zahrnují zdokonalení uživatelských zkušeností, aby informace vycházely z provedených úkolů, což uživatelům umožňuje lépe a rychleji rozhodovat. Směřuje se k přímému propojení s provozním hardwarem, jako jsou programovatelné automaty (PLC), řídicí systémy strojů, počítačové číslicově řízené (CNC) stroje a další. Byly také vyvinuty jednotné platformy k propojení systémů pro řízení výroby (MES).

V souvislosti s digitální továrnou je dobré zmínit i termín **virtuální továrna** (virtuální reprezentace fyzického výrobního systému), resp. virtuální realita (VR), která vytváří 3D obraz zahrnující všechny potřebné informace. VR vytváří simulované prostředí. S pomocí této technologie je možné vstoupit do vzdálené továrny, nebo do továrny ve výstavbě, např. jako je vzdálená ropná plošina, aniž je nutné opustit kancelář.

Nevýhody a úskalí konceptu

Přes všechna pozitiva a prokázanou přidanou hodnotu digitální továrny v podobě úspory času a peněz je třeba zmínit i možná úskalí. Schopnost simulovat téměř vše ve velmi realistickém virtuálním prostředí vede k *přesimulování*, tedy tendenci simulovat opravdu všechny procesy, tj. i ty, kde je aplikace běžných nástrojů, či inženýrské intuice dostačující. Mezi další rizika se obecně řadí:

- Potřeba vlastnit rozsáhlé know-how: vývoj modelu digitální továrny je časově náročný a vyžaduje pracovníky se specifickými znalostmi. V případě provozu, který se dynamicky vyvíjí (rozdává, mění) ani vývoj není nikdy u konce. Z toho plyne vysoká finanční náročnost.
- Nedotažení digitalizace do úspěšného konce: pokud z různých důvodů není digitální dvojče zcela dokončeno, investice je zmařena.
- Stává se, že i při největší snaze výsledky nesplní očekávání a digitální dvojče, do jehož vývoje se investovalo značné množství finančních prostředků, se nevyužije, či jen zcela marginálně (chybí know-how, ochota, odborníci). Využívání digitálního dvojčete totiž vyžaduje také radikální změnu fungování procesů a názorovou jednotu napříč celou společností. V případě, kdy byl vývoj spolufinancován např. z dotačních projektů, je toto riziko ještě vyšší.

I přes uvedená rizika současný vývoj v průmyslu dokazuje, že prostředky a metody digitální továrny nejsou slepou cestou a do budoucna se budou dále rozvíjet a jejich využití rozšiřovat.

2.7.1 Virtuální obraz reálné výroby

Máme-li hovořit o výrobním závodě nebo továrně, je dobré si připomenout, že továrnu tvoří *výrobní procesy + výrobní prostředky + zdroje*. Digitální dvojče výrobního zařízení, nebo též digitální továrnu je možné vidět jako souhrn informací, které vznikají v integrovaném souboru různých digitálních nástrojů při plánování a ověřování výrobních procesů a výrobních prostředků.

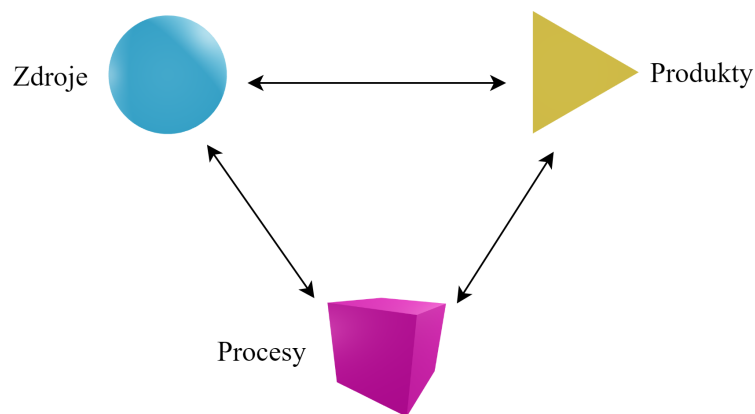
Nástroje digitální továrny zajišťují úzkou integraci konstrukce a technologie a spolupráci různých inženýrských disciplín od vývoje konceptu a konstrukce výrobku přes plánování a realizaci výroby až po post-výrobní etapy života produktu. V digitální továrně jsou všechny výrobní informace v jednom informačním zdroji, kde je vše snadno dostupné a dohledatelné pro všechny určené uživatele, a to během celého životního cyklu výrobku. Koncept digitální továrny tak nejen zjednodušuje práci s velkým množstvím dat, která je nutné zpracovat v průběhu plánování, projektování a realizace výroby, ale umožňuje také opakovaně využít zkušenosti a zdroje z předchozích projektů (*best practice a lesson learned*). V digitální továrně lze vydělit oblasti digitálního *plánování výrobních procesů*, digitálního *ověřování výroby a podporu výroby*. Při plánování výroby v softwarech digitální výroby se definuje kompletní seznam procesů (Bill of Process), kde základními stavebními kameny jsou:

- **co se bude vyrábět** – data produktu
- **čím se bude vyrábět** – data výrobních zdrojů, jako jsou pracovníci, roboti, nástroje, přípravky a další zařízení
- **jak se bude vyrábět** – informace o sledu jednotlivých operací s přiřazeným časem pomocí standardních metod normování času práce
- **kde se bude vyrábět** – rozvržení výrobní haly, tedy layout jednotlivých strojů.

2.7.2 Plánování výrobního procesu

Základním a nejdůležitějším nástrojem pro plánování či optimalizaci výrobního procesu, a zejména pro *tvorbu prostorového uspořádání systému*, může být produkt Process Designer⁷. Ten umožňuje souběžné plánování technologických týmů, analýzu ve

⁷Samozřejmě i jiný výrobce, nicméně v tomto textu budou i dále jako příklady použity právě produkty společnosti *Siemens*, nyní konkrétně portfolio produktů s názvem *Tecnomatix*.



Obrázek 2.13: Zdroje, produkty, procesy - propojení

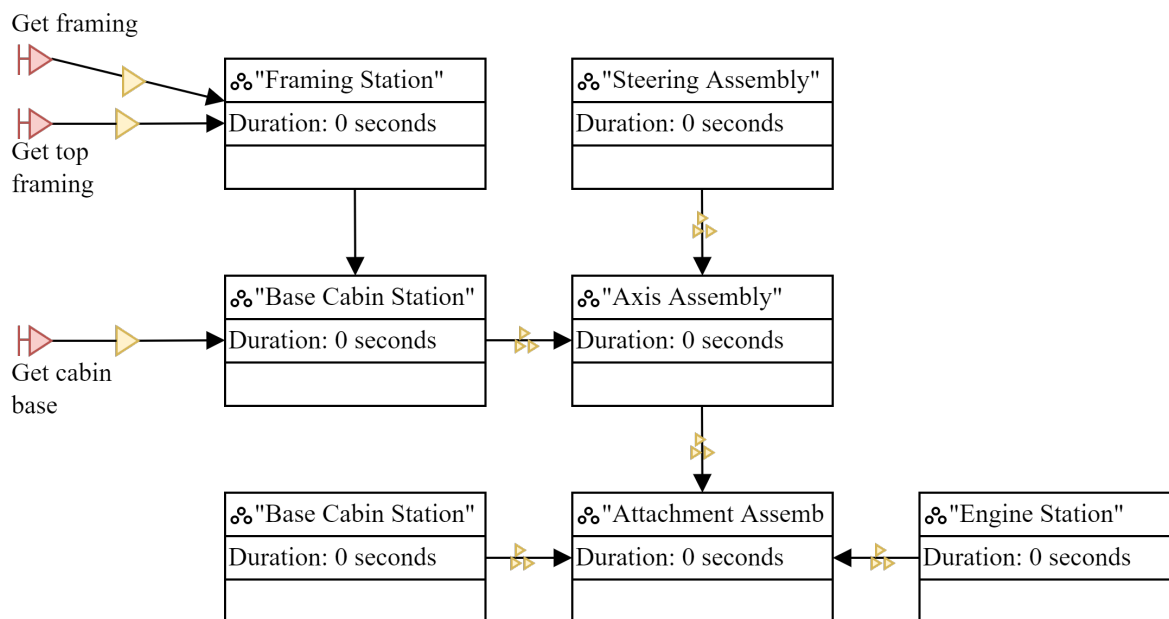
3D grafickém prostředí, vyhodnocení času, nákladů, vytížení zdrojů, logistiky, validace a hrubé optimalizace plánované výroby.

Konkrétní práce s programem probíhá tak, že se nejprve importuje kusovník a 3D modely dílů a. Následně se z knihovny prostředků vytvoří 3D model budoucího pracoviště včetně modelů strojů, robotů, přípravků, upínek, oplocení, lidí, dopravníků, kontejnerů, přepravek atp. Vytvoří se tedy kompletně 3D layout jako podklad pro následné plánování. V případě, že již vytvořené prostorové uspořádání z nějakého důvodu v pozdějších krocích přestane vyhovovat, je možné kdykoli provést jeho úpravu.

Při návrhu nových (nebo optimalizaci stávajících) procesů je nutné si ujasnit, co je potřeba k jejich realizaci - co se bude vyrábět, čím a kde. Následně je pak možné všechny tyto informace propojit v diagramu typu PERT (viz obr. 2.14). PERT diagram pak znázorňuje procesní mapu, na které je vidět posloupnost operací a zdrojů, které do jednotlivých procesů přicházejí. Jsou zde znázorněny i produkty (jednotlivé díly a podsestavy) které proces opouštějí. Pomocí tohoto diagramu má projektant možnost vidět celkové využití zdrojů a časovou náročnost výrobního procesu a podle potřeby může vytvořit varianty prostorového uspořádání (layoutu výrobního systému).

Komplexní popis vzájemných vazeb mezi produkty, procesy a zdroji, včetně prostorového uspořádání výrobního systému napomáhá předcházení chyb a poskytuje detailní pohled v počátečních stádiích na celý výrobní cyklus.

Jednotlivé definované výrobní procesy lze následně několika způsoby spojovat do sekvencí s pomocí diagramu typu Gantt, vytvářet jejich různě složité topologie, a ty následně využívat v dalších softwarových produktech, které se zaměřují detailněji na simulační stránku celé digitální továrny. Další systémy v rámci portfolia Tecnomatix slouží například pro simulace montážních operací, robotických linek či operací, které realizují živí operátoři [101].



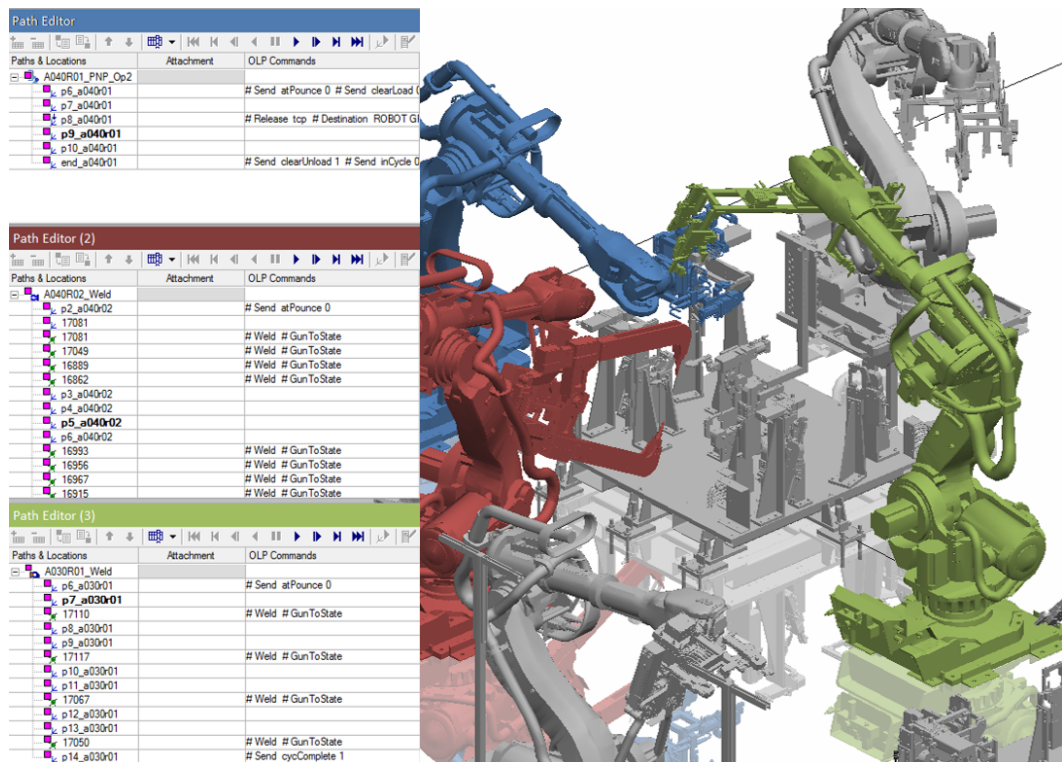
Obrázek 2.14: Pertův diagram znázorňující propojení zdrojů, produktů v procesech

2.7.3 Detailní simulace výrobního procesu

Nástroje portfolia Tecnomatix představují komplexní řešení digitální továrny. Dopusud byl zřejmě popsán software *Process Designer*, který slouží k vytváření layoutů výrobních linek a definování výrobních (ale např. i montážních) procesů. Na tento nástroj jsou dále navázány další softwarové nástroje se sjednocujícím názvem *Process Simulate*⁸ [101]:

- Velkou část digitální továrny představují nástroje pro projektování, simulace a verifikace robotických a automatizovaných výrobních linek. V software *Tecnomatix Process Simulate Robotic* se ověří proveditelnost, prostor, přístupnost, dosah, kolize a další parametry robotického pracoviště včetně off-line programování a virtuálního zprovoznění. Funkčnost reálného nebo emulovaného PLC automatu je tak možné nejprve ověřit napojením na digitální model robotické/automatizované linky a teprve po virtuálním zprovoznění jej pustit do reálné výroby.
- Továrna většinou nezahrnuje pouze linky, na kterých probíhá vlastní výroba jednotlivých součástí. Často je nutno provádět montáže dvou či více komponent dohromady. V rámci software *Tecnomatix Process Simulate Assembly* je možné tyto montážní postupy naplánovat, analyzovat, případně navrhovat a verifikovat optimální cesty jednotlivých dílů, odhalovat a řešit kolize. To vše přispívá k tomu,

⁸V současnosti se jedná o jedinou aplikaci, dříve však byla funkcionality rozdělena do samostatných SW balíčků. Proto jsou také v následujícím textu jednotlivé oblasti popisovány do jisté míry odděleně.



Obrázek 2.15: Process Simulate Robotic [101]

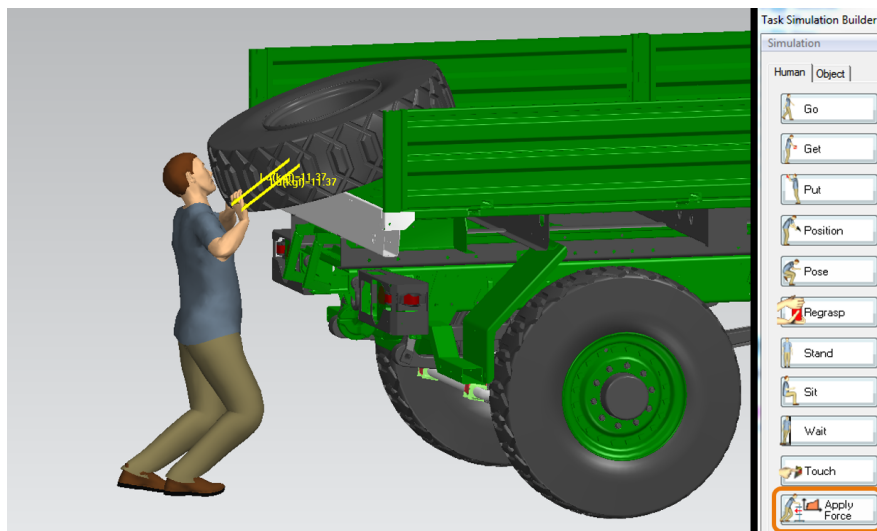
že montážní operace budou efektivní, ale zejména proveditelné⁹.

- Montáž ve většině případů obsahuje vysoký podíl manuální práce a přináší s sebou rizikové faktory, mezi které patří především nepřírodné polohy těla, daleké dosahy při montáži dílů, vysoká opakovatelnost pracovního cyklu (často kratšího než 1 minuta) a velkých sil při instalaci komponentů. Česká legislativa (např. nařízení vlády 361/2007 Sb.) vyžaduje po zaměstnavateli, aby kontinuálně vyhodnocoval ergonomická rizika a chránil zdraví a bezpečí svých pracovníků [15].

V systémech PLM předních výrobců jsou software (v našem “oblíbeném balíku” je to pak *Tecnomatix Process Simulate Human*), které pomocí biomechanicky přesného digitálního modelu člověka vyhodnotí, jestli manuální operace budou proveditelné a ergonomicky přijatelné a zda budou v souladu s ergonomickými standardy a legislativou. Počítačová simulace lidského faktoru umožní analyzovat interakci člověka a pracoviště a vyhodnotit podobu a rozložení pracovního místa, montážní výšky, dosahy, prostor, viditelnost, časové trvání manuálních operací, přijatelnost pracovních poloh, síly a zatížení, manipulaci s břemeny, biomechanické přetížení a zdravotní rizika související s pracovní činností. Zohlednění lidského faktoru pak vede k maximální produktivitě, kvalitě a bezpečnosti pracovního procesu.

⁹Jistě si vzpomenete na nějakou drobnou opravu, kterou jste dělali svépomocí na svém autě – například výměnu žárovky na předním světlometu – častokrát jsou to krkolonné manipulace v malém prostoru, kam není vidět. Takovým věcem je žádoucí se v průběhu výroby vyhnout (nehovoříme již o servisu, ten samozřejmě zákazník platí zvlášť) a nastavit všechny procesy a jejich vzájemné pořadí tak, aby výsledný efekt byl co nejlepší.

Nástroj dále umožňuje realizovat časové rozklady činností a jejich rozřazení na jednotlivé operace. Následně je možné určit, které operace v průběhu výrobního procesu **přidávají hodnotu**, a které ne. Je nutno eliminovat co nejvíce, nebo alespoň zkrátit dobu operací, které hodnotu nepřidávají. K časovým analýzám se používají metody MTM a MOST, které jednotlivé operace (zejména operace prováděné živými operátory) rozdělují na sekvence elementárních úkolů (typu sáhni, uchop, umísti, běž) a každý takový úkol ohodnotí definovanou dobou trvání. Součet všech potřebných časů úkonů pak samozřejmě dává čas potřebný pro realizaci operace. S pomocí uvedených časových rozborů je možné také relativně přesně simulovat průběh výroby a návaznost jednotlivých úkonů na sebe.



Obrázek 2.16: Process Simulate Human [101]

Možností, které skupina nástrojů *Tecnomatix Process Simulate* nabízí, je (téměř) nekonečné množství a jejich popis dalece přesahuje rozsah tohoto dokumentu.

2.7.4 Případová studie - Škoda Auto

Firma Škoda Auto byla již v tomto dokumentu zmíněna v souvislosti se správou životního cyklu produktů. V rámci strategie digitalizace bylo vedením Škody definováno několik strategických iniciativ pro podporu růstu podniku:

- Sdílení a opětovné využití modulů mezi více typy automobilů, a to nejen v rámci Škody, ale napříč celým koncernem VW.
- Zkrácení výrobní doby (času potřebného na výrobu jednotky). To je důležité zejména při zavádění nového modelu do stávající již provozované linky. Škoda chce minimalizovat prostoje pracující výrobní linky, které vznikají přijetím nového či upraveného zařízení.

- Optimalizace vývoje a výroby, tedy maximalizace návratnosti investic.
- Vytvoření nových závodů a výrobních linek v zámoří. Díky snadné přenositelnosti digitálních simulací se tento proces stává levnějším.
- Komunikace se skupinou VW a společné rozvíjení a uplatňování skupinových norem. Vzhledem k tomu, že ŠKODA je součástí koncernu, je nutné, aby se podílela na udržování jednotného přístupu ke společným směrnícím a normám – jak vnitřním, tak i globálním.

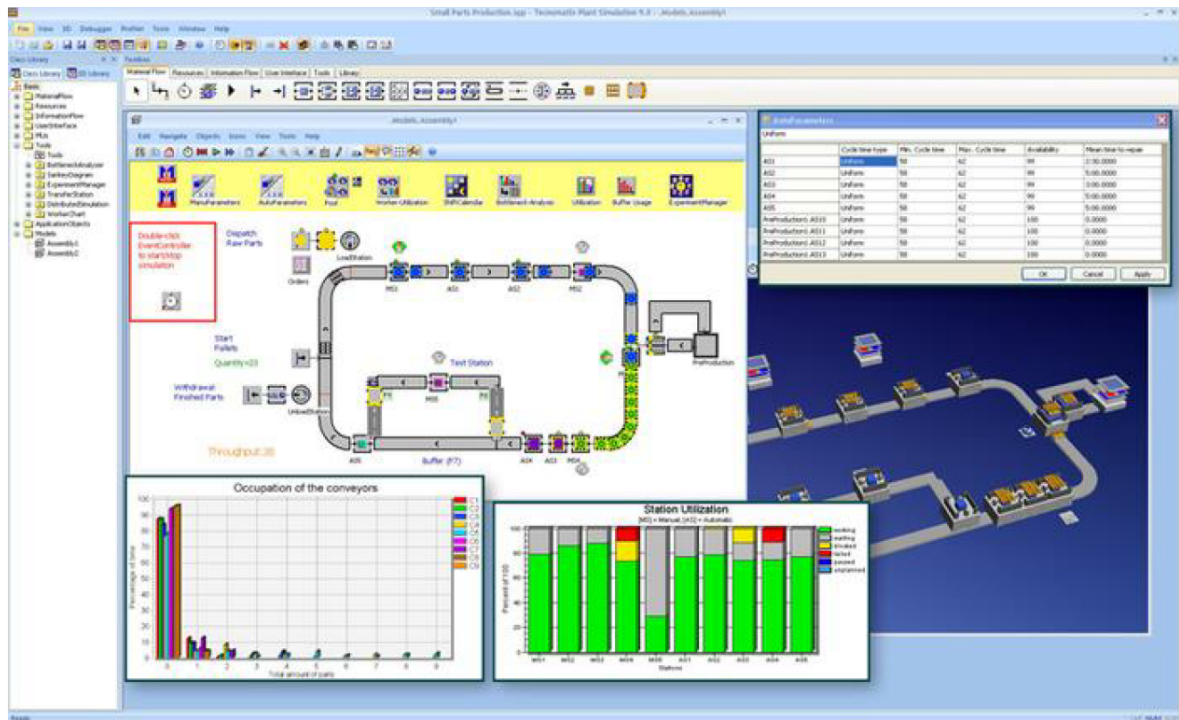
Tyto strategické iniciativy vedly ve firmě k založení týmu digitální továrny, který definoval několik projektů digitalizace a ty postupně po krocích zaváděl. Vzhledem k tomu, že bylo nutné zavést výrobu tohoto modelu do stávající linky tak, aby byla navýšena její výrobní kapacita, týkal se první projekt s Process Simulate simulace svařování bočního panelu u modelu Yeti. Následně byla vybrána druhá linka vhodná pro virtuální provoz a zprovoznění pomocí Process Simulate – svařovací buňka podvozku Yeti zahrnující 5 robotů a zabezpečení proti selhání. Zde Škoda vyzkoušela nejen simulaci pomocí Process Simulate, ale také virtuální zprovoznění s pomocí programovatelných automatů Simatic. Další projekty už následovaly v rychlém sledu.

Vzhledem k tomu, že práce v automobilové výrobě a montáži přináší rizikové faktory pro zaměstnance (práce v nepřírodných pracovních pozicích, vysoké úrovně repetitivnosti pracovního cyklu (při periodě často kratší než 1 minuta) a intenzivní působení sil při instalaci komponentů), rozhodla se Škoda po několika srovnávacích studiích experimentovat a ověřit ve výrobě i nástroj Process Simulate Human. Pohyby pracovníka byly snímány pomocí speciálního obleku (tzv. Motion Capture založeného na snímání pohybů lidského těla pomocí senzorů zrychlení, intenzity magnetického pole a orientace v prostoru [15]). Pořízená data byla přenesena právě do nástroje Tecnomatix Process Simulate Human. S jeho pomocí pak byla řešena ergonomie již v raných fázích návrhu výrobku a plánování výroby. V průběhu roku 2012 byl software ověřen při testování předmontážní linky na výrobu dveří. Bylo potvrzeno, že zaznamenané pohyby jsou téměř stejné, jako pohyby fyzické, a pořizovaná data proto měla vysokou hodnotu pro další zpracování. Na základě těchto výsledků byla vybudována laboratoř ergonomické virtuální reality, ve které vznikají, ve spolupráci s dalšími odděleními, postupy a doporučení směřující k lepší ochraně zdraví a vyšší efektivitě pracovníků.

2.8 Digitální dvojče výrobního procesu

Pokud existuje digitální dvojče výrobku, výrobního zařízení a výrobní linky, jistě je možné posunout se ještě o krok výše v hierarchii a vytvořit s pomocí nástroje *Tecnomatix Plant Simulate* digitální dvojče celé továrny. Tento nástroj slouží pro simulaci a optimalizaci průchodu zboží skrz libovolný logistický systém. Umožňuje vytvořit strukturovaný hierarchický model výrobních závodů, linek, procesů, dopravy a vizualizovat tok skrz velmi složité systémy.

Princip funkčnosti nástroje spočívá ve virtualizaci celé výroby. Ta je postavena jako model tak, aby bylo možné poznávat její chování lépe než v reálném světě. To umožní s celým výrobním cyklem provádět nejrůznější experimenty, testovat různé typy scénářů typu *co se stane, když udělám či změním toto, jaký to bude mít vliv na celkovou produkci, na náklady, na rychlost produkce*, atd.



Obrázek 2.17: Tecnomatix Plant Simulate [101]

Prostřednictvím testování variant lze lépe poznat chování systému a na základě získaných informací přistoupit k nejrůznějším optimalizacím - ať se jedná o optimalizaci transportních dávek, lepší vytíženost strojů, lidí, zvýšení průchodnosti odstraněním úzkých hrdel (tzv. bottleneck), snižování míry rozpracovanosti, apod.

Obecně je možné nástroj použít pro linky – výrobu, která již funguje – vymodelovat to co již existuje, najít lepší řešení, lepší uspořádání a následně implementovat tu změnu do reality, nebo pro výrobu, která ještě neexistuje, kdy se podle požadovaných vlastností a informací linka nejprve vymodeluje, odladí, identifikuje její chování a případné chyby – a až je vše připraveno, teprve poté je realizována ve stavu, kdy už bude perfektně fungovat bez nutnosti zásahů a úprav v budoucnosti, které s sebou nesou zastavování výroby.

Mezi přínosy nástroje patří například zvýšení produktivity stávajícího systému o 12 % až 20 %, snížení investičních nákladů při plánování nového systému až o 20 %, snížení zásob a doby průchodu materiálu o 20 % až 60 % a zkrácení náběhu výroby [101].

2.9 Digitální dvojče pro jiné typy výroby

Všechny dosud popsané nástroje a myšlenky byly striktně propojeny s diskrétní výrobou. V procesním průmyslu, tedy při dávkové výrobě se typicky vyrábějí mezi-produkty ve větších dávkách, avšak vzhledem k měnícímu se charakteru následných zpracovatelských procesů, směřujících k diskrétní výrobě bude i zde nutné přizpůsobit se požadavkům na variabilitu a flexibilitu. V této kapitole je uveden příklad využití systému COMOS, který slouží jako globální datové centrum pro inženýring a poskytuje moduly vhodné pro navrhování všech typů výrobních procesů, popisující v hrubých obrysech integrační možnosti jednotlivých modulů během realizace projektu [100]. Výsledkem inženýrského procesu v popsaném systému je kompletní digitální dvojče výrobní technologie, které je možné opět využít napříč životním cyklem pro testování řídicích algoritmů, ladění efektivity výroby, zaškolení obsluhy a mnoho dalších činností.

- **Projektování technologických zařízení:** pomocí blokových diagramů je možné sestavit model výrobního procesu (funkce *Process Engineering*, obr. 2.18) ①. Na základě tohoto modelu lze efektivně vypracovat schémata jednotlivých technologických procesů. Dále je možné definovat v blokových diagramech směry toků materiálu a vztahy mezi jednotlivými procesy. Tím vzniká diagram procesních toků (PFD¹⁰) ②. Objekty umístěné v těchto diagramech jsou automaticky propojeny s důležitými informacemi.

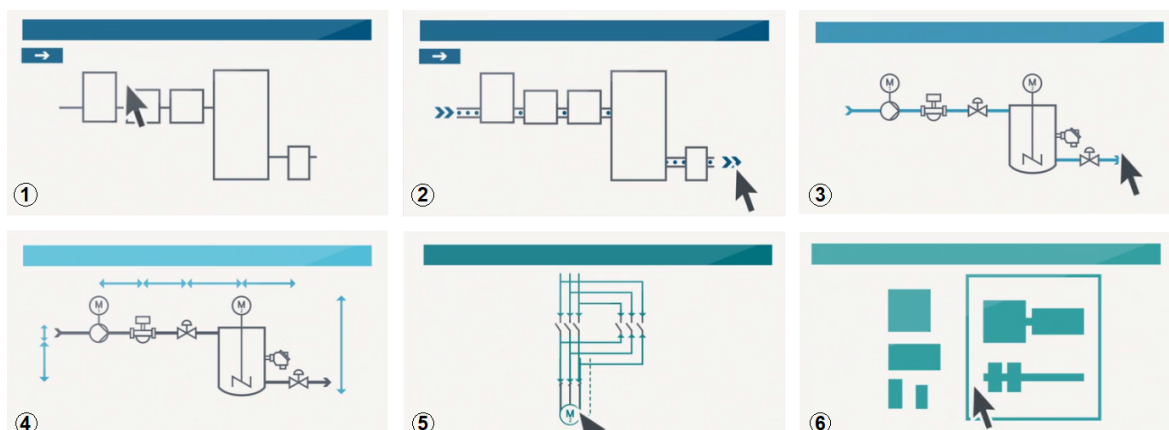
Důležitou vlastností při projektování je možnost sestavit *předběžný hrubý odhad nákladů*, který slouží jako výchozí údaj pro stanovení rozpočtu projektu.

Tato funkce obsahuje vlastní knihovnu bloků, kterou je možné doplnit vlastními prvky a opakovaně je používat. Lze také využívat již dříve vytvořená schémata technologických procesů a upravit je podle nároků na nový projekt.

- **Tvorba schémat P&ID:** diagram procesních toků je možné efektivně převést na *schéma procesní instrumentace* (P&ID) ③. Schémata se vytváří pomocí symbolů, které jsou opět umístěny v objektové knihovně a splňují mezinárodní normy. Tato prostředí automaticky rozpoznávají, jakých spojovacích dílů je potřeba, a správně je zařadí podle směru toku média. Poté se již nemusí kontrolovat správnost zařazení těchto dílů do schématu.
- **Návrh a správa potrubí:** používá se pro definování a správu tříd potrubí, definování vzdáleností jednotlivých prvků mezi sebou, a tedy i potřebných délek vedení ④. Při projektování lze využívat předdefinovaných dílů potrubí, které opět splňují standardy. Vybírání dílů pro tvorbu technologického zařízení se orientuje pomocí parametrů, jako jsou tlak, teplota či materiálová omezení (např. použití v potravinářství).
- **Integrovaný návrh technologických zařízení:** nachází uplatnění v odvětví elektroinstalace při návrhu elektrického propojení jednotlivých komponent¹¹

¹⁰Process Flow Diagram

¹¹(EI&C) Electrical Instrument and Control



Obrázek 2.18: Modely vytvořené v systému COMOS [100]

. Návrh vychází z instrumentace, která byla naplánována v rámci předchozích inženýrských kroků (5).

Zobrazení schémat je možné ve dvou režimech, a to v jednopólové nebo vícepólové podobě. Při tvorbě schématu je prvek definován jen jednou a nezáleží, jaké zobrazení je aktivováno, protože prvky mají definovány atributy, které odpovídají všem zobrazením, specializacím a oborům. Lze také vytvářet a ukládat šablony a vzory, což šetří čas a zvyšuje kvalitu.

Informace o napětí se odvozují automaticky z napětí napájecích sběrnic, se kterými jsou převzata i jejich označení. Normované zobrazení je možné přepínat mezi různými standardy, a tak není potřeba manuálního převodu dat. Opět jsou k dispozici knihovny, odpovídající mezinárodním standardům. Tyto objekty, definované v knihovně, lze přizpůsobovat individuálním potřebám a případně vytvářet objekty nové.

Dalším krokem je návrh rozvaděče pro elektronické komponenty (6). Pro jednotlivé přístroje a sítě lze stanovit zatížení a výkon. Pokud dojde k překročení maximálního výkonu, jsou tyto části vizuálně znázorněny. Při tvorbě rozvaděče lze mít náhled ve 2D nebo 3D, přičemž systém automaticky hlídá např. stavební rozměry, nutný hloubkový a výškový odstup zařízení, apod.

Dále probíhá automatické propojení podle stanovených kritérií, výpočet potřebné délky vodičů a jejich dimenzování. Mimo elektrické součásti a spoje jsou ve výkresech k dispozici data otvorů rozvaděče, která mohou být dále přímo předána řídicím jednotkám vrtaček a fréz.

Samozřejmostí je možnost exportu požadovaných délek propojovacích vodičů v rozvaděči, seznamu potřebných komponent a mnoha dalších datových pohledů. Při každé provedené změně se tyto dokumenty automaticky aktualizují.

- **Propojení s vývojem software:** Data týkající se hardwarové konfigurace PLC automatů jsou díky propojením s automatizačními systémy (např. SIMATIC nebo PCS7) automaticky poskytnuta těmto systémům, a tak není potřeba je znovu definovat. To samozřejmě kromě zásadního urychlení práce také eliminuje

chyby vzniklé ručním přepisováním množství dat. Propojení s dalšími nástroji, jako je SIMIT, umožňuje zkrácení času tvorby funkčního fyzikálního modelu potřebného pro virtuální zprovoznění technologie.

- **Simulace a virtuální zprovoznění:** je možné vytvořit kompletní 3D digitální model technologie. V tomto modelu je pak možné se virtuálně procházet. Při této virtuální inspekci je možné odhalit chyby v návrhu technologie, zejména vedení potrubí, nedodržení předepsaných vzdáleností, apod. Model je rovněž možné využívat například pro zaškolování nových zaměstnanců výroby či údržby. Také je možné jej využít pro virtuální zprovoznění.
- **Digitální dvojče:** kompletní model slouží i dále - během provozu jako digitální dvojče. Systém udržuje pravidelnou výměnu procesních dat s řídicím systémem technologie a umožňuje provádět simulace a optimalizace výrobních procesů.
- **Management servisu a údržby:** slouží pro archivaci a správu všech dat významných pro údržbu a servis. Umožňuje přidělovat obsluhujícím pracovníkům kvalifikace a pracovní úkony. Tím lze ušetřit finanční prostředky a úkony optimalizovat.
- **Plánování a provedení provozních odstávek výrobního zařízení:** umožňuje přesnou a jistou komunikaci při plánování odstávky zařízení. K tomu jsou využívána data z předešlých odstávek, mezi které patří doba odstávky, řešené problémy, vzniklé nečekané události, atd.

Příkazy a postupy pro servis, údržbu nebo odstávku mohou být nahrány do mobilního zařízení. Pracovník je má tudíž neustále k dispozici, což sníží riziko chyb a usnadní práci s vyplňováním servisního protokolu.

- **Správa dokumentů:** Je potřeba, aby všechny dokumenty týkající se daného výrobního zařízení byly vždy a jednoduše přístupné. Jedná se převážně o materiály vzniklé v jiných aplikacích než COMOS, a nebo o podklady dodané výrobcí (např. specifikace přístrojů). Jelikož jde o data, která vznikla mimo COMOS, nemusí v něm být kompletně integrovaná, a proto je nutná jejich správa. Tu zajišťuje tato funkce, která zaručuje spolupráci s běžnými kancelářskými programy, archivaci kdo, kdy a co nahrál. Umožňuje schvalování dokumentace a export do PDF, i zde jsou k objektu připojena data o historii, autorovi, kontrolních krocích a schvalujícím zaměstnanci [100].

2.10 Digitální dvojče mimo průmysl

Nasazení digitálního dvojčete se neomezuje jen na průmyslovou výrobu, ale je dnes běžné i v jiných odvětvích a oblastech.

Stavebnictví

Je zcela přirozené, že stejně jako vznikají konstrukční modely strojů, zařízení a vlastně i samotných produktů, vznikají ve stavebním sektoru konstrukční modely budov, mostů, přehrad, či dalších. Využití digitálního dvojčete se ani v této oblasti neomezuje jen na rané fáze životního cyklu (využití konstrukčních modelů pro simulaci různých pevnostních charakteristik)¹². Uvažujme například ocelový most, u kterého trvalé měření dat v reálném čase společně s fyzikálními modely umožní nejen nepřetržité sledování různých fyzikálních parametrů, ale také analýzu chování a různé simulace. Bude tedy možné v čase předpovídat budoucí stav celé konstrukce. Na základě získaných údajů je možné efektivně plánovat údržbu mostu a udržet tak celý systém provozuschopný a splňující bezpečnostní kritéria.

Zdravotnictví

I Zdravotnický sektor těží z nástupu moderních technologií, které umožnily zmenšení, snížení ceny a zjednodušení implementaci různých měřicích a diagnostických zařízení.

Ve zdravotnictví je v důsledku aktuálních celospolečenských změn velký nedostatek pracovníků. Vůbec největším problémem je dnes zajištění péče o seniory, kteří potřebují téměř nepřetržitý dohled, ačkoli jsou jinak zdraví a mobilní. Pro tyto osoby, ale i pro celý zdravotnický systém je ideální, aby trávili čas doma, pokud je to alespoň trochu možné. Proto vzniká nespočet různých aplikací, které monitorují činnost těchto osob – od detekce pohybu osoby mezi místnostmi bytu, až po analýzu pohybů během spánku. Tyto systémy, založené nejčastěji na internetu věcí, komunikují pořízená data do centrálního úložiště, odkud je následně zpracovává digitální dvojče. Výsledkem jsou informace o kondici dané osoby. Systémy samozřejmě umožňují vyvolání okamžitého alarmu, pokud vyhodnotí nestandardní stav – např. pád osoby na zem, dlouhé ležení v posteli bez přetočení apod. Uvedené systémy pak mají tu výhodu, že péče o osobu může být rozdělena mezi blízkou osobu a pečovatelskou službu – systémy např. pomocí webového rozhraní informují dispečera pečovatelské služby o stavu celé skupiny pacientů. Pokud dojde k nepředvídané události, může být např. pomocí SMS informována okamžitě také blízká osoba, která je schopna provést kontrolu rychleji.

Jiným příkladem nasazení digitálního dvojčete v medicíně může být modelování procesů probíhajících v lidském těle a simulace efektu, užívání specifických léků.

Smart Cities

Smart Cities jsou společenství lidí komunikující a používající toky energie, materiálů, služeb, financí, dat a informací s cílem urychlit udržitelný hospodářský rozvoj, stabilitu a vysokou kvalitu života. Toky a informace se stávají *chytrými* prostřednictvím

¹²Modely bývají dále často používány opakovaně pro další zakázky

strategického využívání informačních a komunikačních infrastruktur a služeb v procesu transparentního územního plánování a řízení, které je citlivé vůči sociálním a ekonomickým potřebám společnosti. Pro realizaci Smart Cities je charakteristické využívání nejmodernějších metod informačních technologií, např. cloud computing, geocomputation, prostorové analýzy, modelování, simulace, atd [23].

Transformace současných měst do moderních smart cities je založena na realizaci celé řady koordinovaných projektů a integrace procesů směřujících k udržitelnému rozvoji společnosti a zvyšování kvality života občanů. Jednotlivá řešení musí usilovat o minimalizaci použitých zdrojů a zároveň o maximální efektivitu. Na úrovni města zpravidla celou transformaci koordinuje, koriguje a na dodavatelské firmy dohlíží příslušný odbor magistrátu či městského úřadu. Pokud jsou do celého procesu zapojeny lokální dodavatelské firmy spolupracující navíc např. s provinčními vysokými školami, dochází ke vzniku nových pracovních nabídek, což přináší pozitivní efekt v podobě menšího odlivu mladých pracujících do větších měst.

Na úrovni moderního města existuje mnoho oblastí, ve kterých je vhodné nasazení moderních technologií, mezi jinými i digitálních dvojčat. Mezi ty nejviditelnější můžeme řadit dopravu a životní prostředí [9], např.:

- řízení a zklidňování dopravy - měření a modelování stavu dopravní situace a nasazení řešení pro zvýšení plynulosti a bezpečnosti silničního provozu bez nutnosti budování infrastrukturních staveb.
- tlak na omezení objemu individuální motorové dopravy a její náhrada za udržitelné dopravní způsoby (prioritizace veřejné dopravy);
- podpora obnovitelných zdrojů energie, optimalizace distribuce i spotřeby elektrické energie;
- plánování budoucí výstavby měst s ohledem právě na snížení znečištění ovzduší jak v nových čtvrtích, tak ve stávajících (např. kvůli zvýšené intenzitě dopravy).

Pro všechny uvedené případy platí, že úspěšné řešení projektu v dané oblasti se opírá o vytvoření přesného modelu. Tento je vytvářen z údajů, jako jsou územní plány, data provozovatelů hromadné dopravy, dopravní generely, dotazníková šetření, a samozřejmě sběr dat sítěmi snímačů, který otevírá možnosti využít statický model jako digitální dvojče systému. V případě městské infrastruktury je nedocenitelnou možností provádění simulací různých scénářů chování.

V posledních letech se směr vývoje v oblasti chytrých měst postupně přesouvá. Zatímco dříve byl kladen důraz na vývoj technického řešení prostředků sběru dat, jejich integraci do městské infrastruktury a zabezpečení dlouhodobého sběru dat, dnes, se zdokonalováním prostředků umělé inteligence vývoj směřuje k integraci dat z více oblastí a nasazení pokročilých algoritmů pro jejich zpracování. Potenciál využití konceptu digitálního dvojčete pro chytrá města se tak postupně zvyšuje [19].

2.11 Shrnutí kapitoly

V rámci této kapitoly byly představeny základní koncepty teorie životního cyklu produktů a vhodných způsobů jeho řízení. Dnešní moderní softwarové platformy pro řízení životního cyklu produktů slouží jako datová centra pro navazující modelovací a simulační aplikace.

V raných fázích životního cyklu bývá v současnosti vytvořen kompletní digitální model produktu, na základě kterého lze následně provádět množství simulací, či ověření výrobních postupů. Kapitola dále jmenuje dva důležité koncepty - digitální stín a digitální dvojče, které je moderním paradigmatem používaným pro zvýšení efektivity v průběhu celého životního cyklu typů i instancí produktů.

Kapitola 3

Průmysl 4.0?

Pokusme se popsat Průmysl 4.0 na základě znalostí o tom, kde jsme v oboru průmyslové automatizace dnes, a kam se chceme podle představ posunout v následujících letech. Předem je nutno zmínit, že k označení pojmu Průmysl 4.0 je nejpřiléhavější slovo iniciativa. Iniciativa, která má zafungovat jako spouštěč k běhu na dlouhou trať a zahrnuje téměř vše, čeho bylo v současnosti v oblasti průmyslové automatizace dosaženo. Dotýká se všech *typů průmyslových výrob* (spojité, kusové i dávkové), všech *průmyslových sektorů, sociálních a environmentálních* aspektů průmyslu, všech entit, které se v průmyslu vyskytují (např. pojem internet věcí zahrnuje mezi *věci* výrobek ve všech stadiích jeho vývoje a výroby, výrobní nástroje (SW i HW), modely, celé linky, továrny, atd.). Něco tak heterogenního pod jedním názvem snad odbornou průmyslovou veřejnost ještě nikdy nevyzvalo ke společné cestě. Je to skutečně spíše cesta, než cíl, je však poněkud vágní a snad i neznámý, otevřený. Je to vize, pro někoho až nepředstavitelná, ale i ty největší dosažené technické úspěchy byly původně pouze odvážnými představami.

Kdybychom se probírali historií technických vynálezů a objevů aplikovaných v průmyslu od 70. let, bylo by to na několik stran. Mezi mnoha hardwarovými řešeními se vyskytuje překvapující počet různých těch softwarových.

Vždyť jen v předchozí kapitole jich bylo popsáno dost, a to se jednalo pouze o jednu úzkou oblast zaměření a portfolio jednoho výrobce. Za poslední dvě dekády jich vzniklo tolik softwarových nástrojů, že se nabízí otázka: “proč tolik softwarů, systémů, platforem, když nakonec výrobní linku řídí PLC?” Z praxe víme, že vytváření hodnot s sebou nese spoustu neproduktivních činností - režii. Dobrý podnikatel se snaží tuto režii redukovat a zbavit balastu. Je to velký problém, protože necitlivý přístup zde může přivodit až kolaps výroby. Každý běžný produktivní pracovník musí dělat neproduktivní činnosti související se životem jako najíst se, nakoupit oblečení, zaplatit faktury, apod. Ve výrobě to obecně nazvěme manažerské činnosti.

Na obrázku 3.1 jsou uvedeny příklady činností, které je potřeba vykonávat proto, aby výrobní nebo zpracovatelský závod nějakým způsobem fungoval a vůbec mohl vyrábět. Za každým zeleným okénkem můžeme vidět samostatné oddělení a co okénko, to režie. Naštěstí reálná výroba ještě není tak sofistikovaná, takže těch oddělení podnik



Obrázek 3.1: Přehled manažerských funkcí platformy Teamcenter [101]

ve skutečnosti nepotřebuje tolik. Znamenalo by to mít značný počet manažerů (lidí, rozhraní, SW agentů). Pak je zde druhá možnost, mít jednoho úžasného, mocného manažera, který všechny tyto činnosti ohlídá a bude je monitorovat a spravovat. Jeho mozek a jeho stůl tvoří platformu, kterou je Management životního cyklu výrobku neboli PLM [101].

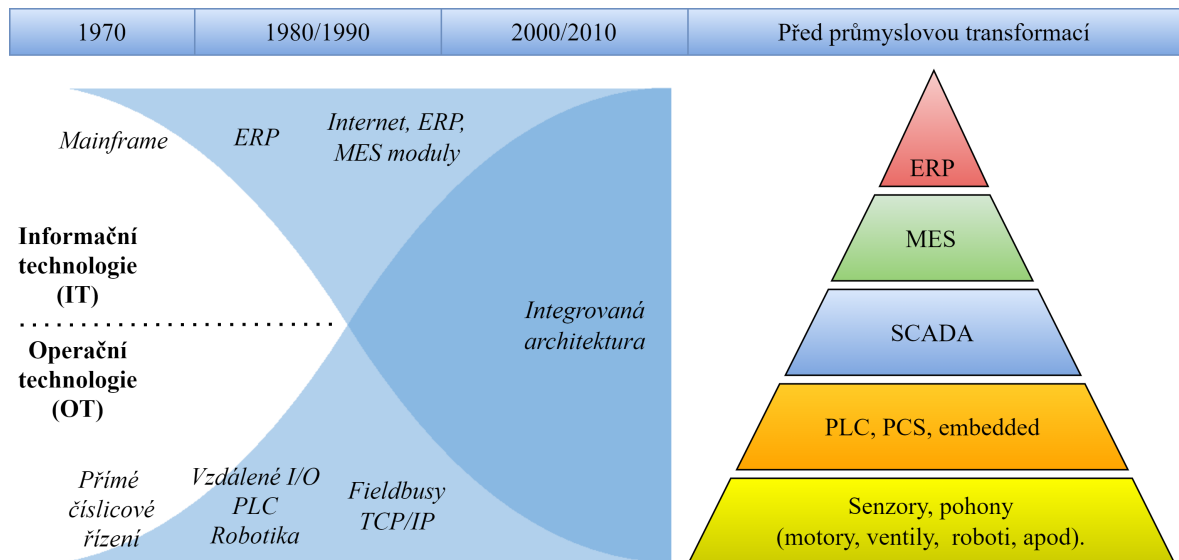
Konvergence IT a OT

Vraťme se opět k již dříve uvedenému příkladu výroby aut. Každý jeden díl auta, kterých jsou tisíce, ponese informaci o materiálu, postupy a průběh výroby, místa montáže, atd. To je nevídané množství informací, se kterým jsme se dosud v oboru kybernetiky nikdy předtím nesetkali. Tyto informace vyžadují nové typy databází, metody dolování a zpracování dat, datové aplikace, inženýrské *frameworky*, speciální platformy a také nové typy komunikační infrastruktury. Je to oblast, se kterou nemáme v našem oboru mnoho zkušeností.

Podívejme se ale nyní znovu na pyramidu výrobního podniku. Historicky vzato pracovníci IT úspěšně razili cestu, ke které se posléze připojili průmysloví kybernetici (OT – operační technologie).

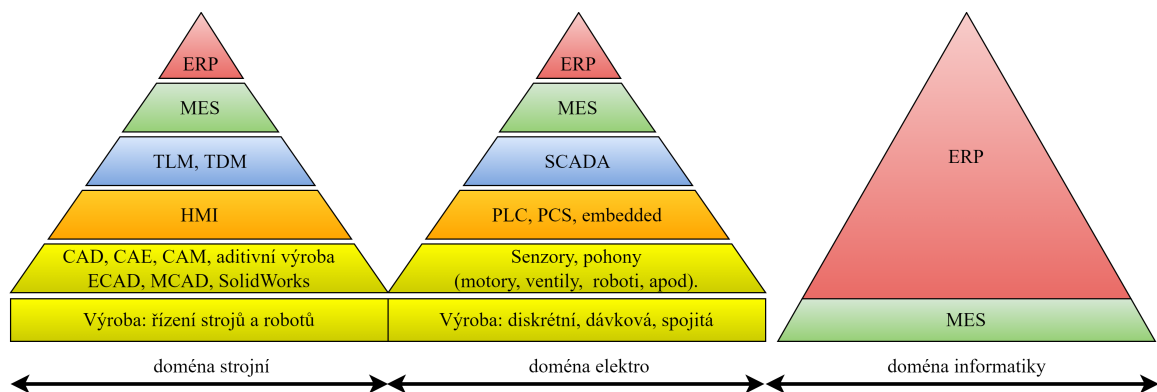
Jakkoli se pro laika mohou zdát podobné, IT a OT jsou dvě rozdílné skupiny v rámci podniku, které mají rozdílný původ, cíle, pohledy a zájmy. Pracovníci OT jsou ve svých profesích mnohem blíže k výrobním procesům, pracovníkům *modrým límcům*, se kterými často spolupracují jako součást jednoho týmu. Na těchto pracovnících leží veškerá odpovědnost za hladký průběh výroby a tito pracovníci si to dlouhodobě uvědomují. Technické prostředky, které se v OT využívají, se vyvinuly jako průmyslové deriváty klasických PC (ve stopách IT) a jsou to především programovatelné automaty PLC, jednotky s vestavěnou mikroprocesorovou elektronikou (MCU) a distribuované řídicí systémy DCS, v naprosté většině používající real-time OS a využívající komunikaci na bázi průmyslových Ethernet řešení [39].

Naproti tomu pracovníci IT bývají většinou více vzdáleni od reality výroby, neboť výkonem své profese se přímo nepodílejí na zisku nebo ztrátě. Jsou mnohem blíže vrstvě vedení podniku, využívají své odborné znalosti k metodickému vedení ostatních oddělení k používání efektivních metod a technologií. Jejich práce ale více



Obrázek 3.2: Konvergence IT a OT, dle [122]

připomíná vedení než spolupráci. Technologie IT zahrnují počítačové systémy, standardní komunikační sítě typu Ethernet, platformy, *cloud*, používají *time-sharing OS* a zabezpečují a spravují funkce obchodní a finanční i spolupráci s podnikovými partnery. Z odděleného pohledu IT a OT je možné říci, že existují dvě *pyramidy* popisující hierarchicky zaměření v těchto dvou oblastech. Vzhledem k tomu, co již bylo uvedeno v kapitole 1.4.1 můžeme počet oddělených pyramid bez obav rozšířit na *tři* (viz obr. 3.3).

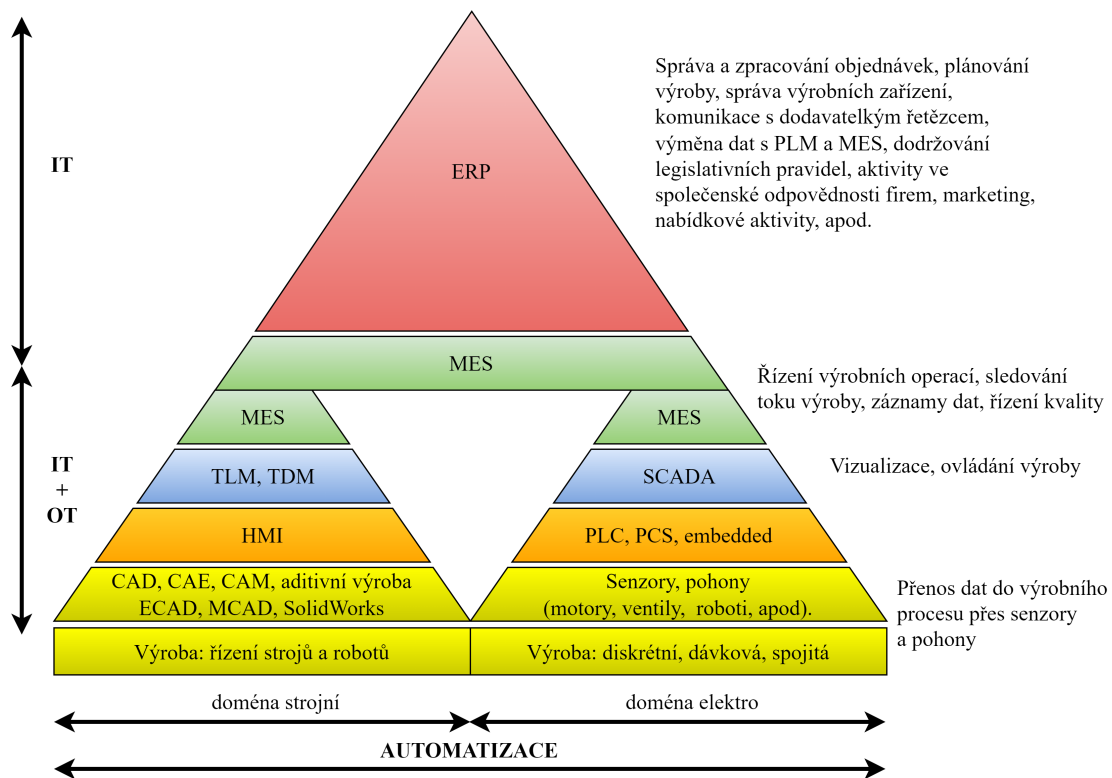


Obrázek 3.3: Rozdělení automatizace na 3 domény

Stejně tak, jak velká byla historická mezera mezi informačními a operačními technologiemi, panuje mezi IT a OT skupinami pracovníků odloučenost. Z hlediska odborného zaměření můžeme říci, že v Průmyslu 3.0 je IT doménou programátorů inženýrů a OT je doménou programátorů kybernetiků. Vzhledem k faktu, že jedním z typických znaků Průmyslu 4.0 (a vlastně naprostou nezbytností) je přibližování IT a OT - technologická konvergence (stále více technologií z IT se využívá také v doméně OT a oba technologické světy jsou stále více propojené). Na úrovni podniku je to však komplikovaný proces vyžadující volbu vhodných strategií [39] a probíhá na mnoha úrovních - od zajištění interoperability jednotlivých zařízení a jejich propojení, po systémovou

a procesní integraci. Postupně také vznikne nový typ inženýrských pracovníků, kteří se rekrutují z IT nebo OT a přitom absorbují dostatek znalostí a zkušeností z toho druhého oboru tak, aby byli schopni uvedené činnosti plánovat a realizovat.

Vize Průmyslu 4.0 se má týkat všech sektorů výrobního průmyslu a slučování IT a OT bude představovat synergii velkého množství heterogenních postupů, metod, funkcí a systémů z obou oblastí, první přiblížení tohoto jednotícího procesu můžeme znázornit sloučením tří pyramid (obr. 3.4).



Obrázek 3.4: Výchozí bod k transformaci 4.0 (vlastní zpracování)

Pyramida na obr. 3.4 zachycuje epochu automatizace, která reprezentuje startovací bod pro vizi Průmyslu 4.0. Z této pyramidy se dají predikovat znalosti, které by měl mít řešitelský tým, pracující na nějakém projektu v intencích Průmyslu 4.0, a které jsou popsány dále. Předpokládejme zatím zachování statu quo v rozdělení automatizace typů průmyslových výrob, tzn., že strojírenství (obrábění, tváření) zůstane v zájmu strojírenského inženýrství.

3.1 Hlavní aspekty průmyslové integrace

Koncept Průmyslu 4.0 je založen na hluboké **průmyslové integraci** prostřednictvím informačních technologií a s ní spojeném zpracování dat v reálném či takřka reálném čase, sdílení informací a kontinuální komunikaci. Tato integrace má tři základní pilíře:

- Vertikální propojení

- Horizontální propojení
- Integrace všech inženýrských procesů

Podívejme se na tyto pilíře detailněji v kontextu stavu před Průmyslem 4.0.

3.1.1 Vertikální propojení 3.0

Hierarchické propojení jednotlivých zařízení a systémů v rámci současných výrobních podniků 3.0 je možné znázornit ve formě pyramidy, a takto je také přímo či nepřímo součástí několika norem vydaných organizací International Society of Automation (ISA), jmenovitě ISA 88 a ISA 95 (a její mezinárodní obdoby IEC 62264). Dle normy ISA 95 je činnost jednotlivých systémů ve výrobních podnicích možné rozdělit do 5 vrstev, které společně tvoří onu pomyslnou pyramidu (viz obr. ??). K ní je třeba říci, že představuje klasické vertikální uspořádání automatizačních funkcí výroby, a to pro všechny tři základní typy výroby, tj. diskrétní, spojitou, dávkovou.

- Úroveň 0 představuje samotný výrobní proces, tedy jednak postupně vznikající produkt a jednak technické zařízení sloužící pro transformaci vstupního materiálu na produkt či k jeho transportu.
- Úroveň 1 definuje prostředky, které slouží ke snímání stavu výrobního procesu (snímače) a k zásahům do něj (akční členy). Komunikace těchto prostředků na vyšší úroveň byla historicky realizována tzv. Fieldbusy, v dnešní době se často využívá průmyslový Ethernet v různých podobách. Obecně lze říci, že komunikovaná procesní data jsou na této úrovni jen málo strukturovaná, naproti tomu na komunikaci a zpracování jsou kladeny vysoké nároky na včasnost¹ a současnost². Perioda komunikace dat bývá zlomky až jednotky sekund.
- Na úrovni 2 stojí prostředky pro řízení a monitorování procesů, realizované konkrétně PLC, DCS, resp. SCADA systémy. Popsané tři úrovně společně zahrnují samotný výrobní proces a jeho přímé řízení, a to pro všechny typy procesů popsaných v odstavci 1.4 (spojité, dávkové i diskrétní). Komunikace s vyšší úrovní dnes již probíhá prostřednictvím rozhraní Ethernet a většinou klasickými TCP/IP protokoly v časovém rámci jednotek sekund. Pro komunikaci jsou specifikovány a využívány čtyři typy datových toků: ① řízení procesu, ② zjištění aktuálního stavu procesu, ③ přenos konfigurace a ④ přenos výrobních a diagnostických dat.

¹Včasnost reprezentuje požadovanou dobu odezvy řídicího (nebo komunikačního) systému vzhledem k časovým požadavkům a parametrům řízeného procesu.

²Současnost je popsána jako schopnost korelovaně zpracovávat data z více než jednoho vstupu v tomtéž časovém horizontu.

- Úroveň 3 popisuje prostředky pro řízení výrobních operací, v dnešní době tedy téměř *vše zahrnující* platformy, definované v rámci ISA 95 (tzv. *Manufacturing Operation Management* (MOM) se základy v systémech MES. Z hlediska výrobního je možné říci, že z této úrovně jsou nižším úrovním předávány výrobní příkazy ke zpracování a jsou z nich získávána výrobní data. Systémy MES také představují rozhraní mezi řídicími systémy výroby (nižší vrstvy) a podniku (vyšší vrstvy). Strukturovaná komunikace se systémy vyšší úrovně pak již probíhá výhradně prostřednictvím rozhraní Ethernet a je záležitostí např. využívání REST API, databázových komunikací, či přenosu formátovaných souborů.
- Na úrovni 4 pak stojí celopodnikové plánování realizované systémy ERP.

Mezi jednotlivými vrstvami pětistupňové podnikové hierarchie probíhá komunikace, přičemž prostředky se časem vyvíjí a zdokonalují pod tlakem požadavku na stále větší množství dat, které se musí mezi jednotlivými vrstvami přenášet. Způsoby přenosu dat (toky dat) mezi jednotlivými úrovněmi jsou v dnešní době v naprosté většině případů standardizovány, mezi jinými právě např. v rámci ISA 95 (rozhraní mezi MES a ERP úrovní), ISA 88 (rozhraní mezi MES a řídicím systémem v *dávkovém řízení*) či PackML (pro *diskrétní řízení*). Mozkem výrobního podniku je tedy systém ERP, srdcem je MES, oči jsou SCADA/HMI a ruce jsou PLC a systémy pro řízení dávkových procesů BATCH. Problém je, že tyto orgány nejsou umístěny v jednom organismu, ale každý z nich *vlastní jiná entita*. To není optimální, ale díky standardizaci se to dá jednoduše nasadit a úspěšně provozovat.

Vzájemná spolupráce heterogenních systémů

Několikrát již bylo zmíněno, že propojení jednotlivých vrstev a způsoby přenosu dat jsou dány standardy. Předpokladem pro to, aby spolu nějaké systémy mohly spolupracovat, je vybavenost takovým komunikačním rozhraním, které jim umožní spolu určitým způsobem vyměňovat data. Při pohledu na sedmivrstvý ISO/OSI model je v prostředí OT komunikačních protokolů často kladen velký důraz na spolupráci zejména na úrovni linkové vrstvy. V případě IT řešení (většinou prostřednictvím rozhraní Ethernet) pak na vyšší vrstvy, až po tu aplikační. Způsob zpracování dat v *uživatelské* aplikaci (z pohledu výrobce hardware a firmware/operačního systému) se pak často odsouvá na vedlejší kolej, přestože právě vzájemná spolupráce na vyšší úrovni je tím nejvíce komplexním problémem při integraci heterogenních systémů.

Míra vzájemné propojenosti a informační spolupráce mezi systémy se nazývá *interoperabilita* a na aplikační vrstvě se rozlišují dva druhy - syntaktická a sémantická. Zatímco syntaxe se, zjednodušeně řečeno, týká formátu zpráv, sémantika označuje jejich význam.

Syntaktická interoperabilita

Syntaktická interoperabilita je *vyjádření struktury metadat umožňující syntakticky kombinovat datové prvky z různých schémat, slovníků, thesaurů, taxonů a jiných nástrojů* [60]. Této formy interoperability se dosáhne vyznačením dat předem určeným způsobem, takže je možné sdílet data v různých systémech. Existuje několik praktických způsobů, jak syntaktické interoperability dosáhnout:

- **Kódování řetězců** v textově orientovaných protokolech (např. XML, HTML) - při přenosu textových zpráv napříč různými systémy je vždy nutno dbát na použití správného kódování pro nestandardní znaky, nebo alespoň na uvedení použitého kódování v těle zprávy (je přítomno např. u XML)
- **Znalost a uplatnění požadované syntaxe** - z textově orientovaných protokolů můžeme uvést opět např. XML, u něhož je nutno dbát na dodržení korektní (validní) struktury vytvářeného dokumentu, tedy dbát na správné uzavírání otevřených uzlů, správné označování atributů, prostě sledování všech syntaktických pravidel. Pro binárně orientované protokoly, např. ModbusRTU je zase nutné ctít formát zpráv, přesně dodržet strukturu hlavičky (která obsahuje např. adresu protistrany, kód funkce), a způsob kódování vlastních dat (což jsou v tomto případě dvoubajtové hodnoty).

Syntaktickou stránku přenášené zprávy je možné na straně příjemce snadno ověřit procesem validace, při kterém dochází k pokusu o tzv. parsování zprávy, tedy jejího rozkladu. Pro tyto rozklady jsou využívány algoritmy na bázi tzv. lexikálních analyzátorů, které dělí vstupní posloupnost na tzv. lexémy - lexikální jednotky. Tyto lexémy jsou reprezentovány ve formě *tokenů*, které dále zpracovává syntaktický analyzátor. Ten v jednoduchosti ověřuje, že jednotlivé tokeny jsou řazeny v sekvencích, které odpovídají syntaxi jazyka ověřované zprávy.

Sémantická interoperabilita

Sémantická interoperabilita je obsahové vyjádření struktury metadat, které dovoluje sémanticky kombinovat datové prvky z různých schémat, slovníků, thesaurů, taxonů a jiných nástrojů. Umožňuje tak vyhledávat informace napříč heterogenními distribuovanými databázemi zadáním jediného dotazu. Pomocí sémantické interoperability jsou řešeny případy, kdy jednotlivé zdroje používají *různé termíny pro popis téhož pojmu* nebo naopak používají *stejně termíny pro různé pojmy*. Proto je její dodržování důležité zejména v kritických systémech (např. v biomedicíně). Dá se říci, že sémantická interoperabilita je důležitá proto, že zajišťuje pojmovou čistotu při převodu mezi různými prostředími, tj. pracuje s porozuměním odborným termínům [60].

Pro heterogenní výrobní prostředí je sémantická interoperabilita neméně důležitá, než syntaktická. V rámci tohoto dokumentu se obou forem interoperability dotkneme detailněji při popisu tzv. Industry 4.0 komponenty a jejích možností.

Hierarchické řízení výroby

Hierarchická struktura, uvedená na obrázku ?? popisuje kromě integrace jednotlivých systémů ještě něco dalšího. Jsou to tři typy automatizovaného řízení výroby:

- Základní řízení (spojitá regulace, diskrétní regulace, stavové automaty, atd...)
- Procedurální (spouštění jednotlivých funkcí základního řízení vedoucích k výrobě produktu)
- Koordinační (koordinace výroby jednotlivých produktů za účelem maximalizace efektivity výroby)

Když tato tři řízení hierarchicky (vertikálně) propojíme (integrujeme) vznikne dobře fungující automatizovaný systém (pyramida) výroby.

Zavádění prvků Průmyslu 4.0 bude pro mnohé podniky, hlavně pro malé a střední, obtížné a někdy dost problematické. Tomu napovídá fakt, že se tyto společnosti často dosud nevyrovnaly se zaváděním systémů MES, případně Batch. Zní to drasticky, ale důvodem může být, že mnohé, zejména malé firmy vlastně zatím nepochopily podstatu systémů MES a Batch (konkrétní čísla jsou uvedena již v závěru kapitoly 1.1). Nevědí, zda se některý z těchto systémů vůbec pro jejich výrobu hodí, a co jim může přinést. V blízké budoucnosti se jejich rozhodování nebude týkat dvou systémů, ale desítek systémů, přístupů, metod apod. Základem těchto rozhodování vždy bude uvědomit si, jaký systém či přístup se pro daný typ výroby hodí. V kapitole 1.4.1 je uvedena formulace: *Průmysl 4.0, ale pro všechny typy a sektory průmyslu*. MES a Batch jsou konkrétní standardizované postupy, ale Průmysl 4.0 je vize, technické éra, iniciativa a cesta. Tedy něco mnohem rozsáhlejšího a obecnějšího než MES a Batch. Nedokáží-li se podniky orientovat v prostředí nových technologií, nebudou schopny zvolit si správný investiční záměr pro inovaci svých výrobních aktiv. Výhodu budou mít ty, které schopnost orientovat se prokázaly správnou volbou systémů MES nebo Batch.

Jak už zaznělo v kapitole 1.1, dotáhnutí systému MES na standardizovanou platformu MOM je milníkem nástupu Průmyslu 4.0. Doplněním systému MES do automatizační pyramidy se vertikálně propojily výrobní a podniková (obchodní) doména, a vše nasvědčovalo tomu, že bylo dosažené *komplexní automatizace celého podniku* a člověk se zapojí do výroby jen jako operátor nebo manažer. Nicméně je třeba vzít v úvahu, že průmyslová výroba je pestrá a i jinak bude vypadat MES pro mlékárnu a jinak pro automobilku.

Jakmile je celý podnik vertikálně propojen, nutně se musí zvýšení úrovně automatizace v kterékoliv části pyramidy projevit na vyšší produktivitě výroby. Otázkou je volba místa, kde úroveň automatizace zvyšovat. Odpověď vyplyne z typu výroby. U procesních výrob je vhodné zacílit automatizaci na dolní (shop floor) úroveň nasazením systému Batch (ISA 88), zatímco u výrob diskrétních přinese větší benefit zvýšení automatizace na úrovni MES, tedy řízení výroby. Tato volba, toto místo rozhodování je zásadní pro každého, kdo chce pochopit smysl Průmyslu 4.0. Uveďme dva příklady:

- Procesní výroba - míchání a ohřev mléka v tanku. Procesy míchání a ohřevu, pakliže jsou konstrukčně dobře navrženy, už nijak neumíme optimalizovat, jsou dány fyzikálními zákony. Takže cesta optimalizace by se ubírala organizací těchto procesů, vyloučením ztrátových časů a stanovením optimálních parametrů a podmínek pro vykonání míchání a ohřevu. Důležité je mít správná data ve správný čas, sladit procesy míchání a ohřevu nejen vzájemně, ale i v návaznosti na předchozí a následné procesy. Zásadní roli zde hraje příprava a přeprava materiálu před procesem a po něm (většinou potrubím či dopravníky, často pomocí gravitace). Důležité v těchto procesech je také dosažení variability a škálovatelnosti (teplota, objem, skladba složek) s minimální ztrátou času při modifikaci řídicího programu. Pro efektivní řízení dávkových procesů byl vytvořen standard ISA 88 (Batch).
- Kusová výroba - řezání desek pro nábytek. Proces řezání je také dán fyzikálními zákony a při správných parametrech stroje (pily) ho nedokážeme příliš urychlit. Je však důležité optimalizovat organizaci procesů řezání a transportu materiálu před a po procesu, do kterých se často zapojuje také lidský faktor. Pro efektivní správu diskretních procesů byl vytvořen standard ISA 95 (MES/MOM).

3.1.2 Horizontální propojení 3.0

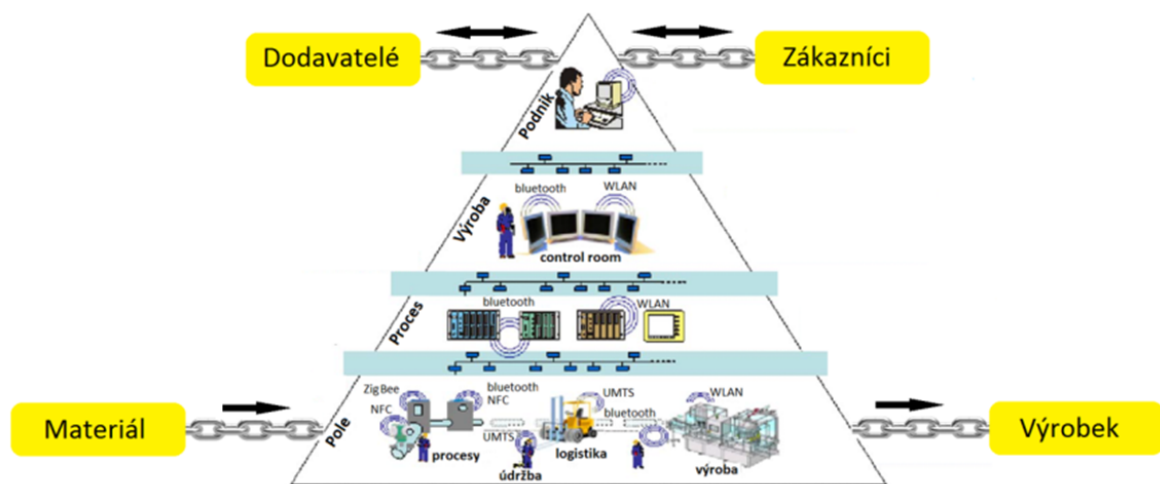
V současné průmyslové výrobě lze identifikovat systémovou horizontální integraci, která je realizována propojením obchodních (dodavatelských) řetězců. Se řízením řetězců dodavatelů/zákazníků a materiálu/výrobků jsou pevně spjaty systémy *Supply Chain Management* (SCM) a *Customer Relation Management* (CRM).

Avšak na rozdíl od relativně dobře zavedené vertikální integrace výrobních zařízení (zejména v případě ERP systémů) je horizontální integrace a koncepční přístup k uspořádání logistických procesů ve firmách často na začátku. Ze statistik ČSÚ vyplývá, že pouze 5 % malých, 12 % středních a 32 % velkých podniků využívá systémy SCM, a 14 % malých, 28 % středních a 52 % velkých podniků používá systémy CRM, přičemž problematické je, že tato čísla již delší dobu stagnují [130]. Oproti široce rozšířeným ERP systémům je to stále ještě výrazný rozdíl.

Skutečnost, že podnik v dnešní době nepoužívá systémy SCM, CRM či ERP může znamenat, že nemá zavedeny digitální kanály, prostřednictvím kterých se automaticky realizuje vazba plánování s výrobou a na dodavatelské řetězce.

Také je možné říci, že současná integrace je založena na výměně informací/statků pouze podél dvou krajních vrstev celé hierarchické struktury (viz obrázek 3.5), tedy jinými slovy, že

- na základě objednávky konkrétního produktu firma objednává materiál či polotovary a
- po dodání materiálu firma realizuje výrobek a ten předává zákazníkovi.



Obrázek 3.5: Horizontální integrace v Průmyslu 3.0

3.1.3 Propojení inženýrských procesů 3.0

Významnou součástí životního cyklu výrobku je inženýring. Odvíjí se od prvotní myšlenky na realizaci výrobku nebo výrobního zařízení, přes poptávku, nabídku, studie, koncepci, procesní 3D model, výrobní model, řešení automatizace, atd. až po zahájení výroby. Inženýring buď řeší sama výrobní firma a pak je přidružen k systému MES, nebo, což bývá častěji, je řešen jako subdodávka a je součástí dodavatelského řetězce.

Historicky firmy k inženýringu přistupovaly tak, že jednotlivé činnosti prováděla jednotlivá oddělení, která si projekt *přehazovala*. Jednalo se zejména oddělení procesní, strojírenská elektrotechnická a softwarové inženýrství. V angličtině se pro tento postup vžila fráze: *throw it over the wall*. Tedy „hodilo se to přes zeď ke kolegovi“.

V současnosti se systémy pro správu životního cyklu výrobku, které jsou pro asistenci v průběhu inženýringu jako stvořené, stávají de facto standardem. Nejen digitální návrh výrobku v různých CAD programech, ale také automatické vytvoření výrobní dokumentace, návrh a simulace výrobní linky pro výrobek či virtuální zprovoznění celého systému se dnes prosazuje stále častěji.

3.2 Přejít k Průmyslu 4.0

V posledních letech spatřila světlo světa řada technologií, které samotné a izolované nepůsobily na trhu dostatečně přesvědčivě, aby se staly hybnou silou konkurence a tím i trhu. Heroický boj na tomto poli svádí už dlouho systémy pro řízení výroby (MES) a podobná situace hrozí i systémům pro správu životního cyklu (PLM). Jsou ale desítky dalších menších technologií a nástrojů, které by značně usnadnily a zefektivnily výrobu, a přesto se nemohou prosadit. Vytvořil se převis nabídky nových technologií oproti konzervativnosti a neochotě investovat ze strany výrobců. Nejkritičtější byla tato situace v Německu. Právě odtud přišel spouštěč, na který se čekalo. Ono těch spouštěčů

je vlastně více a dohromady tvoří snad až revoluční vizi Průmyslu 4.0.

3.2.1 Spouštěče Průmyslu 4.0

Jedná se o technologie či nástroje, které se v posledních desetiletích vyvíjely často odděleně, vzájemnou synergií však pomáhají tvořit a naplňovat vizi Průmyslu 4.0 [72, 125, 39, 13, 9]:

1. **Pokročilá mechatronika:** Jedná se o součinnou kombinaci mechaniky, elektroniky a softwarového inženýrství. Příklady: Automatická pračka, ABS, či ESP systémy v autech, třídící a balicí linky, CNC stroje, letadla (obsahují stovky a snad tisíce malých mechatronických subsystémů, vestavěné elektroniky a vestavěného SW).
2. **Vestavěná elektronika (Embedded electronics):** S miniaturizací elektronických čipů a čidel a klesala jejich cena, stávaly se stále více součástí mnoha výrobků a dnes je můžeme nalézt téměř v každém zařízení.
3. **Senzory RFID či NFC:** Jsou to obvykle malé až miniaturní vysílače (často velmi tenké - do 1 mm), které mohou obsahovat libovolná data o objektu, k němuž jsou fyzicky připojeny. V současnosti se využívají zejména pro sledování skladových zásob, pohybu dílů a výrobků ve výrobních fázích životního cyklu. Informace načtené ve výrobním procesu se přenášejí do ERP systémů, a tak v každém okamžiku jsou známy přesné stavy skladových zásob a pohybů výrobků. Operace načtení RFID značek zboží na paletě během expedice způsobí automatickou změnu stavu všech dotčených objednávek v příslušném systému. Díky automatizaci sledování pohybu zboží a materiálu došlo k výraznému zvýšení efektivity výrobních a logistických operací.
4. **Analýza velkých dat (Big Data):** Objem vytvářených dat a s ním i potenciální množství v nich obsažených informací exponenciálně narůstá. Jde například o data různých čidel sledujících výrobní proces, logistiku výrobních závodů, inteligentní senzory, CRM systémy, a také samozřejmě obrazová data. Přestože cena za snímání dat ve většině oblastí rychle klesá, je schopnost získávání užitečných informací stále omezená, a tak většina znalostí, které by bylo možné z dat určit zůstává neodkryta. Za velká data se obvykle považují taková data, která přesahují možnosti současných databázových technologií ve všech těchto aspektech:
 - *Volume* - velký objem dat (v rozsahu 10^{15} bajtů a více).
 - *Velocity* - velká rychlost pořizování dat.
 - *Variety* - velká rozmanitost pořizovaných dat.

Problematika velkých dat se nejčastěji řeší distribucí dat na více zařízení se speciálními souborovými systémy (např. HDFS - Hadoop Distributed File System). Druhou možností je využití jiných typů databází než relačních (NoSQL

úložišť typu klíč-hodnota), které jsou projektovány za účelem vysoké rychlosti a škálovatelnosti na úkol datové integrity.

5. **Autonomní roboty:** Na rozdíl od klasických výrobních robotů, které jsou navrženy pro plnění specifických úloh v rámci výrobního procesu jsou, zejména ve velkých firmách nově zaváděny také univerzální inteligentní roboty, kteří se bez rekonfigurace mohou uplatnit při plnění širšího spektra úloh. To vede ke zkvalitnění produkce firem a úspoře pracovních sil. Nevýhodou jsou samozřejmě vysoké investice a poptávka po nových typech profesí, a proto je velmi důležité a obtížné odhadovat návratnost investic.
6. **Datová úložiště a cloudové výpočty:** Spolu se vzrůstajícími požadavky na uložení dat rostou i požadavky na jejich rychlé a bezpečné zpracování. Geograficky oddělená datová centra dostupná po celém světě v tomto směru nabízí řadu dalších služeb založených na třech základních principech. Těmi jsou jednak ① poskytování software jako služby³, ② poskytování platformy jako služby⁴ a ③ poskytování infrastruktury jako služby⁵. Zatímco první dva zmiňované poskytují uživateli resp. integrátorovi předem připravené softwarové aplikace či platformní moduly a nabízejí možnost jejich integrace, poslední zmiňovaná varianta představuje “pronájem” hardwarové infrastruktury a zprovoznění operačního systému, na kterém již integrátor staví svá specializovaná řešení.
7. **Průmyslový internet věcí a internet služeb:** Se zavedením protokolu IPv6 v roce 2012 vznikl prostor pro připojování velkého množství nových zařízení k Internetu, protože IPv6 k tomu nabízí dostatečný rozsah adres. S vývojem internetu věcí se rozvíjí komunikace typu machine-to-machine (M2M), tedy propojení mezi jednotlivými *vestavěnými zařízeními*. Dosavadní existující M2M zařízení využívají frekvence pod 1 GHz a slouží pro přenos dat relativně nízkou rychlostí od zařízení např. do centrální databáze nebo mezi zařízeními a člověkem. Využívají se v širokém spektru oborů, od individuálního řízení “chytrých” domácností, dohledových systémů či měření spotřeb energie v energetických systémech po moderní lékařské aplikace sloužící pro diagnostiku zdraví pacientů. Téměř nekonečné množství zařízení připojených k internetu však nebylo, a ani nemohlo být v počátcích vzniků síťových protokolů očekáváno. Proto také stále ještě hojně využívaný protokol IPv4 neposkytuje dostatečné možnosti adresování - prostor IP adres je již z více než 99 % obsazen. Mnoho počítačů, tabletů a chytrých telefonů je proto organizováno ve vyhrazených podsítích (v privátních rozsazích adres 192.168.x.x či 10.x.x.x). Tyto adresy lze adresovat pouze prostřednictvím směrovačů a dynamických IP adres. Webové kamery, herní konzole, televize a další domácí spotřebiče však stále zvyšují poptávku po vlastních IP adresách. S protokolem IPv6 (RFC 2460) je k dispozici 32 místné adresování, které pokrývá adresní prostor více než 340 sextilionů IP adres. To umožní adresovat cca 600

³SaaS - *Software as a service*

⁴PaaS - *Platform as a service*

⁵IaaS - *Infrastructure as a service*

bilionů adres na každý čtvereční milimetr zemského povrchu⁶. IIoT představuje sítě propojující zařízení ve výrobních závodech, tedy např. průmyslové senzory, výrobní stroje a další. Z těch jsou data nahrávána na cloud a následně zpracovávána. IIoT je způsob, jak se zbavit nedostatků mezi oblastmi IT a OT. IIoT by po integraci do výrobního závodu mělo přinést zpřístupnění monitoringu výroby. Na základě vyhodnocování (big) dat pořizovaných prostřednictvím IIoT je možné různě optimalizovat výrobní procesy a další průmyslové procesy [39].

8. **Aditivní výroba:** Nelze na tomto místě nejmenovat, ale vzhledem k její výjimečnosti byla již zmíněna kapitole 1.4.2.
9. **Indoor-positioning systémy:** Znalost polohy materiálu i pracoviště je klíčová pro optimalizaci toku materiálu napříč výrobními i logistickými procesy. Ve venkovním prostředí dobře poslouží systémy satelitní navigace GPS. Kdekoli „pod střechou“ však tyto systémy přestávají být dostatečně přesné, případně nefungují vůbec. Systémy lokalizace ve vnitřním prostředí obvykle neposkytují přesnost srovnatelnou s GPS venku, avšak pro hrubou informaci o poloze objektů pro účely optimalizace postačují. Zásadní jsou také geometricky kvalitní digitální mapy, tedy digitální modely prostředí, na základě kterých může optimalizace probíhat (nejkratší spojnice mezi dvěma body nemusí být v rámci tovární haly nutně tou nejlépe použitelnou).

Indoor lokalizace je klíčovou technologií v případě, kdy jsou do výroby zapojeny např. autonomní mobilní dopravníky. Sebelokalizace, jakožto i znalost přesné informace o umístění výrobku, cíli jeho přesunu, polohy stroje dokáže pomoci nejen při plánování dráhy pro stroj, ale také při rozhodování, který z mnoha strojů povolát.

10. **Rozšířená realita:** Rozšířená realita může být považována za vylepšení zobrazení v reálném čase s počítačově generovaným senzorickým vstupem, jako jsou video, obrázky a grafika, zvuk nebo data z GPS. Příkladem rozšířené reality je např. v automobilovém průmyslu projekce informací na čelní sklo. Použití rozšířené reality spolu se simulačními modely může urychlit celý výrobní řetězec - od počátečního dotazování zákazníka až po vývoj, výrobu a dodávku produktu. Ke skutečným objektům jsou (virtuálně) umístěny (například pomocí chytrých telefonů, tabletů nebo speciálních brýlí) další virtuální kontextové informace. Rozšíření (augmentace) může být v pokročilých systémech velmi bohaté (2D, 3D) a je umístěno přesně do prostoru sledovaných objektů nebo je může i překrývat nebo nahrazovat.

Dalšími možnostmi rozšíření informačního kanálu směrem k uživateli může být přidání zvukových vjemů. Tyto prostředky se obvykle používají například v navigačních systémech a výukových aplikacích. Integrace dalších typů informací,

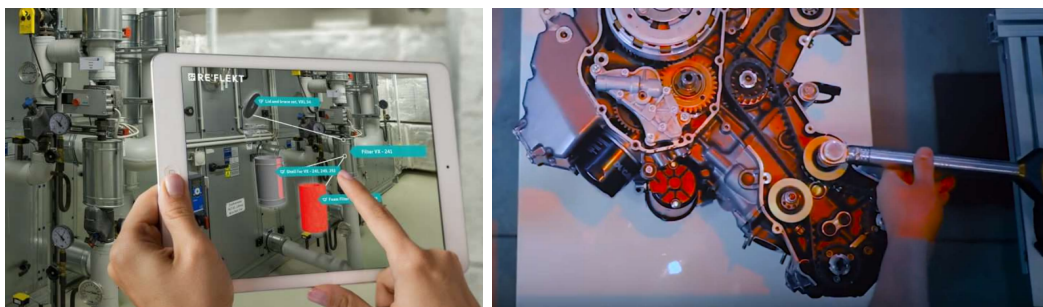
⁶Pro kompaktní reprezentaci IPv6 adres se používají hexadecimální čísla. Značně nepřehledný zápis IPv6 adresy je možné v některých případech zkrátit, a tak např. hodnotu “::1” lze použít k popisu místní adresy hostitele složené z 15 nul a jedničky, což odpovídá místnímu hostiteli 127.0.0.1 v IPv4

např. hmatových nebo čichových není pro AR typické a objevuje se spíše v aplikacích *virtuální reality* (VR⁷) [91].

Velice slibnou budoucnost mají technologie rozšířené reality také v následujících případech:

- Skladové a logistické operace - rozpoznávání objektů a čtení čárových kódů na velkou vzdálenost, automatická navigace v dynamicky rozložených skladech.
- Doprava - projekce s informacemi o provozu a navigace na skle nebo v brýlích.
- Value-services - dodavatel zařizuje montáž zboží, které přejímá od subdodavatelů, pracovníci montáže jsou následně navigováni pomocí AR, což přinese možnost kontroly výsledku a snížení nároků na proškolení pracovníků. Pracovníci budou moci vykonávat širokou škálu úkonů.
- Servis - pomoc při servisu od zkušenějšího pracovníka, vizualizace postupu servisu (např. rozebrání a složení výrobku) v AR.
- Internet věcí - vizualizace stavu věcí.
- Likvidace a recyklace výrobků - demontáž cenných komponent naváděná AR aplikací.

Sloučení virtuálního a reálného světa v ideálním případě pomůže uživateli pochopit jeho prostředí. Zaměstnanci a servisní pracovníci tím mohou dosahovat lepších výkonů a generovat mnohem méně chyb, pokud mají před očima klíčové informace a údaje, s brýlemi umístěnými na hlavě, a tím obě ruce uvolněné pro práci.



Obrázek 3.6: Dva přístupy implementace rozšířené reality

Při kombinování reality a její augmentace se používají dva principy:

- **Video see-through (mobilní telefon, tablet)** - generované vizuální objekty jsou vkládány do videosignálu, který je zobrazen na obrazovce zařízení (obr. 3.6 vlevo). Mezi výhody můžeme řadit vizuálně přesnější augmentaci, nevýhodami je pak nutnost zařízení držet v ruce nebo pomocí držáku a

⁷Virtual Reality - systémy, které poskytují kompletně digitální vjem bez jeho kombinace s vjemem ze skutečného světa.

možnost mírného zpoždění oproti pohledu na reálný svět. Je také potřeba brát v potaz, že záběr ze zařízení nemusí být totožný s pohledem uživatele, což může či nemusí být problém. Tento přístup je vhodný například ke kontrole stavu technologie, kdy obsluha může mířit na jednotlivá zařízení a získávat detailní informace o jejich stavu.

- **Optical see-through (průhledové brýle)** - vizuální informace jsou přidávány přímo do cesty, kterou prochází obraz reálného světa do očí uživatele (obr. 3.6 vpravo) a ten pak může mít volné ruce. Tento přístup se tak může uplatnit např. u pracovníků montáže či servisu, kteří díky němu mohou získat důležité informace během práce. Právě díky tomu, že přístroj není nutné držet v ruce, bude tento způsob implementace do budoucna více preferovaný. Největším problémem při aplikaci této metody je lokalizace uživatele v prostoru a určení směru pohledu [72].

11. **Kyber-fyzické systémy:** jsou kombinací fyzických zařízení ve skutečném světě a počítačových systémů. Lze je zařadit nejen mezi tzv. spouštěče Průmyslu 4.0, ale jsou také jeho velkým cílem, jak bude blíže popsáno v následující kapitole.

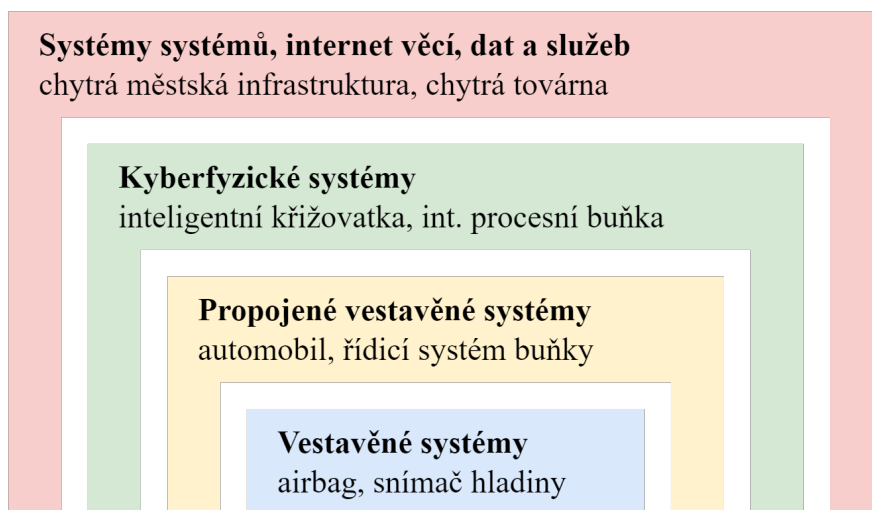
3.2.2 Kyber-fyzické systémy

Termín kyber-fyzické systémy se poprvé objevují v *National Science Foundation v USA* v roce 2006. Zatímco *kyber* má hovorově také význam iluzorního světa vytvořeného z počítačů, *fyzický* je skutečný svět lidí a věcí [3].

Kyber-fyzické systémy (CPS) mohou být definovány jako spojení virtuálního světa a fyzického světa dohromady v jeden skutečně propojený svět, v němž inteligentní objekty spolu vzájemně komunikují a spolupracují. Jejich základem jsou mechatronické systémy. Ty typicky zahrnují senzory, ovladače, uživatelská rozhraní, embedded systémy pro řízení, stejně jako interní a externí komunikační rozhraní.

Kyber-fyzický systém představuje další evoluční krok od mechatronického systému, neboť je mnohem více integrován v prostředí, v němž se nachází. Často také obsahuje model prostředí a podle něj dokáže přizpůsobit své chování. CPS je propojen a dokáže komunikovat s jinými CPS, či se službami. Skupiny CPS poskytují základ pro vytvoření internetu věcí, který v kombinaci s internetem služeb umožňuje činnosti podle vize Průmyslu 4.0 [88].

Obrázek 3.7 ilustruje vizi globálního *internetu věcí, dat a služeb*. Pro příklad je možné uvést mechatronický systém - airbag automobilu, brzdový systém apod. Ty představují počáteční bod (nejnižší stupeň integrace). Tyto systémy jsou propojeny vnitřní komunikací a společně formují kyber-fyzický systém (automobil), který má díky integraci velkého množství potřebných subsystémů schopnosti snímat a ovlivňovat stav okolního světa a zároveň komunikovat s dalšími podobnými kyber-fyzickými systémy např. prostřednictvím některé globální sítě. V případě automobilu to mohou být okolní automobily, mezi kterými se vyměňují informace o dopravní situaci, či např. systém



Obrázek 3.7: Od embedded systémů k internetu věcí, dle [3]

řídicí dopravní křižovatku. Takový systém zase bude komunikovat s okolními systémy tak, aby společně optimalizovaly průjezdnost městem. Takové řešení dopravní situace pak bude formovat jeden z aspektu chytrého města (*Systému systémů*) [3, 9].

Kyber-fyzické výrobní systémy

V oblasti výroby se v souvislosti s CPS často používá také pojem *kyber-fyzický výrobní systém* (CPPS - Cyber-Physical Production System) - složení výrobních zařízení, agregovaných produktů, lidských zdrojů, které spolu vzájemně interagují prostřednictvím komunikačních rozhraní. Tato rozhraní se využívají pro monitorování a řízení operací CPPS, jakož i pro využívání znalostí generovaných jak lidskými zdroji, zařízeními během výrobního procesu, tak i agregovanými produkty během jejich životního cyklu. Tyto interní informace a znalosti jsou součástí digitálního stínu systému a bývají využity v různých časových měřítcích k neustálému zlepšování provozu či např. k informování o potřebě surovin, energie či servisu systému. Kyber-fyzicky definované produkty vytváří další možnosti využití výrobních a stavových informací v externích systémech prostřednictvím internetu věcí a služeb [107].

Příkladem může být uzavřený embedded systém - snímač hladiny umístěný v konkrétní výrobní buňce (viz obr. 3.7), či driver servopohonu podavače materiálu v těžební buňce. Tato zařízení jsou kombinací mechatroniky, elektroniky a řídicího software. Průmyslovou komunikační sběrnici jsou pak propojeny k řídicímu systému buňky. Celou výrobní buňku s operátorským rozhraním pak můžeme vnímat jako kyber-fyzický systém, která může s pomocí externích *služeb* (v rámci internetu věcí - IoT, služeb - IoS či osob - IoP) poskytujících další informace a případně funkcionalitu jednat autonomně.

Okolní výrobní buňky, tedy okolní CPPS jsou schopny vzájemného dorozumívání a koordinace výroby a společně tvoří výrobní linku (opět vnímána jako kyber-fyzický systém). Propojení mnoha kyber-fyzických systémů vytváří kompozitní systém (systém

systémů). V tomto případě je označován jako *chytrá továrna* [39].

Při udržování dynamiky a cyklu neustálého zlepšování CPPS hraje zásadní roli zachycování a ukládání dat a automatické generování znalostí, stejně tak jako činnost lidských zdrojů, protože často nelze jejich znalosti plně formalizovat a zpracovávat automaticky [87].

Kyber-fyzické systémy mimo průmyslovou oblast

Vzhledem k tomu, že transformace embedded systémů na kyber-fyzické systémy je dnes všudypřítomná, je vhodné zmínit několik dalších příkladů z jiné než výrobní oblasti.

V oblasti energetiky dnes také dochází k velkému přerodu a vzniká mnoho lokálních zdrojů elektrické energie. Jednotlivé zdroje různých výkonů (jistě také CPPS) mohou v kybernetickém světě jednat autonomně dle předem daných pravidel a samy rozhodovat o způsobu využití vyrobené energie (prodeji, uložení či spotřebě). Celý trh je založen na známé ceně elektrické energie (a její předpovědi). Jiný způsob řízení prakticky není možný, protože vzhledem k rozlehlosti sítě a množství připojených zařízení nelze provádět plánování centrálně.

Během kardiologické operace je v dnešní době nutné na určitou dobu zastavit srdce. Je zřejmé, že taková procedura je extrémně riskantní, a fakt, že pacientovo srdce nepracuje, přináší množství nežádoucích efektů jak během operace, tak později při rekonvalescenci. Mnoho vědeckých týmů a výrobců přístrojů pro mikrochirurgii se proto zabývá vývojem zařízení, které by umožnilo chirurgům operovat srdce, které bije. Tento požadavek přináší dvě výzvy – samotné fyzické zásahy musí samozřejmě provádět robot, který sleduje nejen pokyny operátora, ale také automaticky uzpůsobuje polohu nástrojů tvaru srdce v příslušné části jeho cyklu. Kamerový systém tedy srdce nepřetržitě sleduje a porovnává s detailním modelem. Takový systém umožní chirurgovi vyvinout nástrojem konstantní tlak na určité místo, přestože je celé srdce neustále v pohybu. Druhou výzvou je pak umožnit chirurgovi prostřednictvím stereoskopického obrazu sledovat srdce jakoby v klidu, tedy vytvořit iluzi zastaveného srdce a promítnout do takového obrazu polohu používaných nástrojů. Je zřejmé, že realizace této úlohy je náročná po stránce hardwarové, softwarové, je také nutno vytvořit detailní model srdce. Celý kyber-fyzický systém nadto musí být extrémně bezpečný a robustní [12, 39].

Kybernetický systém a digitální dvojče

Fyzický a virtuální svět existují bok po boku a vzájemně se doplňují. Fyzické výrobky a stroje spolu komunikují prostřednictvím svých zástupců (softwarových modulů - *agentů*) ve virtuální továrně s cílem detailně monitorovat průběh výroby a organizovat, na které lince a na kterém stroji bude další krok výroby uskutečněn.

Aby mohl kyber-fyzický systém korektně fungovat a správně reagovat na podněty okolních systémů, musí obsahovat digitální dvojče fyzického systému. Fyzická a

virtuální část spolu komunikují, interagují, a tedy spolupracují prostřednictvím řídicího algoritmu. Digitální dvojče může zrcadlit jedno koncové zařízení, ale také integrovat celou výrobní linku či kompletní inteligentní továrnu, ve které mezi sebou komunikují nejen fyzický a virtuální model, ale i jednotlivé kyber-fyzické systémy. Takové propojení pak tvoří kompozitní systém (systém systémů). Kyber-fyzický výrobní systém může mít více různých digitálních stínů popisujících různé aspekty systému (nemusí se tedy jednat o *úplnou* reprezentaci celého systému) v různých okamžicích, zatímco digitální dvojče existuje pouze jediné.

V kybernetickém systému je možno simulovat a optimalizovat skutečný fyzikální systém, například výrobu v předstihu, a to zejména pokud jde o problémy. V ideálním případě je virtuální obraz stále aktuální a bude aktualizován po celý životní cyklus.

3.2.3 Vertikální integrace v Průmyslu 4.0

V předchozím textu bylo popsáno vertikální uspořádání hierarchie pro Průmysl 3.0. Jak bylo řečeno, základem řízení výroby je MES, který je nejefektivnější, je-li propojen se systémem ERP, který je mu v hierarchii nadřazen, a s výrobními zařízeními a monitorovacím systémem (SCADA), které jsou mu podřízeny.

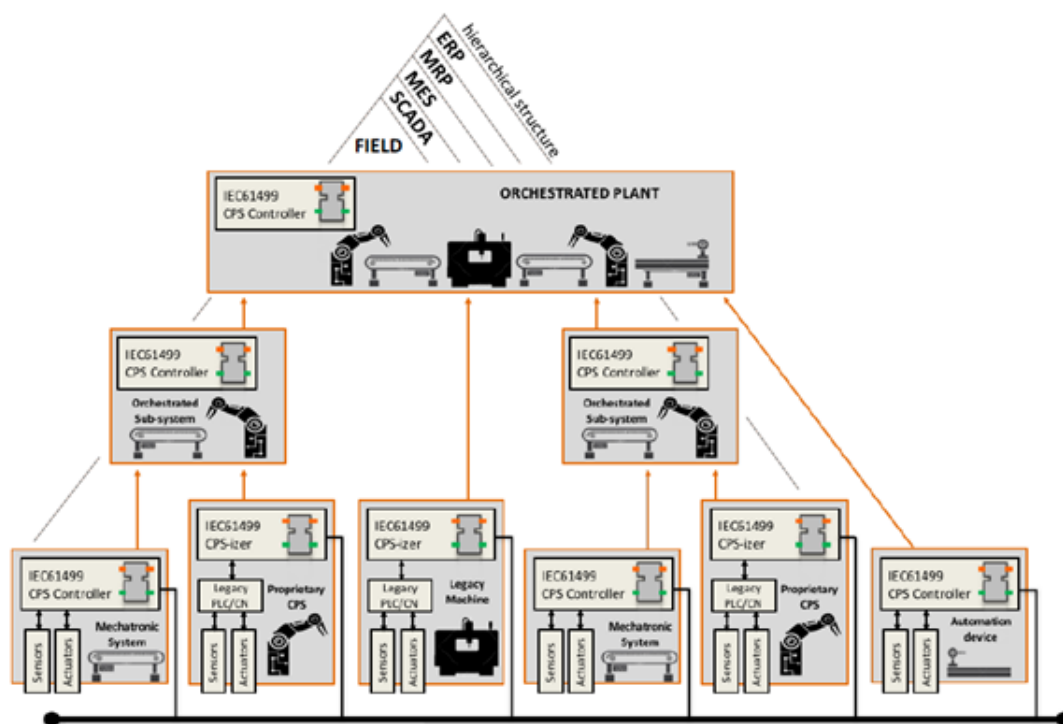
Průmysl 4.0 tuto hierarchii narušuje a zavádí a legitimizuje propojení *každého s každým*, přičemž nezáleží, jakou funkci konkrétní zařízení (hmotný stroj či jen softwarová služba) v rámci procesu vykonává, a na jaké úrovni hierarchie by se nacházelo v tradičním pojetí výroby. CPS a CPPS nabízejí po zapojení do výrobního procesu nové datové toky, které je zapotřebí integrovat. V takovém uspořádání pak např. CPPS stroje ví, že musí zastavit výrobní dávku nebo odeslat určitá procesní data. Jiná entita pak tato data přijímá. CPPS obsahující inteligentní výrobek zná svůj výrobní postup a má schopnosti vyjednat jeho uskutečnění. Co se ale v takovém případě stane s dalšími entitami a systémy, které byly dosud přítomny v hierarchii podniku? Konkrétně se systémem MES?

Je jasné, že v MES systémech musí nastat změna. Existují dvě skupiny příznivců vize Průmyslu 4.0 s odlišnými názory na existenci systému MES. První skupina předpokládá, že MES zůstane a jeho funkcí bude agregace nových datových toků a dávání je do souvislostí. Druhá skupina ve vertikálním propojení se systémem MES nepočítá. Skutečnost bude zřejmě taková, že v první fázi uskutečňování vize Průmyslu 4.0 se bude MES ještě využívat a přímé propojení řídicí procesní vrstvy a cloudových platform bude vzdálenější vizí.

Ve skutečnosti je to zajímavá otázka: „Mohou MES z podniků zcela zmizet, protože jejich funkce budou integrovány do jiných systémů?“ Co vše zastřešuje pojem *Manufacturing Execution System* - Systém pro řízení výroby?

Dle organizace *Manufacturing Execution System Association* (MESA), na jejíž práci je založen standard ISA 95 je moderní MES modulární softwarovou platformou, dle standardu nazývanou *Manufacturing Operation Management* (MOM). Jejím jádrem je

jedna či více databází (většinou relačních) a její jednotlivé, a do určité míry nezávislé softwarové moduly plní funkce definované ve čtyřech podnikových doménách - doméně *výrobní*, *skladové*, *údržby* a doméně *kvality*. V rámci těchto čtyř domén existuje nepřehledné množství činností, které musí být řešeny a na jejichž výsledcích je chod moderního podniku zcela závislý.



Obrázek 3.8: Propojení jednotlivých CPPS [110]

Jako protipól k centralizovanému MES systému se nyní zaměříme na CPPS řízené výpočetním systémem s omezeným výkonem, v nichž jsou implementováni softwaroví agenti umožňující autonomní rozhodování. Je zřejmé, že výrobní zařízení mohou být schopny rozhodovat o některých funkcích domény *výrobní*, určitě mohou sledovat svůj stav a rozhodovat o nutnosti *údržby* a mohou být také schopné sledovat *kvalitu* své produkce, či požadavky na obsluhu (*doplnění materiálu*). V každém produkčním systému však musí existovat entita či entity, se kterou mohou jednotlivá CPPS o svých potřebách vyjednávat. V doméně výrobní to mohou být softwaroví agenti zastupující výrobky, v dalších doménách pak agenti zastupující např. pracovníky - skladníky řešící potřeby naskladnění/vyskladnění různých výrobků, údržbáři udržující stroje v činnosti na základě rozhodování jejich agentů atd. Dosud popsané možnosti samostatného rozhodování zní poměrně představitelně a racionálně. Jinými slovy řečeno - kromě běžných činností spjatých s řízením procesů (snímání veličin, ovládání akčních členů, sběr dat, reporting nadřazenému systému), budou CPPS částečně implementovat i funkcionalitu z vyšších vrstev původní hierarchie (naznačeno na obr. 3.8).

Existují ale také situace, ve kterých je a bude nutné se stále spolehnout na rozhodování či asistenci určitého centrálního členu. Tyto činnosti mohou být nezbytné například již během koordinace přeplánování výrobních úloh v případě, kdy selže nějaký

výrobní stroj. Takto komplexní úkol, který bude navíc časově kritický, nemusí jednotliví agenti s omezenými výkonovými možnostmi, a s omezeným rozhledem, být schopni efektivně realizovat. Dále bude vždy nutné, aby jednotlivé CPPS, ať jsou jakkoli autonomní a datově nezávislé, vyměňovaly svá výrobní data s dalšími systémy, které je budou dále zpracovávat a archivovat v datových úložištích, např. pro účely výpočtu *Key Performance Indikátorů* (KPI), trasování výrobního postupu, sledování pracovníků a mnoha dalších činností, které v dnešní době zaštiťují systémy MES. Zůstaňme jen u výroby - než se zrodí softwarový agent zastupující objednávku a posléze produkt ve výrobě, je objednávka evidována jako pasivní položka v databázi a musí existovat další systém (agent), který iniciuje spuštění instance agenta budoucího výrobku a jeho naplnění daty.

Funkce MES s největší pravděpodobností nezmizí ani v budoucnu. Možná že společně nebudou označovány jako MES (či MOM), nebudou provozovány jako jeden ohraničený a definovaný systém, zůstanou ale zachovány, a to zřejmě v distribuované podobě jako skupiny CPS - softwarových služeb, které budou spolupracovat s jednotlivými výrobními jednotkami a budou také potenciálně moci řídit či harmonizovat jejich činnost. Hierarchická architektura se rozplyne na zastřešující síť CPPS a softwarových služeb s přímým komunikačním kanálem od úrovně procesu, až po ukládání velkých dat v cloudových systémech. Pro tuto síť bude charakteristické, že komunikačním médiem nebudou průmyslové komunikační sběrnice a sítě v rámci podniku, ale Ethernet, standardní komunikační protokoly a na nejvyšších vrstvách *architektura orientovaná na služby* (SOA). Funkcionalitu jednotlivých CPPS a služeb pak bude možné využívat v souladu s principy SOA a např. ji poskytovat napříč závody i výrobními podniky ve formě webových služeb [50, 69].

3.2.4 Horizontální integrace 4.0

Horizontální integrace ve smyslu Průmyslu 4.0 přesahuje vnitřní operace výrobního podniku. Pomocí nové generace souhrnných hodnototvorných sítí spojuje všechny subjekty dodavatelského řetězce, který začíná u dodavatelů, pokračuje přes výrobce a končí u distribuce výrobku koncovému zákazníkovi, přičemž počítá i s následným servisem. Mimo to zahrnuje obchodní partnery i zákazníky napříč jednotlivými zeměmi.

V souvislosti s Průmyslem 4.0 se často mluví o potřebě efektivně vyrábět malé série produktů, nezdědka dokonce vyrábět každý produkt odlišný na přání zákazníka. Po zadání objednávky zákazníkem vytvořené konfigurace se požadavek subdodavatelé firmě dostane prakticky v reálném čase. Ta musí být schopna rychle vyrobit požadované produkty, případně opět v reálném čase popat materiál či potřebné díly, které pro výrobu bude potřebovat.

Umožňuje to právě horizontální integrace, v rámci které jsou všichni subdodavatelé propojeni s montážním závodem. Navenek se celý tento dodavatelský řetězec chová jako jediná továrna, ve které každý stroj a každý člověk ví, co má právě dělat, aby výsledkem byl produkt, který v co nejkratším čase doputuje ke konečnému zákazníkovi.

Jako konkrétní příklad horizontální integrace poslouží (ostatně jako ve všech oblastech Průmyslu 4.0) právě výroba automobilů. Např. Škoda Auto využívá služeb několika stovek subdodavatelů [61]. To je úctyhodné číslo a sladění takového počtu firem jako dobře fungujícího orchestru je možné jen stěží. Subdodavatel Škody, který např. vyrábí blinkry je navíc nevyrábí jen pro jednu automobilku, ale hned pro několik⁸, a pro každou několik druhů. Taková firma má dvě možnosti: vyrábět na objednávku, tedy se zpožděním, které nastane, než se k ní dostane požadavek konečného zákazníka koupit si konkrétní vůz, nebo vyrábět na sklad a mít požadovaný blinkr ihned k dodání. První způsob přináší dlouhé čekací doby zákazníkovi, druhý zase enormní potřeby firmy na skladovací prostory a na služby se skladováním spojené.

Provést objednávku dílů v (téměř) reálném čase a počkat na jejich dodání ještě samo o sobě nepřináší žádnou podstatnou inovaci oproti způsobům Průmyslu 3.0. Podstatné je zde to, že horizontální integraci v Průmyslu 4.0 si lze představit jako propojení nejen na nejvyšší a nejnižší úrovni hierarchie (tedy zadání objednávky a dodání produktu, jak je uvedeno na obrázku 3.5), ale především jako komplexní sdílení výrobních údajů, postupů, procesů a požadavků prostřednictvím jediné softwarové platformy. V popisovaném případě Škody Auto je jí platforma pro správu životního cyklu produktu *Siemens Teamcenter* popisovaná v kapitole 2.3.

3.2.5 Integrace inženýrských procesů 4.0

Integrace inženýrských činností je speciální případ kombinace horizontální i vertikální integrace vztahující se na inženýrské činnosti v rámci celého životního cyklu výrobního zařízení i produktů, od plánování, přes hrubé zadání, design, vývoj, realizaci, testování, verifikaci až po prodejní služby.

Důležitým faktorem integrace je používání komplexních inženýrských rámců⁹, které umožní provádět inženýrské činnosti v jednotlivých fázích životního cyklu i etapách bez přerušení.

Pro osvětlení funkce a užitečnosti správně navrženého inženýrského rámce je vhodné začít ukázkou postupu práce bez něj. Můžeme si představit vývojové oddělení ve firmě, kde technologové, konstruktéři, návrháři elektro, a softwaroví inženýři používají pro svou práci aplikace, sice špičkové, ale různých výrobců, které vzájemně neumí spolupracovat. Modely, které navrhnu technologové musí být ručně transformovány na konstrukční výkresy, ručně musí být vytvořena elektrická schémata a ručně musí být navržen řídicí systém a jeho hardwarová konfigurace. S trochou nadsázky se dá říct, že v takové firmě se stále jen přepisují informace mezi vzájemně nekompatibilními systémy už v průběhu vývoje. A to jsme zatím neřešili požadavky na změny! K takové práci již není dnes důvod, protože existují prostředky, které integraci činností umožňují.

⁸Např. Panasonic Automotive Systems Europe (má závod i v Pardubicích - PASCZ) dodává audio-techniku pro automobily Toyota, VW, Audi, Honda, Suzuki, Porsche, BMW a kolínskou automobilku TPCA [27].

⁹*frameworků*, komplexních softwarových balíčků.

Ať už se jedná o diskrétní, spojitě či dávkové procesy, je jejich chování možné popsat analyticky a modelovat. A právě existence modelu výrobního procesu je důležitým faktorem integrace. Kolem koncepčního modelu se totiž následně kumulují další činnosti, dochází k jeho zpřesňování a rozšiřování do dalších domén. Je také možné model napojit na fyzický či simulovaný řídicí systém (tedy využít tzv. princip hardware-in-the-loop či software-in-the-loop).

Řetězec softwarových nástrojů napojených na PLM, který umožní neintegrovat zastaralý způsob projektování nahradit koncepčním způsobem práce s důrazem na minimalizaci opakovaného vkládání dat, se nazývá *inženýrský rámec*. Jednotlivé nástroje rámce pak umožňují uživatelům soustředit se na postupné návrhy v procesní, strojírenské, elektrotechnické a softwarové doméně vždy s využitím všech již definovaných informací.

Jak již bylo popsáno, mnoho užitečných výhod přináší v průběhu vývoje využití digitálního modelu jak výrobku, tak i výrobního zařízení. Simulace modelu, ověření jeho funkce a jeho zprovoznění s fyzickým či simulovaným PLC umožňuje konstruktérům v různých fázích vývoje opakované ověřování, zda skutečná funkčnost odpovídá té projektované (tedy virtuální zprovoznění). Výrobní inženýři jsou dále schopni poskytnout okamžitou zpětnou vazbu konstruktérům v případě, že se vyskytnou jakékoliv problémy ve vyrobitelnosti dané součásti. Tento typ spolupráce mezi výrobními inženýry a konstruktéry vytváří ucelený obraz návrhu výrobku a postupu výroby. Tzn., že to americké označení „throw it over the wall“ v Průmyslu 4.0 již neplatí.

3.3 Shrnutí kapitoly

Historicky oddělené informační a operační technologie se v posledních desetiletích začaly vzájemně přibližovat a čím více se prolínají, vznikají také nové požadavky na pracovníky, kteří tyto technologie mají nasazovat a obsluhovat.

Třemi hlavními směry průmyslové integrace jsou *vertikální propojení*, *horizontální propojení* a *propojení inženýrských procesů*, přičemž nezáleží na tom, jestli se jedná o minulé, či moderní pojetí průmyslové výroby. Možnosti, způsoby a prostředky využívané dříve a dnes se však liší.

Následující kapitola již představí konkrétní technické řešení, jehož využití by mělo výše popisované způsoby integrace realizovat a naplnit tak cíle průmyslové transformace.

Kapitola 4

Digitální dvojče z pohledu průmyslu 4.0

4.1 Model RAMI

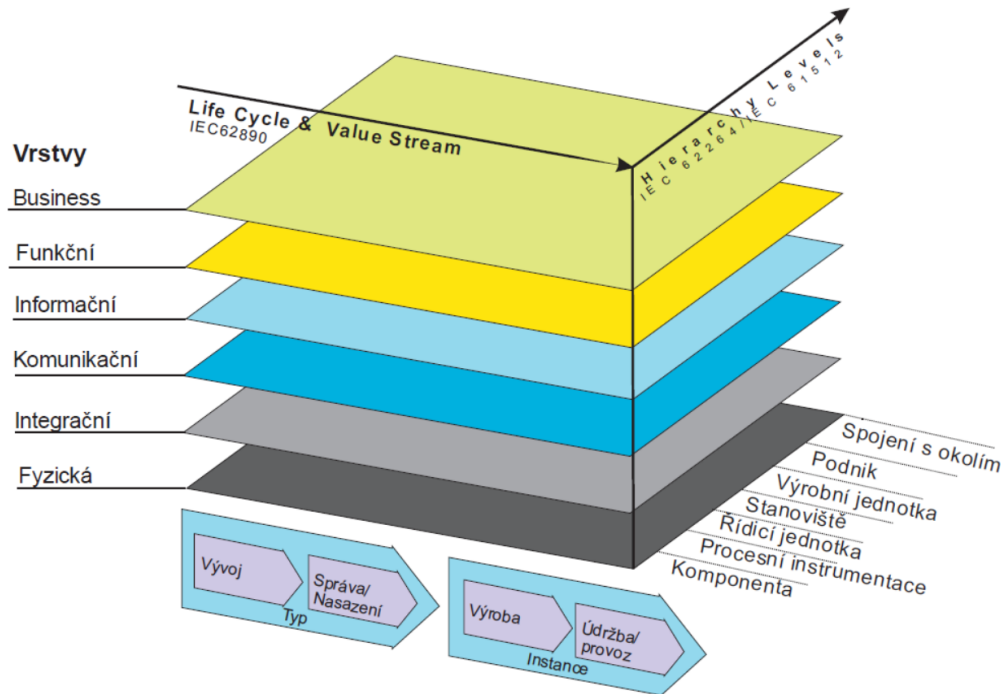
Digitalizace průmyslu, která byla zmíněna v kapitole 3.2, bude vytvářet úplně nové způsoby komunikace a spolupráce mezi firmami. Dodavatelské řetězce budou vzájemně efektivně propojeny, ale umožní také rychlé přerušení a náhradu dodavatele již v případě, že problémy s dodávkami budou teprve hrozit.

Důležitou podmínkou pro možnost sdílení myšlenek a vzájemného porozumění je seskupení velmi rozdílných aspektů do společného *modelu*. Jedním z nich je horizontální integrace rozšiřující pole působnosti za hranice jednoho podniku. Dalším je vertikální integrace jednotlivých zařízení v rámci jediného podniku. K tomu Průmysl 4.0 zavádí další aspekty, jako je integrace produktu do hierarchie či kompletně (*end-to-end*) integrovaný inženýring v rámci hodnotového toku. Model musí být schopen zahrnout úroveň milisekundových řídicích smyček pohonů stejně jako popis spolupráce mezi několika podniky. Z modelu pak vychází referenční architektura, která podrobně popisuje implementační možnosti a pravidla.

Samozřejmě již v současnosti existují přístupy a metody pro modelování a implementaci entit a vztahů z různých aspektů. Jedná se např. o OPC-UA (IEC 62541) pro implementaci komunikačních prostředků, Common Data Dictionary (IEC 61360) pro specifikaci hierarchie a datových modelů průmyslových zařízení či např. AutomationML pro modelování v doméně *end-to-end* inženýringu. Bohužel ale všechny tyto metody zahrnují pouze úzkou část oblasti, kterou uvažuje Průmysl 4.0 [127].

Kvůli potřebě sjednocení více aspektů do jediného modelu byl vytvořen model RAMI 4.0, jehož autory jsou instituce BITCOM, VDMA, VDI/VDE a ZVEI. Jedná se o tříosý model, který má schopnost reprezentovat všechny charakteristické prostředky a aspekty výroby, u kterých se předpokládá vzájemná interoperabilita, byť v současnosti jsou často ještě propojovány *manuálně*. Jako základ pro model této architektury byl převzat již existující a standardizovaný model SGAM (Smart Grid Architecture Model), který se používá pro vývoj aplikací v energetických sítích typu *smart grid* od

úrovně výroby, přenosu až po spotřebu elektrické energie. Unifikovaný model RAMI, který byl vyvinut s využitím modifikované architektury SGAM, zjednodušuje rozpad úkolů či celých workflow na menší a snadněji uchopitelné části, čímž usnadňuje migraci z dnešního světa do světa Průmyslu 4.0 a zjednodušuje nasazení řešení 4.0 [113].



Obrázek 4.1: RAMI 4.0 model [113]

Vertikální osa modelu definuje šest samostatných vrstev, na kterých jsou specifikovány ty aspekty, ze kterých je modelovaný problém nahlížen (mapování datových toků, popisy funkcí, komunikační schopnosti, či integrace jednotlivých komponent). Toto členění koresponduje s principy dekompozice komplexních projektů např. v IT a dává možnost konzistentně vytvořit a udržovat šest dobře uchopitelných vzájemně propojených modelů v odlišných doménách. Vzhledem k tomu, že model podporuje zapouzdření submodelů na dalších osách, je možné flexibilně vytvářet popisy libovolně složitých integrovaných systémů.

Jak již bylo řečeno, životní cyklus výrobku je velmi důležitým a sledovaným aspektem v moderním inženýringu. Na *levé horizontální ose* modelu RAMI 4.0 je uveden životní cyklus výrobku s jeho hodnotovým tokem. Jsou na ní vyjádřeny takové závislosti, jako sběr dat a jejich využití v průběhu vývoje, výroby či provozu zařízení či výrobku. Totální digitalizace celého řetězce vývoj - prodej nabízí velký potenciál pro zlepšení produktu, stroje a dalších úrovní Průmyslu 4.0 architektury během celého životního cyklu. Tento pohled koresponduje se standardem IEC 62890, popisujícím základní principy systémů pro správu životního cyklu produktů.

Pravá horizontální osa pak popisuje funkce jakékoli fyzické či virtuální komponenty Průmyslu 4.0. Jsou na ní specifikovány funkční vlastnosti, nicméně ne jako konkrétní implementační předpisy, ale pouze popisy souboru funkcí. Tato osa respektuje stan-

dardy IEC 62264 a IEC 61512 (ty jsou však určeny pro specifikaci komponent pouze v jednom podniku nebo na jedné dílně). [127]

Nyní je na místě popsat vrstvy na jednotlivých osách ve větším detailu [113, 127].

4.1.1 Integrovaná osa RAMI 4.0

- Vrstva strojů (*Asset layer*) (*Jak mohu zabudovat svůj produkt do reálného světa tak, aby mohl realizovat požadované výrobní operace?*)

Vrstva strojů reprezentuje fyzickou podstatu bez jakékoli schopnosti digitálně komunikovat - realitu, např. fyzické komponenty jako jsou lineární osy stroje, kovové části, dokumentaci, použité materiály či obvodová schémata. Na této vrstvě se modelují také osoby vstupující do procesu. Fyzické části výroby (*assets*) jsou spojeny s integrovanou vrstvou např. prostřednictvím QR kódů.

- Integrovaná vrstva (*Které části mého produktu budou přístupné v digitálním světě?*)

Integrovaná vrstva zajišťuje sběr informací o objektech fyzické vrstvy a jejich digitalizaci tak, aby je bylo možné dále digitálně zpracovávat. Je tedy reprezentována např. senzory, či prostředky automatické identifikace (RFID), případně rozhraním *člověk-stroj* pro integraci osob - operátorů. Každá důležitá událost, která nastane ve fyzické vrstvě má prostřednictvím integrovaná vrstvy svůj obraz ve virtuálním světě. Pokud v reálném světě dojde ke změně, je tato reportována do integrovaná vrstvy, kde je zpracována a následně poskytnuta komunikační vrstvě pro odeslání.

- Komunikační vrstva (*Jakým způsobem bude zabezpečen přístup k nezbytným datům?*)

Komunikační vrstva reprezentuje standardizované komunikační protokoly a infrastrukturu, která umožňuje vzájemnou komunikaci mezi jednotlivými entitami na úrovni integrovaných vrstev a navzájem mezi integrovaná a informační vrstvou.

- Informační vrstva (*Jak bude můj produkt pracovat?*)

Informační vrstva definuje formální pravidla, podle nichž je prováděno předzpracování událostí z integrovaná vrstvy. V této vrstvě jsou příchozí události z integrovaná vrstvy s využitím perzistentních dat (např. předchozích stavů entity, tedy implementací stavového automatu) transformovány na rozhodnutí, která jsou dále poskytována funkční vrstvě. Dalšími funkcemi informační vrstvy jsou pak udržení integrity a konzistence dat, získání nových dat vyšší kvality či poskytování strukturovaných dat prostřednictvím servisního rozhraní.

- Funkční vrstva (*Jaké funkce bude můj produkt zabezpečovat?*)

Funkční vrstva umožňuje formalizovaný popis funkcí a vytváří platformu pro horizontální integraci různých funkcí. Funkce jsou v této vrstvě propojovány prostřednictvím pravidel a rozhodovací logiky (i když ta může být v některých případech

obsažena také v informační či integrační vrstvě). Vrstva dále obsahuje prostředí pro běh a modelování služeb a podporu obchodních procesů. Obsahuje také prostředí pro běh celých aplikací. Pro udržení integrity dat je vhodné realizovat vzdálený přístup a horizontální integraci právě na úrovni této vrstvy.

- Obchodní vrstva (*Jak jsou definovány a implementovány mé obchodní procesy?*)

Obchodní vrstva zajišťuje integraci funkcí v hodnotovém toku, umožňuje jejich mapování do obchodních modelů a tedy vytvoření a správu obchodních procesů. Pokrývá zákonné a regulační podmínky prostředí a umožňuje modelování pravidel, která musí systém respektovat. Tato vrstva také tvoří spojení mezi různými obchodními procesy a orchestrací služeb funkční vrstvy. Obchodní vrstva již nezahrnuje konkrétní softwarové prostředky (jako je např. ERP či MES systém, ty jsou typicky součástí funkční vrstvy).

4.1.2 Osa hodnotového toku RAMI 4.0

Levá horizontální osa modelu RAMI 4.0 reprezentuje životní cyklus a hodnotový tok průmyslové produkce. Tato osa je rozdělena na dvě části - *typ* a *instanci*. Typ každého produktu, stroje nebo SW/HW reprezentuje návrh/design. Typ obsahuje pravidla výzkumu, vývoje a testování až po výrobu prototypu. Po všech testech a validaci výsledků testů je typ připraven pro sériovou výrobu. Každý vyráběný produkt reprezentuje instanci tohoto typu, např. má jedinečné sériové číslo. Pro zákazníka jsou produkty nejprve pouze typem (obrázkem v katalogu). Instancí se stanou teprve v okamžiku nákupu a dodání, kdy dojde k jejich instalaci do nějakého stroje, přístroje apod. Jemnější struktura životního cyklu a hodnotového toku, jak je vidět na levé horizontální ose, ukazuje rozdělení na fázi *vývoje* a *nasazení/správy*, ale vzhledem k fyzikálnímu charakteru problému má instance fázi *produkce* a *údržba/využití-provoz*.

Funkci levé horizontální osy lze demonstrovat na jednoduchém příkladu: Vývoj nového elektrického pohonu reprezentuje tvorbu nového typu stroje. Pohon (elektrický motor s řídicím systémem) je v rámci vývojové fáze vymyšlen, vyvinut, jsou vyrobeny funkční vzory, jsou oživeny a testovány a jsou vyrobeny první prototypy a ověřeny jejich vlastnosti (validace). Po úspěšných testech je nový typ pohonu puštěn do prodeje (je uveden v katalogu produktů výrobce). V tom okamžiku může být odstartována první sériová výroba. Každý pohon v sériové výrobě dostane své unikátní sériové číslo a je instancí tohoto nově vyvinutého pohonu. Zpětná vazba od spotřebitelů na instance tohoto typu mohou vést ke korekci např. v mechanické části pohonu a modifikaci SW v řídicím systému pohonu. Taková modifikace je součástí *procesu správy typu*, tj. tyto změny jsou promítnuty do dokumentace jako vylepšení typu a podle ní jsou vyráběny nové vylepšené instance pohonu. Veškeré uvedené komunikace jsou založeny na využití sítí s přidanou hodnotou¹.

¹ *Value-added networks* - popisuje nejen přenosovou infrastrukturu, ale také služby, které jsou nad touto infrastrukturou provozovány a v určitém smyslu představují *přidanou hodnotu*.

Levá horizontální osa RAMI 4.0 také reprezentuje hodnotový tok. Digitalizace a propojení hodnotového toku představuje rovněž velký potenciál pro vylepšování stávajících typů. Hodnotový tok v totálně digitalizované výrobě umožňuje propojení nákupu, plánování, montáže, logistiky, údržby, například:

- logistická data mohou být využita v montáži a sestavování, nákup vidí zásoby v reálném čase a vidí, kde jsou které části od dodavatelů v určitém momentě, zákazníci vidí výrobní fáze výrobku;
- dodavatel může kontinuálně sledovat parametry každé nasazené instance zařízení a tato data předávat výrobcí. Ty slouží jako podklad ke zkvalitňování typu produktu;
- v případě poruchy je zařízení předáno k opravě výrobcí. Pokud se jedná o systematickou závadu konkrétního typu, může výrobce upravit vlastnosti a vyrábět dále upravené instance, které nebudou k této závadě náchylné atd.

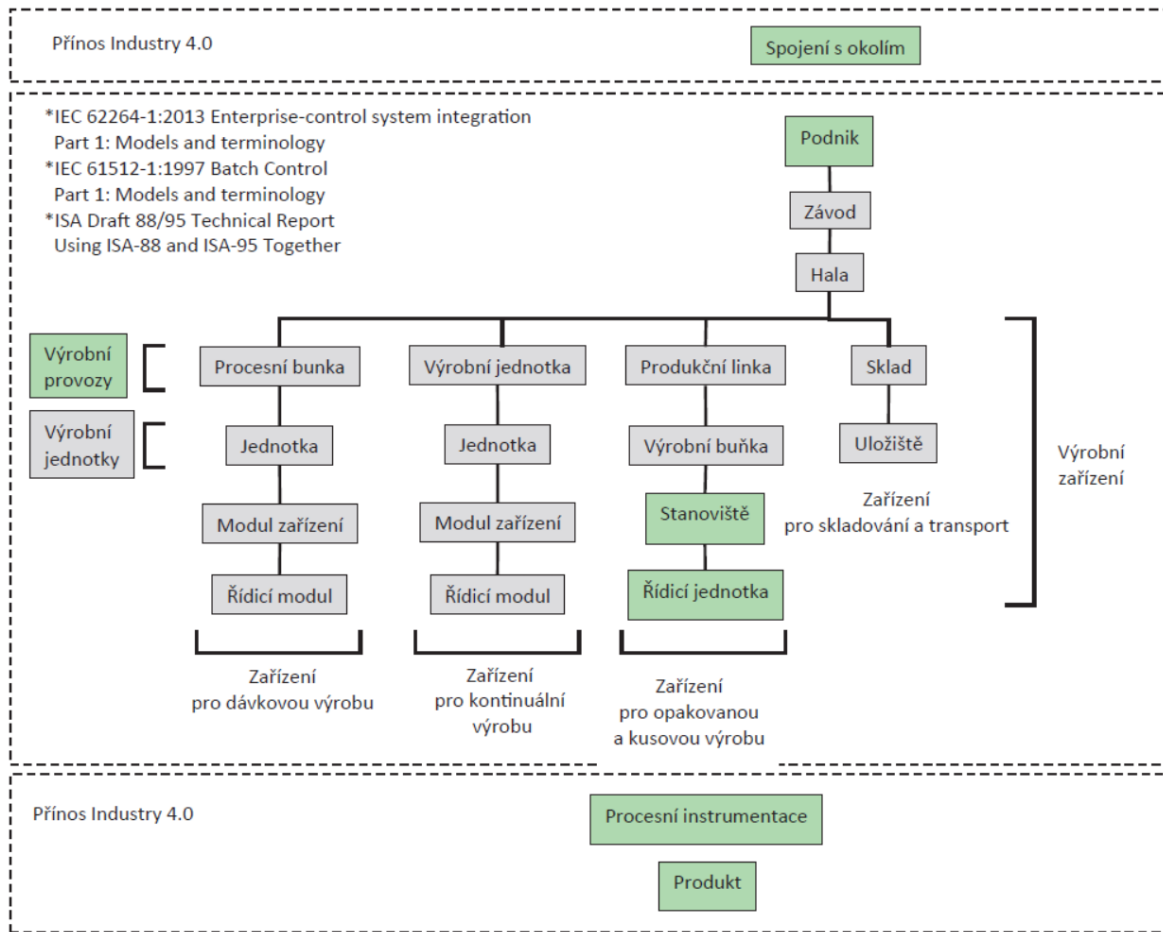
Uvedené příklady, a mnohé další možnosti představují významný potenciál pro zlepšování. Životní cyklus výrobku může být z toho důvodu viděn v souvislostech s procesem tvorby přidané hodnoty, který obsahuje, a ne izolovaně, jak je tomu v současné průmyslové produkci.

4.1.3 Osa hierarchie RAMI 4.0

Průmysl 4.0 přináší změny i do klasické hierarchické řídicí struktury (původní pyramidy) komplexních systémů ve všech typech výrobních procesů. Poslední osa modelu proto definuje propojení entit v podniku a byla vytvořena dle standardů ISA 95 a ISA 88 (jak je zřejmé z obrázku 4.2). Dosah obou standardů je však omezen na střední část obrázku, tedy modelování od podnikové úrovně, přes různé typy výrobních provozů (diskrétní, spojitě či dávkové řízení nebo skladování), až po nejnižší entity - řídicí moduly, resp. řídicí jednotky, představující části systémů ovládající určitou malou část výrobního či technologického procesu, nebo snímající technologické veličiny. Úroveň popsané nad či pod těmito standardy reprezentují další rozšíření v kontextu Průmyslu 4.0. Je to jednak *procesní instrumentace*, čímž je zde myšlena nová vrstva autonomních zařízení (například chytrých snímačů), a *produkt samotný*.

Produkt může být nově nedílně vybaven nejen *balíkem* informací, ale také vlastní funkcionalitou využitelnou v průběhu výrobního procesu. V neposlední řadě pak RAMI popisuje také úroveň propojených spolupracujících továren *Connected World* - (propojení s inženýrskými firmami, dodavateli komponent i zákazníky).

Tato hierarchie netvoří pevný návod pro nasazení řídicího systému výroby a podniku, je pouze funkčním popisem komponent systému z pohledu Průmyslu 4.0.



Obrázek 4.2: Nová řídicí architektura dle RAMI 4.0 [127]

4.2 Model *Industry 4.0 component*

Druhým důležitým modelem platformy Průmysl 4.0 je *Industry 4.0 component* (dále jen I4.0 komponenta), která historicky vychází z modelu RAMI 4.0. Tento model byl opět vyvinut institucemi BITCOM, VDMA a ZVEI a jeho účelem je podat výrobcům a systémovým integrátorům návod, jak přistupovat k vývoji hardwarových a softwarových komponent, které mají být v souladu s principy Průmyslu 4.0. Model formalizuje popis vlastností kyber-fyzických systémů a procesů, hardwarových či softwarových komponent budoucí výroby. a tím otevírá možnost výměny dat mezi těmito (I4.0) komponentami [126].

Charakteristickým rysem I4.0 každé komponenty je administrativní prostředí (obálka) nazývané **Asset Administration Shell**. Každý potřebný objekt (stroj, výrobní zařízení, dokonce také pouze jedna z částí stroje, či jakákoliv fyzická i nefyzická entita, anglicky *Asset*) může být *obalen* Asset Administration Shellem (**AAS**). Spojením objektu a AAS obálky vzniká I4.0 komponenta. Popisované spojení pak umožňuje mezi jinými:

- Jednotným způsobem ve výrobním prostředí definovat konkrétní objekt, jeho

vlastnosti a možnosti, případně tento objekt identifikovat v síti.

- Uchovat důležitá statická či dynamicky vznikající data objektu. Může se jednat o data vzniklá v průběhu celého životního cyklu produktu (jak typu, tak instance), tedy např. konstrukční (CAD) data, manuály či měřené hodnoty.
- Poskytnout uložená data všem entitám obchodního řetězce.
- Mnoho dalšího, např. iniciovat komunikaci objektu s jinými virtuálními či fyzickými objekty a procesy za účelem dosažení určitého cíle².

Asset Administration Shell může nabývat tří obecných forem (viz obr. 4.3):

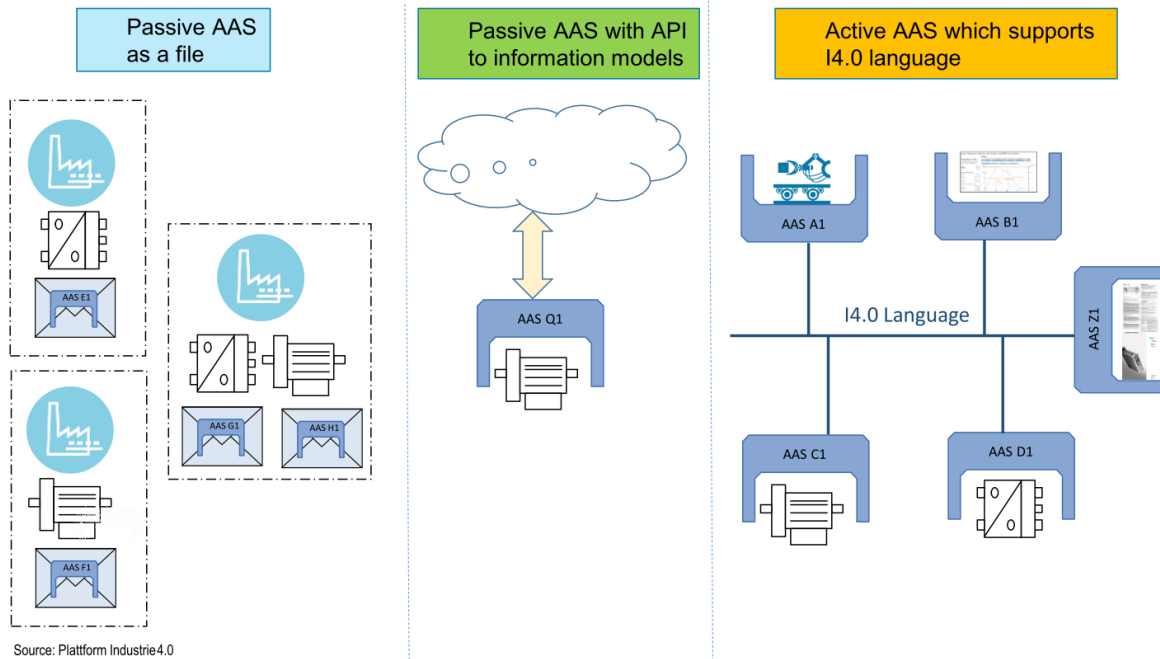
1. *Pasivní AAS jako soubor* je vlastně *digitální objektový model*, který standardizovanou formou popisuje příslušný *asset*, tedy informace o jeho typu a instanci. Takové AAS je možné pro přenos serializovat např. do formátu XML, JSON, a jeho hlavním přínosem je usnadnit interoperabilitu zařízení vyvíjených různými průmyslovými partnery [36].
2. *Pasivní AAS s API přístupem* může nabízet data svého objektového modelu nadřazeným entitám (v rámci vertikální integrace) prostřednictvím standardizovaného interface (většinou se využívá REST API s operacemi typu *CRUD*³). Toto AAS se někdy označuje jako *reaktivní*, neboť může reagovat požadavky okolí a poskytovat jak statická data, tak např. i měřená data (paralela s digitálním stínem (viz kapitola 2.4.2) pořízená v průběhu celého životního cyklu entity (produktu, stroje). Sémanticky precizně definovaný strojově čitelný popis schopností a možností jednotlivých entit pak tvoří základní krok vedoucí směrem ke zvýšení interoperability a integrace všech komponent do jediného celku [29].
3. *Aktivní AAS* svými možnostmi vychází z pasivního AAS (*neexistuje aktivní AAS bez pasivní části!*), avšak navíc k možnostem komunikace prostřednictvím API přidává možnosti horizontální spolupráce s dalšími AAS prostřednictvím univerzálního komunikačního jazyka. Aktivní část AAS pak obsahuje určitou míru autonomie, kterou je možné modelovat na principu agentních systémů.

Aktivní AAS, který přináší interoperabilitu zařízení různých výrobců, realizuje v reálném čase shromažďování potřebných dat objektu a dokáže je poskytovat je ostatním, a který dosahuje určité formy autonomie, může být považován za implementaci *digitálního dvojčete* v moderním průmyslovém prostředí. Pro možnost takové spolupráce je nezbytné, aby celá síť I4.0 komponent byla strukturována tak, že každá komponenta bude mít možnost propojit se se všemi ostatními. Aktivní části AAS musí rovněž ctít společný sémantický model.

²Podpora takového chování však není v AAS předepsána, jedná se o speciální případy, které budou popsány dále.

³create-read-update-delete

Celý koncept I4.0 komponenty, byť byl představen v roce 2015 je relativně nový a určitě si neklade za cíl nahradit všechny funkční a prověřené postupy používané ve výrobním prostředí. Není tedy nutné, aby bylo najednou každé nové zařízení vyráběno jako autonomní I4.0 komponenta, přímo zahrnovalo administrativní obálku, nebo disponovalo možnostmi autonomního chování [2].



Obrázek 4.3: Tři formy AAS [36]

4.2.1 Transformace objektů na I4.0 komponenty

Pro přímou implementaci alespoň reaktivního AAS musí konkrétní objekt (výrobní stroj, robot, atd.) disponovat minimálně komunikačním rozhraním a pasivními schopnostmi komunikace (v případě volby REST API je potřebná implementace HTTP serveru). Není ale bezpodmínečně nutné, aby byl objekt kompatibilní s I4.0 komponentním modelem.

V případě nových zařízení postavených na výkonných řídicích systémech, např. na bázi IPC může být komunikační obálka, zprostředkovávající funkcionalitu ve smyslu poskytování služeb, implementována přímo v nich.

V případě, kdy objekt nedisponuje dostatečnou výpočetní kapacitou, ať už z hlediska výkonu nebo kapacity datového úložiště, či neobsahují vhodné komunikační rozhraní (typicky se může jednat o běžné zařízení procesní instrumentace s podporou Profibus či Modbus komunikace), je možné provozovat jeho AAS obálku v jiném IT systému, který s objektem komunikuje např. prostřednictvím jeho podporovaného komunikačního rozhraní. Objekt může být dále vybaven např. externími snímači, které umožní měřit a ukládat informace o jeho stavu, včetně historie do AAS. V rámci jednoho IT systému s dostatečnou výpočetní kapacitou je možné provozovat AAS pro více

objektů.

Pokud entita nedisponuje potřebnými komunikačními schopnostmi, ani řídicím systémem, ve kterém by provoz AAS připadal v úvahu, je možné provozovat administrativní obálku ve společném repozitáři nadřazeného IT systému. Tato situace se, spíše než výrobních strojů, týká produktů samotných, které jsou nyní rovněž součástí spojené struktury. Na druhou stranu může nastat také situace, kdy se produkt po celou dobu výroby pohybuje např. na autonomním mobilním dopravníku, který disponuje jak potřebným výpočetním výkonem, tak komunikačním rozhraním, a tedy by byl schopen implementaci AAS bez problému hostovat.

Častá je také situace, při které jsou jednotlivé fyzické produkty ve výrobě stěží rozpoznatelné, a když, tak nanejvýš nalepenými čárovými kódy. I v takovém případě může být AAS vytvořeno, může vzniknout digitální popis produktu, jen musí být I4.0 komponenta implementačně přesunuta do společného repozitáře nadřazeného systému.

Samostatnou I4.0 komponentou mohou být také instalace software, duševní vlastnictví, či lidské zdroje. Zde je však komplikovanější situace při rozpoznání vhodnosti takového nasazení: pokud se jedná o řídicí software stroje, jehož existence samotná nedává v systému smysl, o I4.0 komponentu se jednat nebude. Rozdíl však je, pokud software stojí samostatně (např. nezávislý plánovací nástroj, či analytická služba) a je důležitým prvkem při provozu systému. Pak se již za samostatnou I4.0 komponentu považovat může.

4.2.2 Způsoby integrace I4.0 komponenty

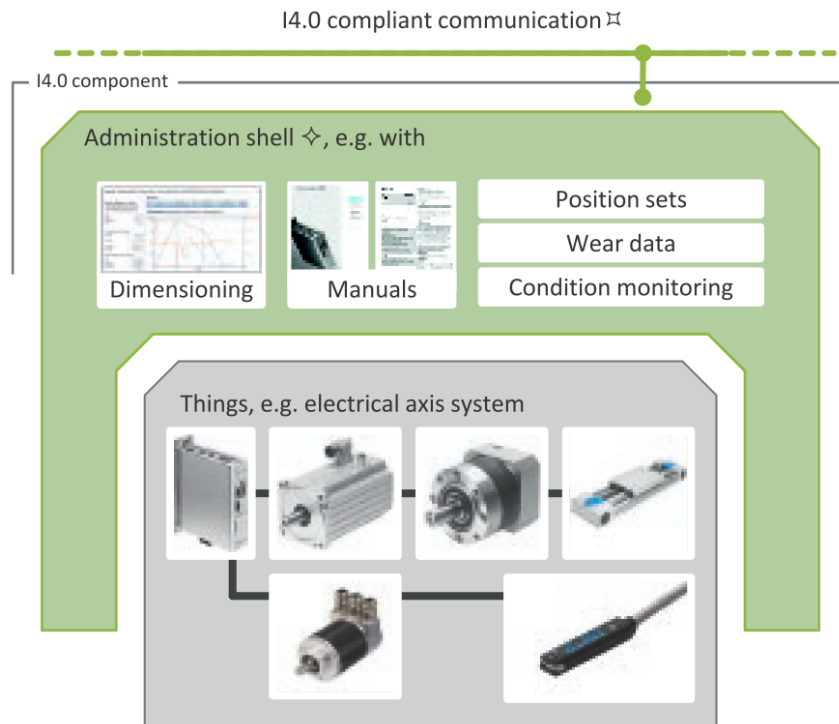
AAS mohou být vytvořeny dvěma způsoby. Tím prvním je vytvoření jediné I4.0 komponenty pro celý stroj. V takovém případě může být obálka hostována v řídicím systému stroje založeném např. na moderním PLC s možnostmi OPC-UA komunikace.

Druhou možností je zapouzdřit strategické části stroje řízené samostatnými kontroléry jako vnitřní I4.0 komponenty, které mohou samy plnit individuální strategii - např. být ve spojení s entitami zabezpečujícími konkrétní úkoly údržby. Obrázek 4.4 zobrazuje soustavu objektů, které dohromady tvoří technologický objekt - osu. V rámci návrhového software, využívaného při vývoji řídicích algoritmů (tedy během inženýrské fáze vývoje typu) dochází k integraci všech částí do jediného celku, který pro nadřazený systém vystupuje již konzistentně (funkcionalita je zapouzdřena a tváří se částečně jako černá skříňka). Jako celek je pak systém konfigurován při integraci (v inženýrské fázi nasazení instance).

Rozhodnutí, který z uvedených způsobů využít, závisí na charakteru systému - pokud zařízení jako celek funguje do vysoké míry uzavřeně a autonomně, není vhodné, aby jeho jednotlivé části byly obaleny vlastními AAS. Odpadne tím nutnost mapovat procesní hodnoty/příkazy a diagnostická data z vnitřních AAS na vnější obálku, celá

komponenta se zjednoduší nejen implementačně, ale také pro přístup externích I4.0 komponent.

Jindy však může být hierarchie vnitřních AAS naopak velice výhodná. Týká se to zejména případů, kdy jednotlivé části většího technologického celku mohou fungovat do vysoké míry samostatně a dává smysl rekonfigurovat a obsluhovat je zvlášť (zřejmě to může být v rámci řízení dávkových procesů, kdy má smysl např. pro každou samostatnou jednotku udržovat vlastní AAS z hlediska např. řízení i diagnostiky). Pak je možné na soubor jednotek nahlížet jako na jediný AAS, které logicky zapouzdřuje *vnitřní* I4.0 komponenty a zpřístupňuje je tak ostatním systémům. Kompletní popis všech částí systému ve formě AAS, a tedy zpřístupnění maximálního možného objemu provozních a diagnostických dat každého zařízení je také žádané z pohledu moderních end-to-end inženýrských nástrojů. Jako příklad je možné uvést možnost rychlého provádění rekonfigurace vnitřních parametrů jednotlivých částí stroje, což lze provádět pouze v případě, kdy jsou tyto parametry zpřístupněny (a to právě nejlépe ve formě logicky zapouzdřené AAS komponenty).

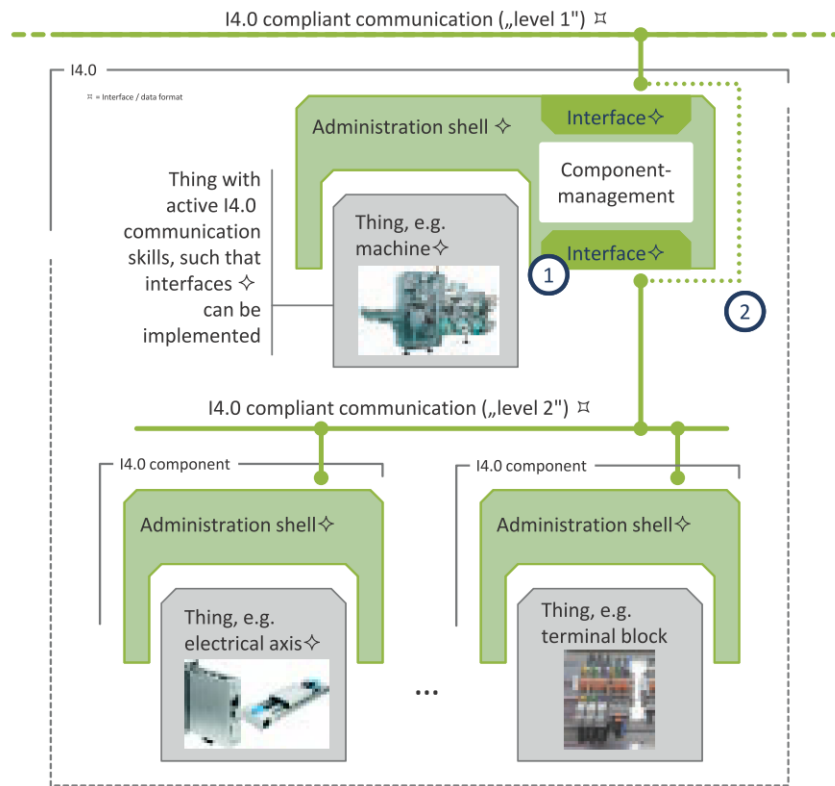


Obrázek 4.4: Jediný AAS pro celý stroj [1]

Existují dvě možnosti, jakým způsobem je možné technicky realizovat zapouzdření I4.0 komponent. Tou první, znázorněnou na obrázku 4.4 (varianta ①) je využití dvou nezávislých komunikačních rozhraní - vnějšího, kterým je *hlavní* logická komponenta propojená do sítě I4.0 komponent a vnitřního, které propojuje jednotlivé zapouzdřené komponenty v rámci technologického celku. V takovém případě existuje jasná logická i fyzická separace *hlavní* a podřízených komponent. Druhou možností ② je využití jediného rozhraní propojeného do globální sítě I4.0 komponent.

Zapouzdření jednotlivých komponent se však netýká pouze několika modulů je-

diného stroje či technologického procesu, ale hierarchickou strukturou jednotlivých do sebe zanořených AAS je možné obalit i vyšší celky dle ISA 95 - celou výrobní halu či továrnu [126].



Obrázek 4.5: I4.0 komponenta s více AAS [1]

4.2.3 Speciální případy mapování dat

Výše popisované hierarchické členění přináší mnoho možností, jak unifikovaně přistupovat k dílčím částem technologického celku. Bez větších obtíží si lze představit, že každá z těchto částí bude moci být navázána na I4.0 komponentu, která zprostředkuje diagnostické služby, případně vyřeší požadavky na servis. Pokud se ale opět podíváme na technologický celek zvnějšku, může jistě nastat situace, kdy někdo (jiná I4.0 entita) položí otázku: *Funguješ?*

Jednoduchá otázka, nicméně odpověď až tak jednoduchá být nemusí. Samozřejmě pokud dojde v případě výše popisovaného stroje k poruše některého z motorů, je jasné, že odpověď bude jednoznačná. V celém systému I4.0 komponent se však může nacházet taková, jejíž činnost není pro funkčnost stroje v určité chvíli zásadní. Dojde-li např. na baličím stroji k vyprázdnění zásobníku fólie určité šířky (systém pro odvíjení a měření bude vytvořen jako samostatná I4.0 komponenta), stále ještě může stroj bez obtíží fungovat s ostatními zásobníky.

Tento příklad ukazuje, že způsob mapování některých datových položek vnitřních AAS na *vnější* komponentu nemusí být jednoznačně globálně definovatelný. Pokud se

přidržíme pouze stavového modelu, je možné vymyslet např. poskytování stavu stroje nikoli jednou agregovanou hodnotou, ale s pomocí většího počtu příznaků, které umožní detailnější pohled na to, co přesně se aktuálně s technologickým objektem děje. Tyto příznaky je vhodné rovněž opatřit historizací, aby bylo možné v budoucnu statisticky detailně vyhodnotit stavy objektu a případné příčiny problémů. Jinou možností je v takovém případě formulovat dotaz jako *Popiš mi své schopnosti*.

4.2.4 Pasivní AAS

Odhlédneme-li od způsobu implementace a místa nasazení komponenty, je další klíčovou částí I4.0 komponenty virtuální reprezentace, kterou popisuje *Informační vrstva* modelu RAMI 4.0, a která zahrnuje data objektu. Tato data jsou strukturována do několika částí. Jednak jsou to povinná data objektu sloužící pro jeho identifikaci, zjištění jeho schopností, etapě životního cyklu, ve které se zrovna nachází, a jeho stavu. Další typy dat pak mohou zahrnovat CAD data či manuály, v neposlední řadě pak také soubory měřených hodnot. Propojení dat a funkcí objektu virtuální reprezentací formálně obstarává modul správy zdrojů⁴.

Důležitou částí virtuální reprezentace je *Manifest*, který lze chápat jako adresář metadat popisující strukturu dat AAS. Celá síť vzájemně propojených I4.0 komponent tvoří tzv. I4.0 repozitář (*vědomostní sklad*, který však není skladem v centralizovaném slova smyslu). Celá fyzická továrna pak je reprezentována v digitální podobě v tomto repozitáři prostřednictvím AAS a jejich spojů, které jsou udržovány a dynamicky aktualizovány během celého životního cyklu systému.

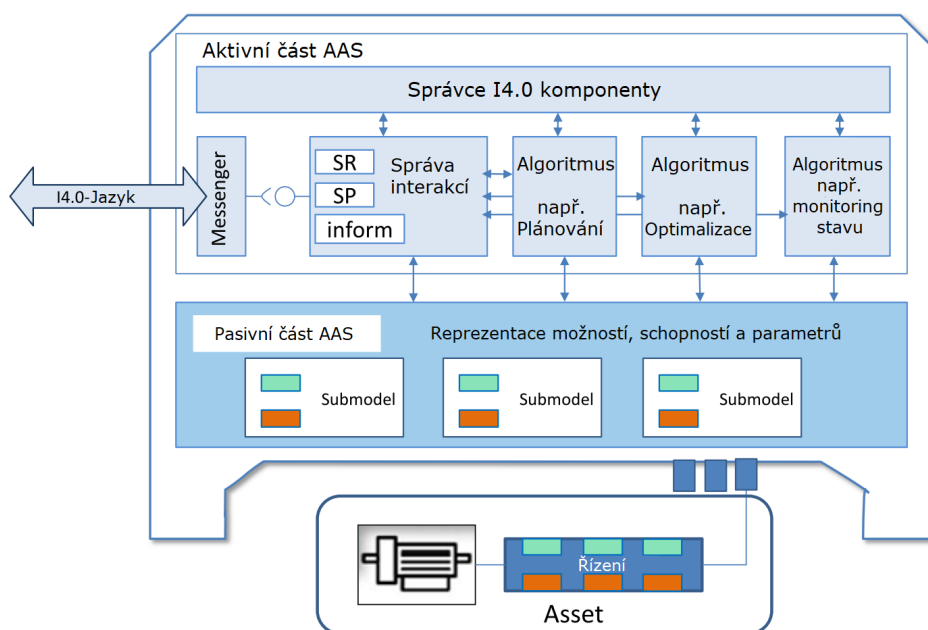
4.2.5 Aktivní AAS

Kromě dat v repozitáři, která zpřístupňuje svému okolí, může I4.0 komponenta obsahovat také autonomní funkcionalitu založenou na různých typech algoritmů. Taková funkcionalita bývá obecně modelována jako *služba* a entita, která ji nabízí, se označuje jako *Service Provider*. Entita, která službu potřebuje (poptává), se označuje jako *Service Requester*. Pro představu je možno zmínit konkrétní I4.0 komponentu - stroj sloužící k obalování vyrobených produktů ochrannou fólií. Administrativní obálka tohoto stroje:

- bude poskytovat identifikační informace a údaje o svých schopnostech okolním I4.0 komponentám prostřednictvím CRUD operací (pasivní AAS). Dále však
- může vyjednávat s obálkou jednotlivých produktů pořadí prací, tedy realizovat *adaptivní výrobu*;
- může komunikovat s obálkou jiného stroje, např. podavače materiálu (I4.0 komponentou) a zjišťovat, zda nedochází balící materiál, případně přizpůsobit svůj čas cyklu, tedy *dynamicky přizpůsobit výrobu aktuálním podmínkám*;

⁴Resource Manager

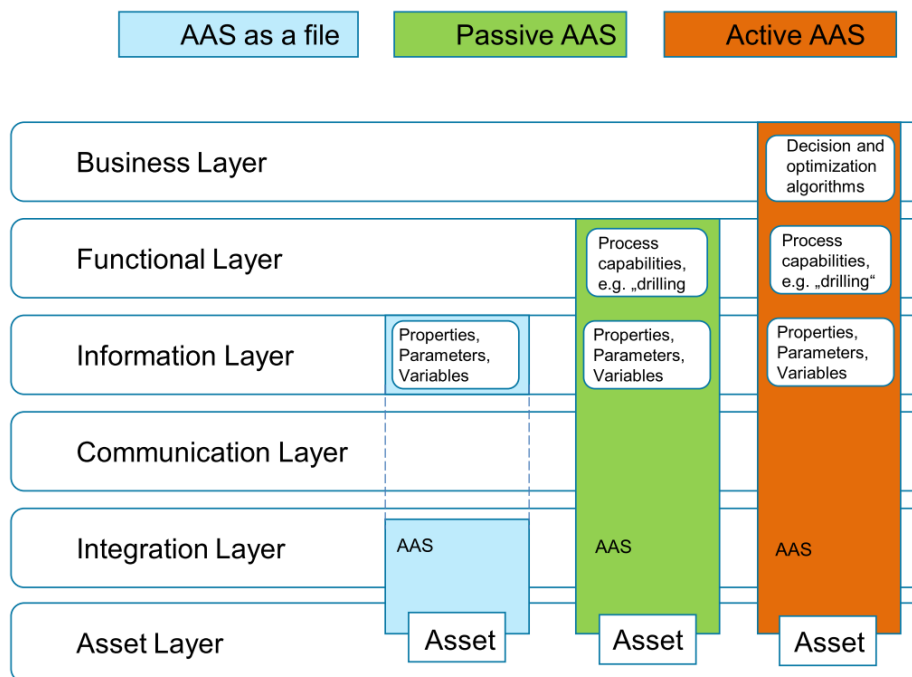
- může komunikovat se servisní entitou a odesílat systémová data k analýze za účelem odhalení blížící se poruchy a možnosti realizovat včasnou opravu, tedy *adaptivně předcházet poruchám a neplánovaným odstávkám*;
- může reportovat svá data KPI modulu systému MES;
- může ze systému PLM zjistit technologický postup realizace svého úkolu (rozměry a požadavky na balení konkrétního produktu);
- může dotazovat modul ERP systému s požadavky na další detaily právě zpracované objednávky.



Obrázek 4.6: Obecná architektura aktivního AAS [29]

Na obrázku 4.6 je znázorněn model architektury aktivního AAS. V modulu *Správce interakcí* jsou implementovány komunikační protokoly (dle specifikace *VDI2193 Sprache für I4.0-Komponenten*), které slouží pro interakci s okolními komponentami. Modul zároveň zprostředkovává rozhraní pro ostatní specifické aktivní moduly (plánovací/optimalizační algoritmy apod.). Ve druhé části již zmíněné specifikace (VDI2193-2) jsou definovány protokoly pro vzájemnou interakci, zejména tzv. protokol *Bidding process*, který není ničím jiným, než *Contract Net Protokolem* [57]. Jeho vlastnostem, výhodám i úskalím je detailně věnována část kapitoly 7. Do budoucna se předpokládá, že vzniknou a budou standardizovány další interakční algoritmy.

Modul *Messenger* zajišťuje transport zpráv. V termínech ISO/OSI komunikačního modelu probíhá komunikace mezi komponentami na úrovni uživatelské aplikace (tedy nad aplikační vrstvou) s využitím komunikačních protokolů, jako jsou MQTT či OPC UA. VDI2193 definuje, jaká by měla být struktura jednotlivých typů zpráv a jejich vhodný způsob serializace (data jsou serializována do *JSON*).



Obrázek 4.7: Mapování schopností jednotlivých typů AAS do RAMI modelu [29]

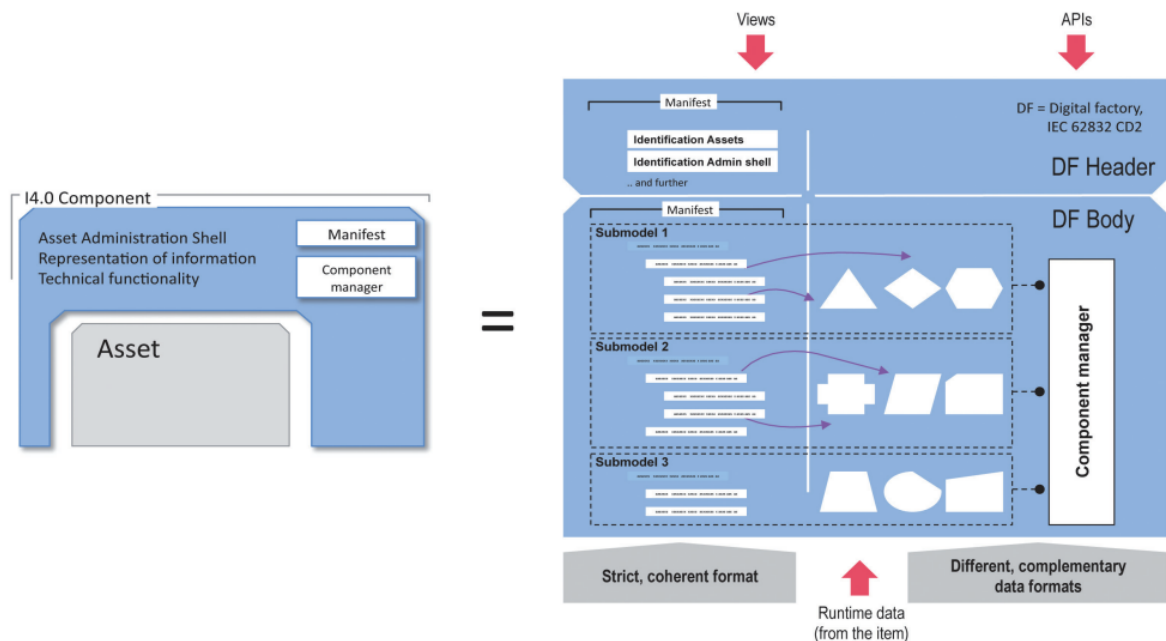
Modul *správce komponenty* pak řídí a synchronizuje jednotlivé části aktivního AAS. Implementace jednotlivých algoritmů závisí na konkrétní funkcionalitě celé I4.0 komponenty (viz dále v textu).

Na obrázku 4.7 jsou jednotlivé stupně implementace uvedeny v kontextu modelu RAMI. Pokud je AAS vytvořeno výrobcem zařízení jako balík statických dat, vidíme mapování hierarchické struktury vlastností, parametrů, proměnných, atd. pouze na úroveň informační vrstvy. V případě, kdy se dostáváme na úroveň pasivních komunikačních schopností, tedy do hry se dostává např. také vzdálené spouštění funkcí komponenty (viz dále), existuje již přesah do funkční vrstvy. Jakmile se jedná o aktivní AAS, které je schopno převzít iniciativu a autonomně interagovat pomocí speciálního jazyka (tzv. *I4.0 language*) s jinými AAS za účelem splnění určitého cíle (např. chovat se ekonomicky), je možné část mapovat funkcionality na *Business* vrstvu RAMI modelu. Není však nutné, aby všechna AAS disponovala takovou funkčností a většina jich zůstane v pasivní formě [29].

4.3 Základní struktura AAS

Každá obálka AAS se, jak je znázorněno na obrázku 4.8, formálně skládá z hlavičky a těla. Hlavička obsahuje identifikační informace týkající se obálky samotné a zapouzdřených objektů. Tělo je pak založeno na konceptu *submodelů*, z nichž každý charakterizuje objekt popisem jeho aspektů v různých doménách - např. *identifikace*, *komunikační schopnosti*, *inženýring*, *bezpečnost*, atd., a popisem funkcí jako *vrtání*, *frézování*,

montáž, ohřev, míchání apod.⁵



Source: ZVEI

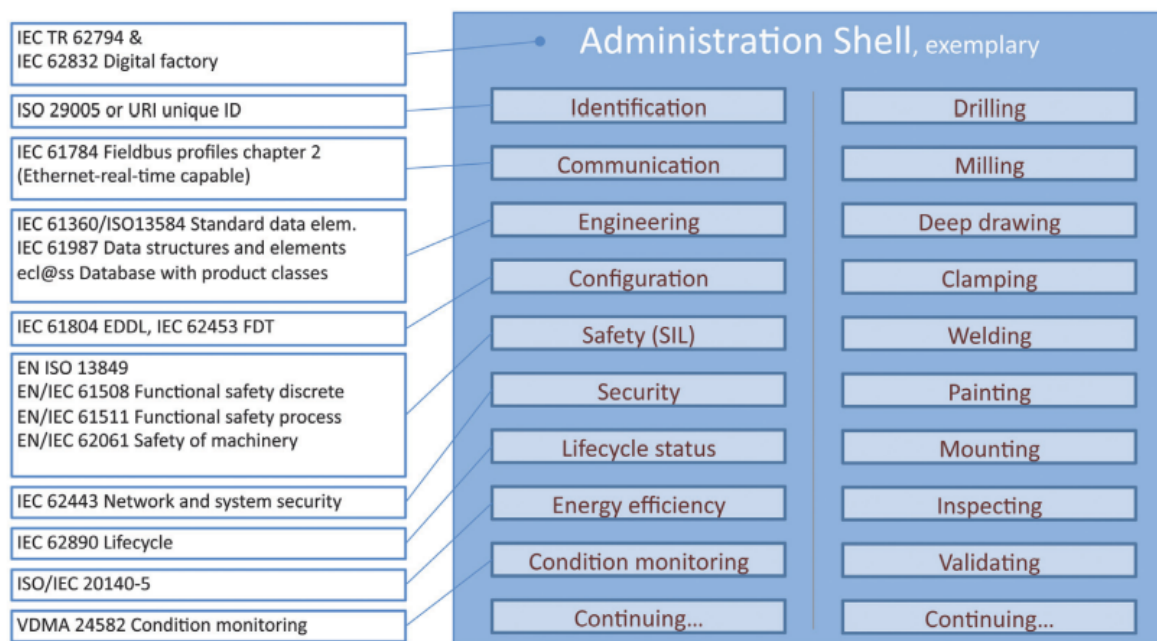
Obrázek 4.8: Struktura digitální obálky

Submodel můžeme definovat jako množinu informací ve standardizovaném tvaru. Těmito informacemi mohou v digitální obálce být např. *preferované jméno, symbol, fyzikální jednotka měření*, apod. Jak je zřejmé z obrázku 4.9, při specifikaci vlastností a stanovení jejich hodnot byla brána v úvahu řada platných norem IEC a ISO. Základem je myšlenka, že pro každý aspekt zařízení (doménu činnosti), jako je např. na obrázku zmiňované *vrtní*, existuje standardizovaný submodel, a v síti I4.0 komponent pak bude možné identifikovat stroj - vrtačku a její vlastnosti právě podle přítomnosti příslušného submodelu a parametrů, které obsahuje. Jinými slovy, existence submodelu ukazuje na existenci určité vlastnosti, či funkce zařízení.

Každá administrativní obálka musí mít implementovány tzv. *základní submodely* definované v [128]. Mezi ty se řadí např. *identifikační submodel* (viz obrázek 4.11). Dále je možné, aby každý výrobce zařízení implementoval specifický submodel. Cílem je rozdělit funkcionalitu zařízení na takový počet submodelů, aby každý zahrnoval a popisoval jeden z jeho aspektů.

Nic není ideální a bezpečné, a skutečný svět už vůbec ne. Proto nesmí zařízení *komukoli, kdo o to požádá* zpřístupnit kompletně všechny informace přítomné v submodelech. Struktura AAS a jeho implementace tak musí brát přirozeně v potaz i takové aspekty, jako je omezení viditelnosti a přístupu k submodelům, a také jejich částem na základě identity tazatele, neboť ne každý partner v rámci organizace (o to méně v rámci hodnotového toku) může mít plný přístup ke všem informacím.

⁵Detailní specifikaci AAS lze nalézt ve standardu [36]. Tento text pak často čerpá také z dokumentu [35] doplněného o příklady.



Obrázek 4.9: Výčet submodelů dostupných pro AAS

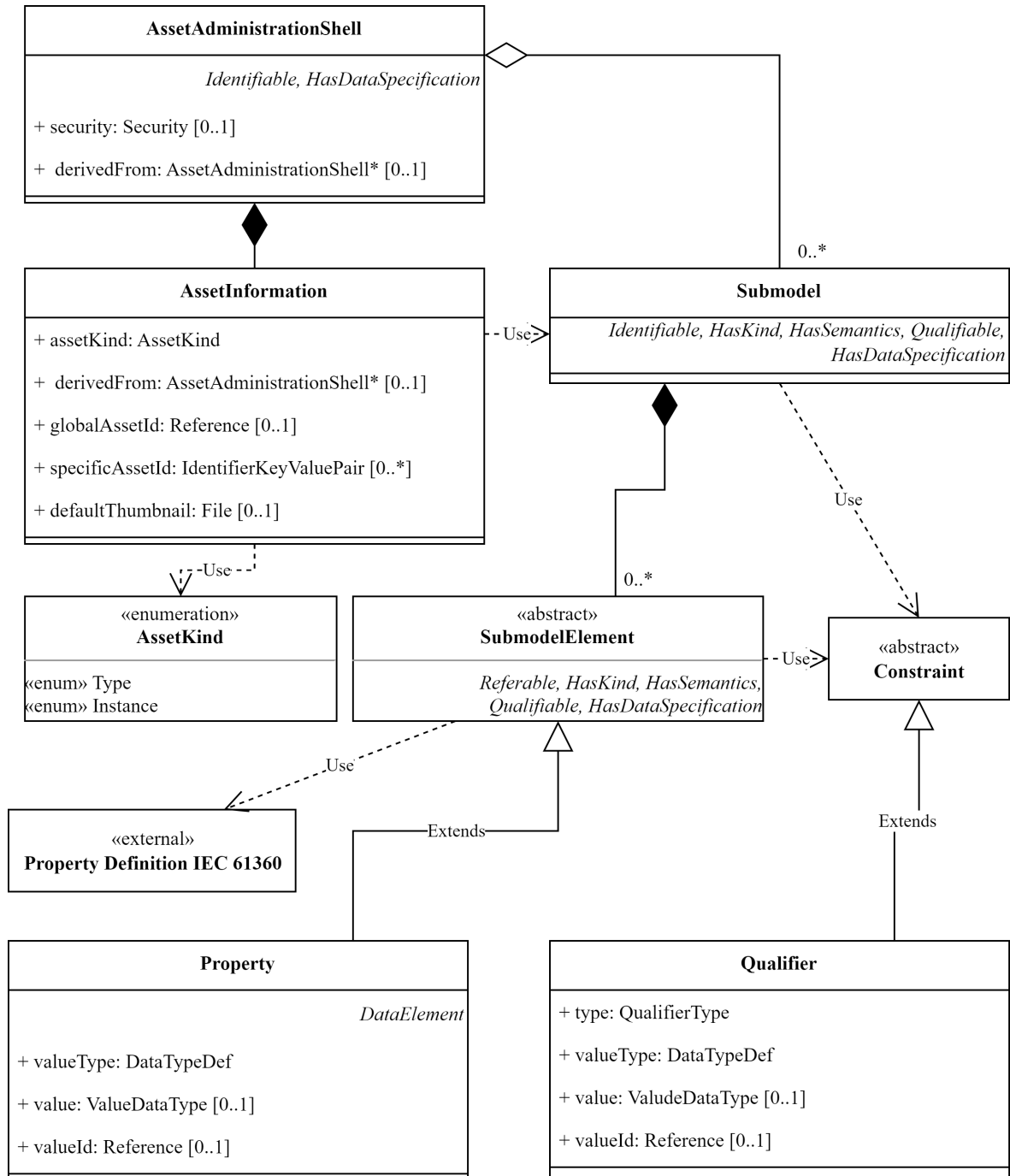
4.3.1 Meta Model AAS

Pro formální popis všech prvků modelu AAS a vazeb mezi nimi byl definován platformně nezávislý meta model. UML diagram, který jej popisuje, je uveden zde na obrázku 4.10. Můžeme vidět, že kořenovým elementem je *AssetAdministrationShell*. Na ten jen navázán model *AssetInformation* a následně kolekce $0..*$ submodelů různých typů (formálně ale všechny vycházejí ze stejného bazového elementu). Každý ze submodelů pak sestává z $0..*$ elementů různého typu. Takto sestavený model je možné serializovat do formátů XML, JSON či RDF. V současné době jsou některé z elementů standardizovány např. prostřednictvím slovníku ECLASS, popsaného v následující kapitole. Většina jich však zatím na standardizaci čeká [95].

V elementech submodelů jsou přítomny nejen objekty, které ukrývají užitečné informace, ale i ty, které zprostředkovávají veškerou interakci mezi AAS a vnějšími entitami. V [36] je specifikováno několik různých typů elementů:

- Hodnota reprezentující statickou vlastnost (tzv. *Property*) zařízení, např. katalogová data typu produktu, ale také např. sériové číslo konkrétní instance produktu. Vzhledem k tomu, že pro AAS je klíčové, aby mohly být definovány ve více jazykových mutacích, existuje také speciální typ vlastnosti (tzv. *MultiLanguageProperty*), pro kterou je možné zadat jednotlivé údaje ve více jazycích⁶.
- Procesní hodnota, telemetrická data či parametr, tedy dynamická data platná pro konkrétní instanci produktu mapovanou prostřednictvím modulu správy zdrojů.

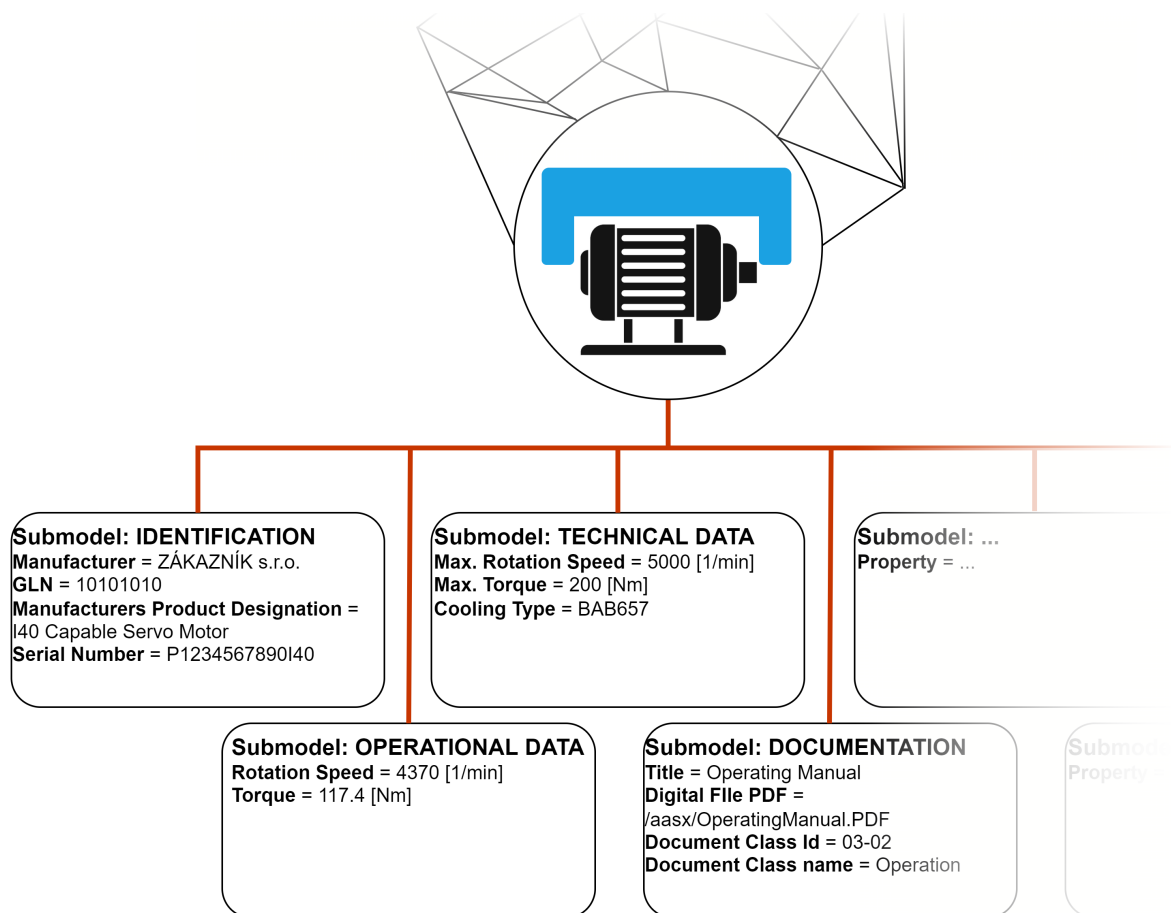
⁶Vícejazyčné popisy jsou zaznamenány ve formátu *popiskód_jazyka_dle_ISO639*, tedy např. *exampleEN*



Obrázek 4.10: Meta model AAS, dle [95]

- Operace, která může být externě spuštěna, a která vykoná v rámci komponenty určitou akci. Můžeme si představit, že AAS výrobního stroje bude nabízet operaci *Spust' výrobu*, která po jejím vyvolání způsobí spuštění konkrétního výrobního postupu. Typ spuštěného postupu (v případě, že jich stroj podporuje více) je možné rozlišit parametrem operace (který může být opět libovolným submodelem).
- Událost, která se může vyskytnout asynchronně (alarm) či být synchronně spuštěna. Rozlišují se dva typy událostí. Prvním typem jsou tzv. *zpětné* události, jejichž příkladem může být např. průběžné reportování stavu zařízení jeho výrobcí či oznámení výskytu chyby (asynchronně generované). Druhým typem jsou *dopředné události*. Příkladem dopředné události je např. aktualizace firmware stroje, která se promítne do aktualizace *typového* AAS, a kterou je AAS konkrétní instance schopen detekovat.
- Reference na externí datový zdroj či soubor, to znamená možnost propojení s informacemi uloženými v externích úložištích. Mimo to je rovněž přípustné ukládat binární objekty (tzv. BLOB) přímo v submodelu.
- Reference na jiný AAS, či jeho část (submodel, vlastnost), a to jak uvnitř podniku, tak i v rámci hodnotového řetězce. Mezi jinými lze této možnosti využít právě pro vybudování kompozitní komponenty, která agreguje více I4.0 komponent.
- Popis technických schopností entity, tedy možnost popsat potenciál fyzické entity dosáhnout určitého cíle ve fyzickém či virtuálním světě. Tento element dosud formálně definován, můžeme zde však vytvořit analogii s *Capabilities* modelem definovaným v ISA 95, který popisuje, jak definovat schopnosti fyzického zařízení provádět určité třídy úkolů.
- Kolekce elementů kteréhokoli z uvedených typů. V rámci submodelu je možné definovat dokonce i komplexní struktury (seznamy, pole) charakteristik. Mimo výčet prvků obsahuje kolekce také informaci o tom, jestli může obsahovat duplicitní položky (tedy položky se stejnou hodnotou *semanticId* znamenající v důsledku odkaz na stejnou instanci). Další důležitou vlastností kolekce je příznak o tom, že se jedná o uspořádanou či neuspořádanou množinu.

Vidíme, že podstatnou podmínkou pro možnost existence a interoperability I4.0 komponent je, aby každý ze submodelů a všechny zanořené kolekce elementů i elementy samotné, byly sémanticky popsány takovým způsobem, aby jim mohli porozumět všichni potřební komunikační partneři. Vzhledem k obrovské variabilitě všech prvků takové hierarchie bylo nutné zvolit pro standardizovaný popis takový prostředek, který jej nebude limitovat. Německá *Platform Industrie 4.0*, zabývající se standardizací na poli Průmyslu 4.0, doporučuje pro tento účel využívat např. slovník ECLASS.



Obrázek 4.11: Příklad submodelů a vlastností pro I4.0 komponentu, dle [128]

4.3.2 Identifikace komponent

V síti I4.0 reprezentuje každý AAS buď konkrétní *asset* či množinu *assetů*. Každý z AAS se pak skládá ze submodelů, a tyto jsou naplněny jednotlivými elementy různých typů. Všechny entity v rámci této hierarchie musí být nějakým způsobem identifikovatelné, a to ze dvou důvodů:

1. Je nutné umět rozlišit jednotlivé entity, adresovat je a odkazovat se na ně.
2. Je nutné mít možnost vazby elementů a jejich externích formálních definic (např. jednotlivé *vlastnosti* submodelů na jejich sémantické specifikace).

Identifikace je vyžadována přinejmenším pro *asset administration shell*, *asset* (identifikátor uveden v zapouzdřujícím AAS), a *submodel* (platí pro typ i instanci). Není však předepsán konkrétní typ identifikátorů, existují pouze doporučené standardy:

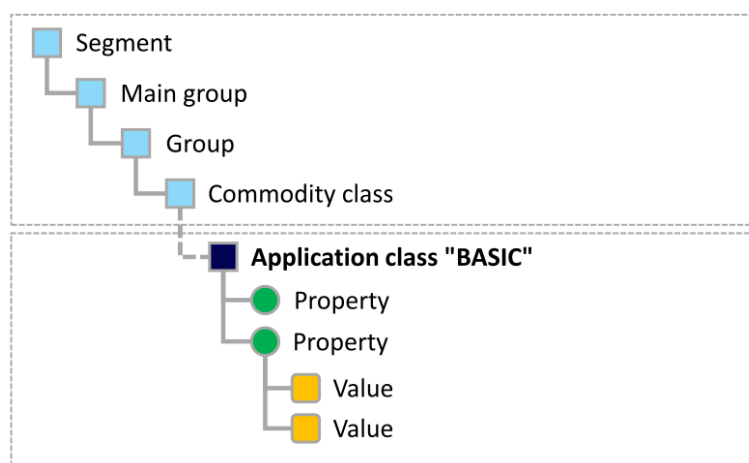
1. *IRDI* dle ISO29002-5, ISO IIEC 6523, resp. ISO IEC 11179-6 jako sémantické schéma pro jednotlivé *vlastnosti* na jedné straně, ale také celé AAS na straně

druhé. Identifikátor obsahuje spojení identifikace registrační autority (kód, identifikátor, identifikátor součásti, spolu s dalšími informacemi), vlastní identifikaci položky (třídy, vlastnosti, jednotky, způsobu reprezentace, typu dat, apod.) a kód verze. Do této skupiny patří široce rozšířený ECLASS.

2. *URI* vycházející ze specifikace RFC 3986, který je vhodný pro identifikaci AAS, *assetů* a dalších entit, které jsou globálně přístupné.
3. *Vlastní* způsob identifikace, např. pomocí GUID, který je ale vhodný pouze pro interní využití v rámci komponenty.

ECLASS

Cílem asociace ECLASS, která dnes sdružuje okolo 150 společností napříč všemi typy průmyslové výroby společně s veřejnými institucemi, je poskytnout metodu pro zjednodušení elektronické výměny dat pomocí jejich standardizovaných popisů. Standard nabízí způsob, jakým lze jednoznačně identifikovat entity, služby a jejich vlastnosti, a to nezávisle na výrobci zařízení, typu výroby či zemi původu. Standard v současnosti specifikuje cca 45 000 tříd produktů uspořádaných do čtyř hierarchických úrovní a celosvětově jej využívají tisíce organizací [34]. ECLASS byl původně navržen pro klasifikaci obchodovaných produktů, komponent a služeb.



Obrázek 4.12: Hierarchie ECLASS [34]

Na nejvyšší úrovni hierarchie (viz obrázek 4.12) stojí nejobecnější *segmenty*, které popisují jednotlivé domény zájmu. Na nejnižší - čtvrté úrovni hierarchie už vystupují třídy konkrétních produktů či služeb. Dále ECLASS specifikuje asi 20 000 vlastností, které jsou mapovány na tyto třídy. S nástupem Průmyslu 4.0 a specifikací I4.0 komponenty společně s AAS byly do ECLASS integrovány také zásadní požadavky z této domény [95].

Jako příklad uveďme klasifikaci zařízení - **stabilizovaného síťového zdroje**. V ECLASS hierarchii [33] se jedná o segment *27 Electric engineering, automation, process*

control engineering. Řadí se do skupiny 27-04 *Power supply devices*, podskupinu 27-04-07 *Power supply device* a třídu produktu 27-04-07-04 *AC-DC supply*.

Tato třída obsahuje seznam cca šedesáti (ve verzi ECLASS 13.0) vlastností charakteristických pro síťový zdroj, jak jsou geometrické rozměry - *instalační výška, šířka, hloubka*, rozsahy napětí *min. output voltage - max. output voltage*, a to jak pro AC, tak pro DC část a pro několik potenciálních kanálů zdroje, *informace o výrobcí* a mnoho dalších.

Pro každou z vlastností jsou pak popsány její sémantické parametry. Pro příklad uvedme vlastnost *max. 1. output voltage with AC*:

Preferred name	max. 1. output voltage with AC
Definition	Maximum effective value of the voltage specified by a digit applied to the windings of a transformer or voltage transformer which takes up the power that is to be transformed
Type	REAL_MEASURE
IRDI	0173-1#02-AAB774#008
Unit	volt (V)

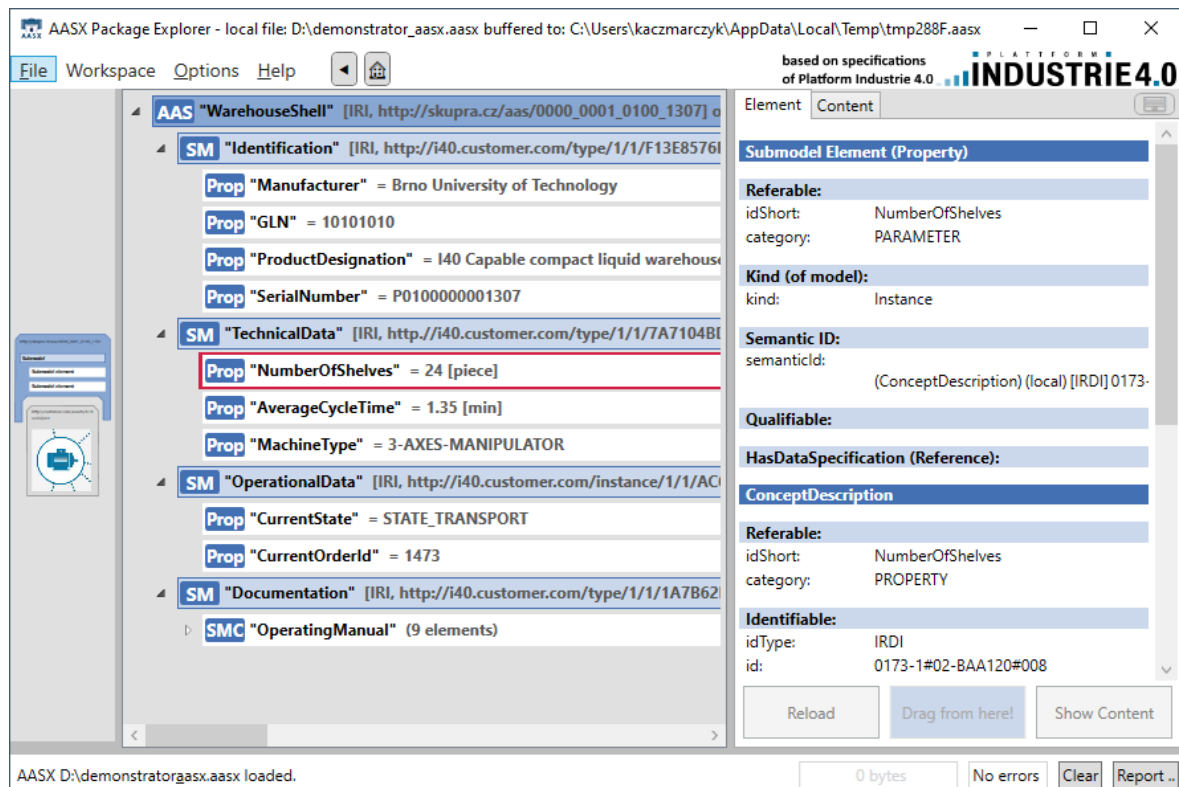
4.4 Možnosti implementace AAS

V současnosti existuje několik softwarových nástrojů, pomocí který je možné vytvářet nová AAS.

4.4.1 AASX Package Explorer

Jedná se o open-source nástroj s grafickým uživatelským rozhraním, který je určen pro implementaci a demonstraci prvků AAS. AASX Package Explorer aplikace obsahuje http server, pomocí kterého zpřístupňuje vytvořené AAS modely prostřednictvím REST rozhraní, stejně tak OPC UA server, do kterého lze nahrát již připravený model. Takto spuštěný server však umí hostovat pouze jedno AAS a v případě požadavku na paralelní běh více entit je nutné použít doplňkový software AASX Server [17].

Na obrázku 4.13 je představena ukázka skupiny AAS pro demonstrátor průmyslu 4.0 podrobněji popsány v kapitole 8.



Obrázek 4.13: AAS model výrobní buňky v prostředí AASX Package Explorer

4.4.2 Další nástroje pro modelování AAS

NOVA Asset Administration Shell

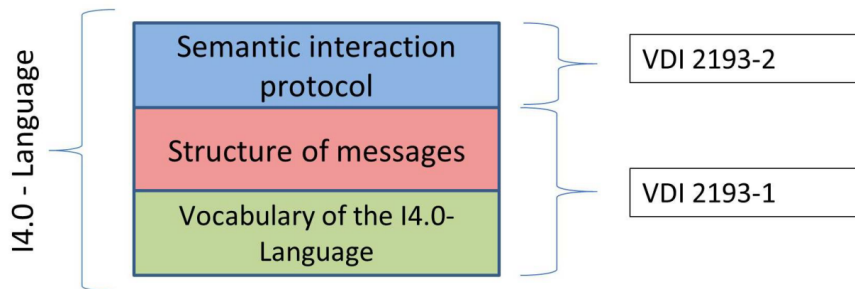
NOVA Asset Administration Shell (NOVAAS) je open-source software vyvinutý organizací NOVA School of Science and Technology používaný pro implementaci prvků AAS. Uživatel může AAS vytvářet pomocí vizuálního programovacího jazyku Node-RED založeného na principu konfigurací toků dat.

Eclipse BaSyx Python SDK

Eclipse BaSyx Python SDK, dříve známý pod názvem PyI40AAS, je knihovna implementovaná do programovacího jazyka Python, která se zaměřuje na implementaci AAS v souladu s metamodelem a rozhraním definovanými v [36]. V rámci knihovny je uživatel schopen vytvářet AAS a číst a zapisovat je do univerzálních souborů s příponou .aasx, které jsou čitelné také ostatními popsány nástroji. Dále je možné serializovat vytvořená AAS do souborů JSON a XML, které jsou již použitelné pro definici REST API na straně AAS serveru, validovat tyto soubory a zpětně je deserializovat.

4.5 Jazyk aktivních komponent

Interakce I4.0 komponent je založena na výměně zpráv. Formát zpráv je popsán v rámci první části specifikace VDI2193-1, kde jsou jednak definovány sémantické elementy, jejich význam, účel využití, typy datových položek, zda se ve zprávě musí vyskytovat povinně či ne, atd., a jednak jsou zde popsány různé typy zpráv jazyka I4.0 a jejich struktura (tedy vlastně z jakých sémantických elementů se skládají). Zprávy jsou dále serializovány do JSON formátu. Tato specifikace tvoří dvě nejnižší vrstvy pomyslné sémantické hierarchie (viz obrázek 4.14).



Obrázek 4.14: Koncepte jazyka I4.0

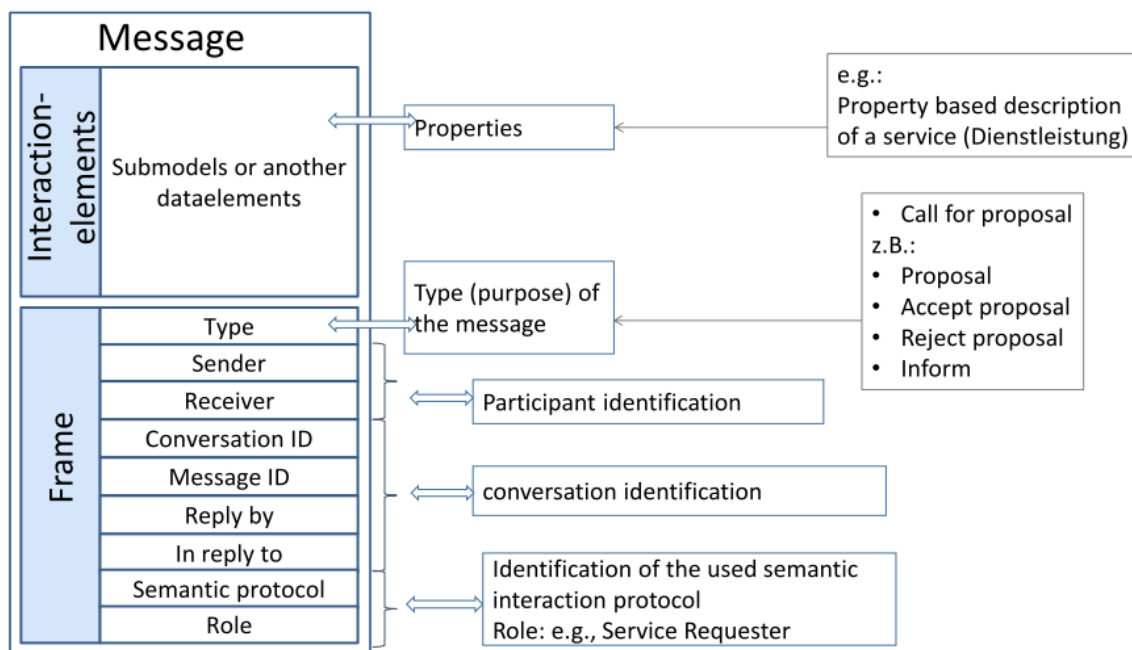
Zprávy jsou přenášeny v rámci relací - sekvencí zpráv, neboť pro naplnění cíle bývá nutné vyměnit si mezi entitami více, než jednu zprávu. Sekvence zpráv, neboli tzv. sémantické interakční protokoly jsou popsány ve VDI2193-2 [29]. Můžeme rozeznat dva typy interakcí I4.0 komponent mezi sebou. Jedním typem je horizontální integrace (tedy komunikace komponent na stejných úrovních = *jednotlivých výrobních entit a výrobků společně*), dalším typem je pak vertikální integrace (tedy komunikace *produktu se systémem řízení výroby*).

Horizontální integrace je výměna zpráv mezi zařízením poptávajícím určitou službu *service requester* a zařízením, které službu nabízí *service provider*. Nejčastěji se jedná o vyjednávání výroby pomocí nabídkového procesu. Typ a obsah vyměňovaných zpráv je definován ve specifikaci. Touto formou horizontální komunikace, procesem vyjednávání a rezervace se zabývá kapitola 7.1.2.

Vertikální integrace, je např. komunikace mezi vyráběným produktem či strojem a modulem pro záznam parametrů výrobního procesu MES systému. Je založena na způsobu komunikace klient-server, přičemž entita výše v hierarchii, je vždy klientem, zatímco entita, níže ve vertikální hierarchii pracuje jako server [50]. Komunikaci vždy iniciuje klient, např. MES modul pravidelně dotazuje stroj o aktuální výrobní data, která stroj poskytuje ve formě standardizovaných submodelů⁷.

Na obrázku 4.15 je uvedena struktura zprávy I4.0 jazyka. VDI2193-1 předepisuje způsob serializace zpráv a jako přípustný stanovuje *Java-Script Object Notation*

⁷Pokud je vertikální integrace implementována tímto způsobem, jedná se o pouze *reaktivní*, nikoli aktivní AAS.



Obrázek 4.15: Struktura zprávy I4.0 jazyka [29]

(JSON) formát. Pouze zprávy v tomto formátu je možné přenášet zvoleným komunikačním protokolem.

4.6 Prostředky komunikace mezi komponentami

Pro dosažení dynamické a flexibilní výroby je potřeba použít standardizovanou, výkonnou a v ideálním případě bezdrátovou síť, která zajistí propojení každého fyzického zařízení se všemi ostatními, a analogické propojení všech digitálních dvojčat. Základním požadavkem na průmyslovou komunikační síť je schopnost pracovat v reálném čase s minimální dobou odezvy. Ideálně bezdrátová komunikace má však své limity, a pro komunikaci mezi většími celky v současnosti zdaleka nedosahuje potřebných parametrů. Tempo vývoje 5G technologií do budoucna slibuje tento nedostatek eliminovat. V současnosti však požadavkům pro moderní průmyslové komunikace nejlépe vyhovují dva komunikační protokoly: *Message Queuing Telemetry Transport*⁸ (MQTT) a architektura *Open Platform Communications Unified Architecture* (OPC UA) [29].

Pro I4.0 komunikaci není nutno budovat dedikovanou infrastrukturu - je možné sdílet ji se stávajícími komunikačními protokoly, klidně pracujícími s určitou formou determinismu a reálného času (např. Ethernet Powerlink, PROFINET apod.). I4.0 komunikace by mohla sdílet s průmyslovým Ethernetem stejné médium, fyzickou a linkovou vrstvou a byla by provozována v asynchronních fázích jinak isochronního komunikačního cyklu [113].

⁸Název protokolu dnes již vlivem vývojových změn nerespektuje jeho skutečné vlastnosti

4.6.1 MQTT

Message Queuing Telemetry Transport (MQTT) je jednoduchý otevřený protokol pro výměnu datových zpráv založený na komunikačním modelu publisher/subscriber. Jeho historie sahá do roku 1999, kdy Andy Stanford-Clark (IBM) a Arlen Nipper (Arcom) stáli před problémem vyvinout úsporný protokol (z hlediska spotřeby energie i šířky pásma) pro satelitní přenos telemetrických dat získávaných měřeními na ropovodech, který by podporoval různé úrovně Quality of Service. V roce 2014 proběhla standardizace OASIS. Současná verze MQTT 5 byla standardizována v roce 2019 a dnes za protokolem stojí Eclipse foundation [42].

Architektura MQTT

Každé zařízení komunikující tímto protokolem může představovat odesílatele (*publisher*) a zároveň i příjemce (*subscriber*). Spojení pak zajišťuje zprostředkovatel (*MQTT broker*), který se zároveň stará o distribuci zpráv jednotlivým příjemcům. Zprostředkovatel zamezuje nutnosti přímé komunikace mezi odesílatelem a příjemcem, díky čemuž nemusí jednotlivá zařízení znát parametry ostatních (např. IP adresu). Zároveň spolu nemusí být jednotlivá zařízení synchronizována. Z toho plyne další výhoda, a totiž že odesílatel a příjemce nemusí být aktivní ve stejný časový okamžik. Nevýhodou je samozřejmě přítomnost centrálního členu, takže v případě výpadku brokeru není možná žádná komunikace.

Obsah zpráv samotných není ani pevně stanovený ani vyžadovaný, MQTT je tedy tzv. *payload agnostic*. Obsah je brán jako binární data, která jsou protokolem přenesena. Maximální velikost zprávy, necelých 256 MB v průmyslovém prostředí nepředstavuje nijak závažný problém. MQTT ke zprávě samotné přidává jen minimum servisních dat.

Témata zpráv

Zprávy v MQTT patří do určitých témat (*topic*), přičemž každá zpráva patří právě do jednoho tématu. Tato témata jsou hierarchická, v zápisu jsou oddělena lomítky, tedy např. teplotní snímač č. 2 na tanku č. 4 může mít téma `cell-1/tank-4/temperature-2`. Hierarchie není pevně určena a je možné přizpůsobit ji dle aplikace. Odesílatel, tedy ten, kdo zprávu publikuje, zvolí téma a pošle ho spolu se zprávou. Téma není nutno nijak zakládat nebo kontrolovat. Jakmile broker přijme zprávu pro téma, které ještě nemá, příslušné téma založí.

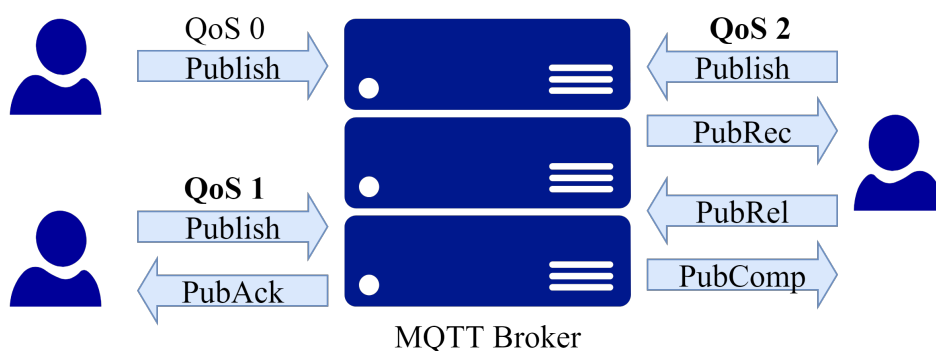
Znalost témat, ve kterých se bude publikovat je ale samozřejmě klíčová pro zamýšlené příjemce. Ti se během připojení k brokeru (a pak i kdykoli později) přihlašují k odběru témat, která chtějí odebírat prostřednictvím jejich názvů a hierarchie. Mohou při tom využít zástupné znaky # a +. Znak + nahrazuje jednu úroveň (tedy např. `cell-1/+/temperature-2` adresuje snímač č. 2 na všech tankách v buňce 1), znak # pak nahrazuje jednu či více úrovní a musí být vždy jako poslední (tedy např.

cell-1/tank-2/# adresuje všechny snímače na cell-1/tank-2).

Úrovně kvality služeb

MQTT popisuje tři různé úrovně *Quality of Service*. Kvalita služeb představuje dohodu mezi příjemcem a odesílatelem, která definuje záruku doručení jedné zprávy. Čím lepší kvalita služeb, tím vyšší je jistota, že zpráva příjemci přišla, a to právě jednou. Při doručování zprávy od odesílatele zprostředkovateli definuje úroveň kvality služeb odesílatel. Při doručování zprávy od zprostředkovatele příjemci se úroveň kvality může snížit na takovou, kterou je příjemce ochoten přijímat.

- Úroveň *QoS 0* - zpráva je odeslána a odesílatel nepožaduje žádné potvrzení o příjmu. Odeslaná zpráva se neukládá, odesílatel se již dále o nic nestará. Zpráva již nemůže být znovu odeslána. Jedná se o doručení s minimálním úsilím s tím, že zpráva bude doručena *nejvíce jednou*.
- Úroveň *QoS 1* - zpráva je odeslána a je zaručeno že bude doručena *alespoň jednou*. Odesílatel zprávu uloží, dokud nedostane od brokeru informaci o jejím přijetí (*PUBACK*).
- Úroveň *QoS 2* - zpráva je příjemci doručena *právě jednou*. Jedná se o nejbezpečnější, ale zároveň nejpomalejší způsob přenosu zabezpečující nejvyšší nabízenou kvalitu služeb. Po odeslání zprávy odesílatel čeká na potvrzení doručení příjemci - paket *PUBREC*. Odesílatel potvrdí, že informaci korektně přijal, je si vědom korektního doručení zprávy odesláním paketu *PUBREL*, po jehož odeslání opět čeká na přijetí zprávy *PUBCOMP*. Aby byla původní zpráva *QoS=2* považovaná za úspěšně odeslanou, musí celá tato výměna proběhnout do určitého času.



Obrázek 4.16: MQTT úrovně kvality služeb

Kromě QoS se u zprávy nastavuje i retain flag, tj. příznak, který říká, že broker nemá zprávu po rozeslání zahazovat, ale uložit a poslat novým odběratelům daného *topicu*. Posílá se vždy poslední uložená zpráva s příznakem retain.

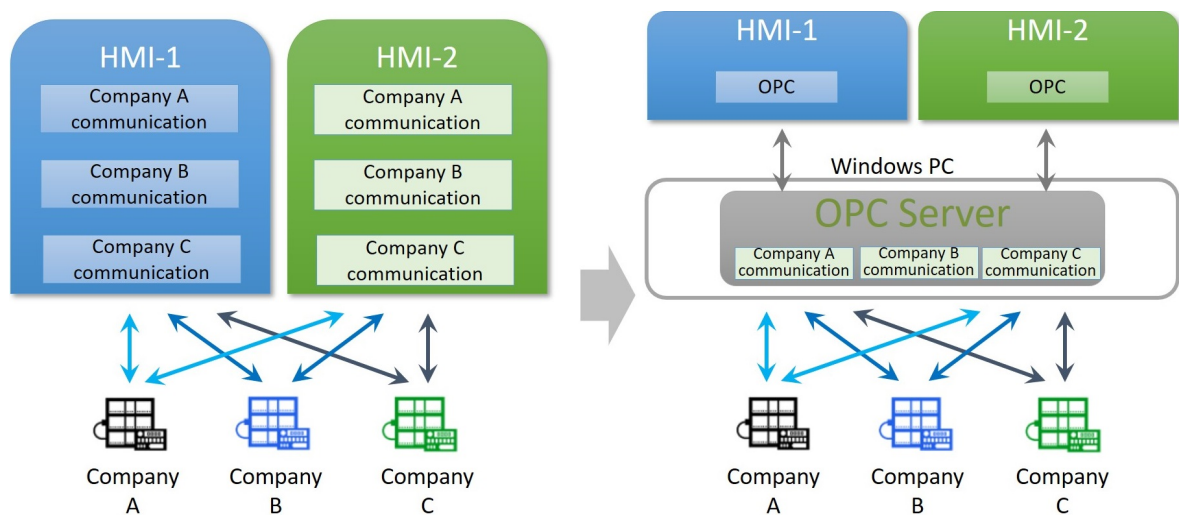
Výhody a nevýhody MQTT

Hlavní výhodou MQTT je bezsporu jednoduchá implementace a univerzálnost při implementaci do zařízení různých výrobců. MQTT navíc nemá pevně danou topologii, což umožňuje současné zapojení více brokerů, a tím zlepšení plynulosti přenosu informací.

Fakt, že MQTT nevyžaduje sám od sebe žádné zabezpečení na jednu stranu usnadňuje např. přidávání dalších brokerů, avšak jedná se o bezpečnostní hrozbu. Standardní broker do zpráv nijak nezasahuje, pouze je bez jakékoli kontroly přeposílá. Může se tedy stát, že budou přeposlána data, která jsou pro ostatní zařízení nečitelná, neboť jsou poškozená, mají neočekávaný formát nebo kódování [66, 42].

4.6.2 OPC

OPC je standard zajišťující interoperabilitu a bezpečnou výměnu dat mezi různými systémy v průmyslové oblasti. Jeho historie sahá k roku 1996, kdy byla konsorciem několika partnerů působících na poli průmyslové automatizace vytvořena první specifikace *OLE for Process control*⁹ (OPC). První vydání specifikace si kladlo za cíl vytvořit abstrakci existujících protokolů pro průmyslovou komunikaci (Modbus, Profibus, atd.) pro možnost jednoduchého využití dat, poskytovaných těmito protokoly v HMI/SCADA systémech. Vznikly tedy OPC konektory pro konkrétní protokoly na standardizované příkazy pro čtení/zápis různých typů dat (aktuálních dat, historických dat, stavů alarmů a událostí, a dalších) [37].



Obrázek 4.17: Srovnání situace před (vlevo) a po (vpravo) rozšíření OPC [90]

Pod standard OPC spadá několik částí:

⁹OLE (Object Linking & Embedding) je technologie vyvinutá společností Microsoft v roce 1990 pro sdílení objektů dat mezi procesy buď vytvořením kopie objektu (*embedding*), nebo propojením odkazem (*linking*).

- OPC DA (*Data Access*) - specifikace přístupu k procesním datům zařízení (hodnoty, časová značka, kvalita dat)
- OPC AE (*Alarms and Events*) - specifikace přístupu k asynchronním událostem, či alarmům včetně práce se stavy a stavovým modelem v zařízení.
- OPC HDA (*Historical Data Access*) - zavádí dotazovací metody a analytické funkce ke zpracování historických časových řad.
- OPC UA (*unified architecture*) - na rozdíl od předchozích tří specifikací založených na COM firmy Microsoft je postaven na obecně používaných standardech a integruje veškerou funkcionalitu předchozích specifikací.

Architektura OPC UA

OPC UA je moderní komunikační protokol postavený na architektuře orientované na služby (SOA), který je založen na předávání dat způsobem klient-server. To znamená, že server nabízí služby, na které se klient může dotazovat, přičemž server následně reaguje odpovědí. Služby poskytované serverem vytváří abstraktní komunikační model. Po připojení klienta k serveru se vytváří a udržuje zabezpečená relace. Ta je aktivní pro veškerou komunikaci.

Kompletní specifikace OPC UA definuje velké množství vlastností a funkcí, ale k tomu, aby se mohla rozběhnout základní výměna dat, stačí implementovat nutné minimum pro provoz a další funkce přidávat podle potřeby. Pro zjištění, které funkce jsou pro konkrétní aplikaci potřebné, existují profily, které obsahují seznamy vlastností. Daná aplikace poskytuje informace o tom, které profily podporuje. Tím umožní dalším aplikacím identifikovat, které části OPC UA využívá. Profily mohou obsahovat sady služeb, použitá kódování, způsoby zabezpečení a další volitelné části specifikace.

Komunikační model

Komunikaci v OPC UA obstarává komunikační model/zásobník OPC UA složený z transportní, komunikační a aplikační vrstvy.

- *Transportní vrstva* specifikuje komunikační protokol pro odesílání a příjem paketů. Dále implementuje šifrovací a ověřovací mechanismy sloužící k zajištění bezpečnosti přenášené zprávy. Při navazování spojení je transportní vrstva vytvořena jako první. Podporovány jsou protokoly HTTP/SOAP, HTTPS a TCP/IP.
- Výše postavená *komunikační vrstva* sestává ze dvou podvrstev - *Secure Channel Layer* a *Serialization Layer*. Zatímco první zmiňovaná sestavuje a po celou dobu relace udržuje zabezpečený kanál mezi serverem a klientem, druhá má na starost serializaci objektových dat do proudu vhodného k přenosu (existují dvě možnosti

- JSON formát nebo binární přenos). Způsob vytvoření zabezpečeného kanálu je závislý na použitém komunikačním protokolu. Pro každé připojení je potřeba použít oddělený kanál. Po vytvoření zabezpečeného kanálu se stanoví jeho identifikátor (Secure-ChannelId) a bezpečnostní token (SecurityToken), kterým se kanál identifikuje.

- Poslední *aplikační vrstva* vytváří a udržuje relace, v rámci kterých probíhá veškeré volání a zpracování služeb. Při vytváření relace předává klient serveru přihlašovací údaje a ten po jejich ověření označí klienta jako konkrétního uživatele s přidělenými právy. OPC UA nedefinuje, v jaké formě musí existovat uživatelé, pouze řeší způsob předávání přihlašovacích údajů. Každý kanál může hostovat nanejvýš jednu relaci, která se automaticky uzavírá po předem určené době nečinnosti [77, 116].

Adresní prostor serveru

Adresní prostor označuje množinu sdílených dat. Je to hierarchická struktura vzájemně odkazovaných objektů. Každý uzel této struktury má vlastní identifikátor *NodeId*, který je složen z názvu jmenného prostoru *Namespace* a unikátní části - číselné, textové nebo případně případně globálního identifikátoru. Jmenný prostor je jedinečný identifikátor (URI) označující, v jakém kontextu je uzel specifikován. Je-li na jiném serveru použit stejný identifikátor, včetně stejného jmenného prostoru, jedná se o totožný uzel. Mezi reference patří například *hasComponent*, která vnitřní uzel označuje jako komponentu zdrojového uzlu, *hasType*, která přiřazuje každému uzlu jeho typ či *hasVariable*, přiřazující cílový uzel jako proměnnou zdrojového uzlu a další.

OPC UA dále definuje vlastní jmenný prostor obsahující předdefinované uzly důležité pro provoz vlastního OPC UA serveru. Jedná se zejména o metadata, která specifikují syntaxi a sémantiku adresního prostoru samotného. Uzly v adresovém prostoru jsou organizovány do několika základních tříd:

- *Objekt* slouží k organizování svých komponent a proměnných. Konstrukce objektu by měla odpovídat jeho *typu*.
- *Proměnná* reprezentuje hodnotu. Jedná se zcela jistě o nejvyužívanější třídu, dělicí se na dva typy - *data variables* (nesoucí samotnou hodnotu) a *properties* (nesoucí přidané vlastnosti objektů, hodnot a dalších uzlů). Například uzel popisující snímač hmotnosti může obsahovat hodnotu hmotnosti (data variable) a také měřící rozsah (property). Hodnoty proměnných jsou komplexní struktury, které obsahují data, datový typ, čas serveru, čas zdroje a stavový kód a mohou reprezentovat jak skaláry, tak i pole s více dimenzemi, řetězce a složitější objekty. Konstrukce proměnné by měla odpovídat jejímu *typu*.
- *Metoda* reprezentuje odkaz, jehož aktivací dojde k volání příslušné metody na serveru se zadanými parametry. Návratová hodnota volání se po zpracování odesílá zpět klientovi.

- *Pohled* určuje, které z uzlů má klient možnost zobrazit. Je možné vytvořit více pohledů, jimiž lze různě filtrovat data zobrazená klientovi.
- *Typ referencí* se používá k vyjádření vztahů mezi uzly. Každý uzel musí být připojen alespoň jednou referencí k jinému, již připojenému uzlu adresního prostoru. V opačném případě uzel do adresního prostoru nenáleží.
- *Typ dat* definuje u proměnných datový typ uložené hodnoty.

Každý uzel v adresním prostoru serveru je popsán množinou atributů, přičemž každá třída uzlů má tuto množinu předepsánu jinak. Existují však také společné atributy pro všechny třídy, mezi které patří *NodeId*, *prohledávané jméno* (BrowseName), *zobrazované jméno* (DisplayName) a *hodnota nebo třída uzlu* (NodeClass). S využitím prohledávaného jména lze hledat uzly z výchozího uzlu pomocí referencí. Nemusí být unikátní - může vyjadřovat vlastnost a umístění uzlu. Další atributy se pak liší podle konkrétní třídy uzlu. Proměnná *hmotnost* zmíněného senzoru může mít následující atributy [116]:

- *Hodnota* - definující vlastní číselnou hodnotu proměnné.
- *Datový typ* - definující datový typ číselné hodnoty.
- *Úroveň přístupu* - určující, zda hodnotu lze z OPC UA klienta pouze číst, nebo také modifikovat.
- *Uživatelská úroveň přístupu* - určující uživatelská práva k přístupu k atributu hodnoty.

Služby poskytované v OPC UA

Komunikace klienta se serverem je realizována voláním a zpracováváním služeb, které umožňují ovládat jednotlivé funkce serveru OPC UA. Dotazy klienta i odpovědi serveru mají společné hlavičky, přičemž klient má například u dotazů možnost nastavit požadované informace, které má server vrátit. V odpovědi server nastavuje stavový kód vykonání požadavku oznamující, zda bylo vykonání služby úspěšné. Stavové kódy se dělí na *dobré* (Good), značící správně provedenou službu, *nejisté* (Uncertain) a *špatné* (Bad), které znamenají selhání během vykonávání služby. Chyby mohou vznikat např. při špatně nastavených přístupových právech, špatně provedené akci, nevhodném stavu serveru, apod. Nejistý výsledek může nastat, pokud alespoň jedna z částečných akcí skončí chybou. Mezi základní OPC UA služby se řadí:

- *Průzkumné služby* - jedná se o sadu služeb, umožňujících zjišťovat údaje o samotném serveru a možnostech jeho připojení. Připojit se lze pouze k přípojnému bodu (endpoint), který je otevřen pro komunikaci. Tyto body mohou (ale nemusí)

být předem známy. Právě pomocí sady průzkumných služeb je možné najít seznam přípojných bodů. Jednou z možností získání informací o serveru je připojení k veřejnému přípojnému bodu tohoto serveru, který poskytuje průzkumné služby. Klient následně může požádat o seznam dalších přípojných bodů, ke kterým se následně bude připojovat. Další možností je spuštění samostatného serveru s veřejně známou adresou v komunikační síti. Tento server následně agreguje adresy přípojných bodů všech ostatních serverů. Přípojný bod pro průzkumné služby bývá nezabezpečený. Spolu se seznamem ostatních přípojných bodů pak klient získává informace o požadované úrovni zabezpečení.

- Zabezpečený kanál - sada služeb umožňuje vytvoření a ukončení zabezpečeného kanálu. Při otevírání kanálu klient nastavuje používané zabezpečení, server při každém volání vrací unikátní bezpečnostní známku.
- Relace - sada služeb umožňující vytvoření a zrušení relace, aktivace existující relace, či zrušení všech probíhajících akcí. Klient může zrušit všechny probíhající služby, které zavolal v aktuální relaci, přičemž všechny odpovědi budou obsahovat chybový kód označující přerušeni akce. Tato funkce může být využita k uvolnění zdrojů, či přerušeni dlouho trvajících akcí.
- Správa uzlů - sada služeb, která umožňuje klientovi vytvářet a mazat uzly a upravovat reference mezi nimi. Klient tak může na serveru vytvářet vlastní objekty.
- Procházení - pomocí této služby lze procházet a načítat adresní prostor nebo jen jeho část (pohled). Lze nastavit, ve kterém směru budou které reference prohledávány, a která data má služba vrátit.
- Pohled (View) - sada služeb umožňujících postupné procházení adresního prostoru, pro hledání uzlů v rámci adresního prostoru a pro efektivní přístup k určitým uzlům (klient může označit uzel, který hodlá často využívat, server k němu následně optimalizuje přístup).
- Dotazování - pomocí služeb dotazování může klient posílat dotazy na specifická data. Dotazy mohou být konstruovány od jednoduchých, až po relativně komplexní. Lze nastavit jak požadované parametry uzlu, tak i složité podmínky. Při příliš velkém počtu výsledků se vrací pouze část, další data se pak získávají voláním jiné služby, které se předává bod, od kterého má seznam pokračovat.
- Atributy - tyto služby umožňují číst a zapisovat do atributů uzlu a číst historické hodnoty.
- Metody - služba, která volá metodu na serveru, předá jí argumenty a po jejím dokončení vrací klientovi návratovou hodnotu.
- Odběr (Subscription) - Odběr definuje frontu, do které jsou přidávána data. Po zavolání publikující služby se obsah fronty odešle klientovi. Mezi publikováním musí uběhnout předem určený interval. Po vícenásobném volání publikující služby se služba znovu vykoná až po uplynutí požadovaného intervalu.

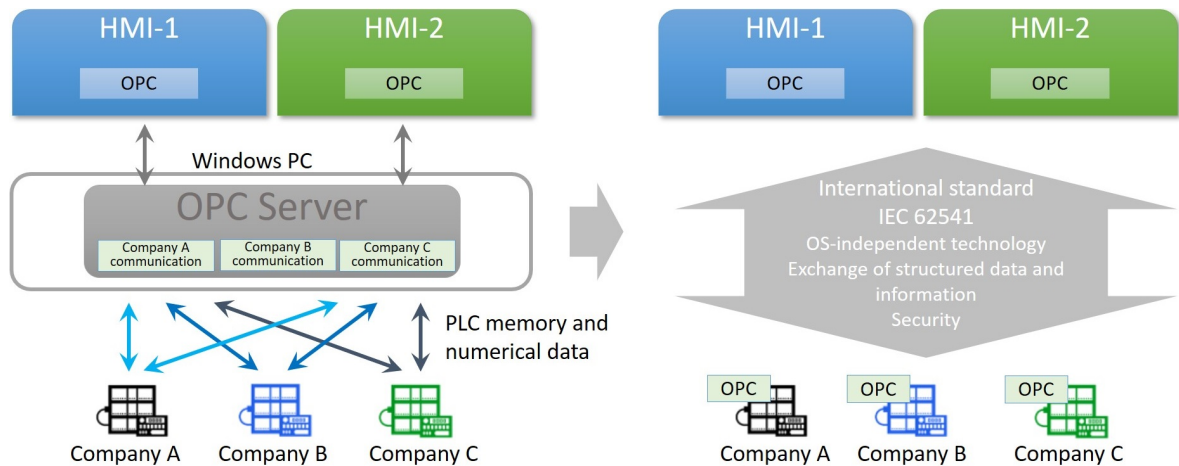
Zabezpečení komunikace OPC UA

Komunikaci OPC UA je nutné zabezpečit. Dostupné způsoby zabezpečení OPC UA lze dělit na *vnější* a *vnitřní*. Vnější zabezpečení znamená např. šifrování zpráv, jejich podepisování, či využití zabezpečeného kanálu. Vnitřní zabezpečení je realizováno zavíráním přebytečných či neužitečných spojení, či záznamem aktivity na serveru (auditováním). Každá aplikace, tedy účastník OPC UA komunikace, musí mít vlastní aplikační certifikát, který jednoznačně identifikuje aplikaci a stroj, na kterém tato komunikace běží. OPC UA pak definuje 4 úrovně zabezpečení [116]:

1. *Úroveň 1 - bez autentizace* - klient i server umožňují komunikovat s kýmkoli, tzn. že všechny platné klientské i serverové certifikáty jsou automaticky považovány za důvěryhodné, přestože nejsou zařazeny do seznamu důvěryhodných certifikátů. Příjemce nemá žádný způsob, jakým by ověřil zda poskytovatel je legitimní držitel certifikátu. Tato úroveň nevyžaduje žádné nastavení na straně klienta ani serveru.
2. *Úroveň 2 - serverová autentizace* - server umožní připojení libovolnému klientovi, přičemž ověření klienta se provádí pomocí ověřovacích údajů (jméno, heslo) zasláných na server po otevření zabezpečeného kanálu. Klient ověřuje certifikát serveru - serverový veřejný klíč musí být explicitně uložen v seznamu důvěryhodných certifikátů (konfigurace na straně klienta), nebo musí certifikát být vydán důvěryhodnou *certifikační autoritou* (CA).
3. *Úroveň 3 - klientská autentizace* - server umožní připojení pouze důvěryhodným klientům, ale klient se může připojit k libovolnému serveru. Server tedy poskytne data pouze, pokud důvěřuje klientskému certifikátu (konfigurace na straně serveru) uloženému v seznamu důvěryhodných certifikátů nebo vydanému důvěryhodnou CA.
4. *Úroveň 4 - oboustranná autentizace* - klient i server umožňují připojení pouze důvěryhodným partnerům. Jedná se o nejvyšší úroveň zabezpečení, která ale vyžaduje nastavení na obou stranách. Tento přístup je doporučen organizací OPC Foundation jako implicitní pro všechny klienty i servery.

Výhody a nevýhody OPC UA

Díky tomu, že OPC UA na rozdíl od svého předchůdce pro svůj provoz nepotřebuje DCOM knihovny společnosti Microsoft, lze jej provozovat i na jiných operačních systémech. Je tedy mnohem univerzálnější a v současné době se rozšířil jako průmyslový standard. Také proto je dnes již implementován v mnoha zařízeních různých výrobců, což značně usnadňuje jeho využívání. Mezi další výhody patří spolehlivá a robustní komunikace a plná integrace funkcí pro šifrování, zabezpečení a autorizaci dat. Na druhou stranu jsou zde omezení dané architekturou klient-server, především to, že při větším počtu požadavků klientů nemusí server stíhat odpovídat. Další nevýhodou pak



Obrázek 4.18: Srovnání situace před (vlevo) a po (vpravo) rozšíření OPC UA [90]

představuje poměrně komplikovaná implementace protokolu do nových zařízení. Starší verze protokolu navíc nejsou s OPC UA kompatibilní.

4.6.3 Srovnání MQTT a OPC UA

Bez uvedení konkrétní situace je srovnání MQTT a OPC UA protokolů obtížné, neboť jak MQTT, tak architektura OPC UA nabízí několik možností realizace. Srovnání tedy bude vycházet především z rozdílných paradigmat komunikace - publisher-subscriber v MQTT a client-server v OPC UA [16, 67, 76].

- *Škálovatelnost* - Architektura publisher-subscriber v MQTT dovoluje připojení velkého počtu příjemců bez ovlivnění události na straně odesilatele. Pro OPC UA je naproti tomu počet klientů, kteří mohou ve stejnou chvíli požadovat odpověď od serveru, značně omezen.
- *Náročnost integrace* - MQTT pracuje v jediném jednotném jmenném prostoru, jehož středobodem je broker. Jednotlivá zařízení a aplikace spolu nekomunikují přímo. To usnadňuje implementaci a integraci nových zařízení do již existujících sítí. MQTT je specifikován na 68 stranách (MQTT Sparkplug [67]). OPC UA naopak vyžaduje přímé propojení zařízení a aplikací. Aplikace jsou tedy přímo navázány na zařízeních, což činí návrh a integraci nových zařízení komplexnějším. Specifikace OPC UA se skládá z více, než 14 částí a čítá cca 1200 stran, je tedy náročnější na implementaci.
- *Rychlost odezvy* - Architektura MQTT je založena na efektivním přenosu asynchronních událostí, např. reakce na výjimky či alarmy je možná v reálném čase. V případě client-server architektury je schopnost komunikovat v reálném čase daná nastavenou frekvencí dotazování. Pokud je frekvence příliš vysoká, dochází k problémům.

- *Možnosti komunikace* - V MQTT může každé zařízení vystupovat jako *publisher*, i jako *subscriber*. To znamená, že každé zařízení může iniciovat komunikaci. Komunikaci OPC UA iniciuje pouze klient připojením na server. Proto ji označujeme za jednosměrnou.
- *Bezpečnost komunikace* - MQTT nespécifikuje žádné možnosti zabezpečení. Místo toho je nutné přenos dat realizovat prostřednictvím TLS/SSL zabezpečeného kanálu, využívat certifikáty pro ověření komunikačního partnera a vrstvu MQTT komunikace realizovat až nad těmito prostředky. Bezpečnostní výhodou MQTT je, že komunikaci vždy iniciuje zařízení, které se připojuje k brokerovi, není tedy nutné na zařízení otevírat port pro příchozí komunikaci. OPC UA specifikuje vlastní protokol pro zabezpečení s několika profily zabezpečení s různými úrovněmi ochrany. Nevýhodou oproti MQTT je nutnost otevřít TCP port pro příchozí komunikaci na straně serveru.

4.7 Shrnutí kapitoly

V průběhu této kapitoly byly představeny základní koncepty pro praktickou realizaci myšlenek Průmyslu 4.0. Tím nejdůležitějším je *Industry 4.0 komponenta*, která obsahuje *Asset Administration Shell* (AAS). Byly představeny tři klíčové varianty AAS: *pasivní*, *reaktivní* a *aktivní*. Zatímco pasivní varianta je vlastně digitální popis - model entity, který může poskytovat její výrobce, a který obsahuje všechna relevantní data entity, další dvě varianty jdou dále a umožňují svá data vyměňovat. Reaktivní AAS dovoluje jiným entitám, aby přistupovaly k datům modelu. Tato data, jakmile jsou udržována digitálně v rámci entity (Industry 4.0 komponenty), již nemusí být pouze statická, ale mohou reflektovat aktuální stav komponenty (vzniká tak vlastně digitální stín). Aktivní AAS pak implementuje určitou míru autonomie, a může s využitím vyjednávacích sémantických protokolů realizovat plnění svých specifických cílů. Komunikace mezi aktivními I4.0 komponentami probíhá s využitím tzv. jazyka I4.0 - jednoduchého standardu popisujícího základní sémantiku zpráv. V prostředí I4.0 komponent se často využívá komunikačních protokolů OPC-UA či MQTT, přičemž každý z nich má svá specifika a jiné možnosti využití.

Aktivní AAS, zejména sémantické interakční protokoly pro distribuované plánování výrobních operací, jsou zajímavou problematikou, proto v rámci následujících kapitol bude tato problematika rozebrána do větší hloubky.

Kapitola 5

Plánování výroby

V kontextu Průmyslu 4.0, RAMI modelu a administrativní obálky nelze vynechat základní myšlenky plánování výroby v podnicích a jaká je vůbec motivace k takovému plánování. Svůj význam přinese i přehled historických a současných plánovacích metod i algoritmů. Tyto informace budou dále užitečné při srovnání centralizovaného a distribuovaného přístupu k plánování, výhod a nevýhod obou konceptů.

5.1 Úrovně plánování v podniku

Výrobní podniky od nepaměti hledaly způsoby, jak zlepšit svou konkurenceschopnost. Počínaje prvními řemeslníky, závislými na úspěchu prodeje svých produktů, bylo v době manufaktur i po vzniku továren klíčové zvládnout výrobu produktů v požadované kvalitě. Až do konce první poloviny 20. století to stačilo. S rostoucí konkurencí však byly společnosti nuceny hledat nové způsoby, jak zlepšit a zefektivnit své výrobní operace, a to nejen uvnitř firmy, jak bylo do té doby zvykem, ale také za svými zdmi - pozorováním konkurence, aplikováním nových poznatků a metod či sledováním a důslednou analýzou chování zákazníků.

Podniky dnes musí být konkurenceschopné v mnoha oblastech. Musí produkovat své výrobky v odpovídající kvalitě, dodržovat termíny dodávek, vyrábět efektivně a vzhledem k měnícímu se prostředí, také s vysokou flexibilitou. Proto je pro ně nezbytné, aby odpovídajícím způsobem plánovaly a řídily své výrobní i nevýrobní činnosti. Úkoly plánování a řízení se stávají stále složitějšími - požaduje se *zkracování dodacích lhůt, zkracování životních cyklů výrobků, efektivní využívání zdrojů a úzkých míst* ve výrobě, a mnoho dalšího.

Plánování můžeme definovat jako dynamický proces, při kterém se rozhoduje o aktivitách, které je potřebné uskutečnit v blízké či vzdálenější budoucnosti. Efektivní plánování vede k dosažení optimálního fungování výrobních celků, eliminuje chyby, které by mohly vést k plýtvání finančních prostředků nebo času. Způsoby plánování a řízení výrobních systémů se liší podle charakteru konkrétní výroby a jsou pro každý

podnik specifické. Důležitou podmínkou pro možnost efektivního plánování je získávání včasných a korektních dat z výrobních informačních systémů.

Problematika plánování a řízení výroby zahrnuje komplexní spektrum dokonale propojených činností a vazeb, které vedou k úspěchu v současném konkurenčním prostředí. Plánování a řízení výroby můžeme rozdělit do tří úrovní [63]:

- **Strategické** plánování a řízení
- **Střednědobé** plánování a řízení
- **Operativní** plánování a řízení

Strategická úroveň plánování

Jedná se o plánování s nejdelším časovým horizontem. Záleží vždy na oboru a typu podnikání, ale obvykle se na této úrovni plánuje na období tří, pěti i deseti let. Na této úrovni plánování a řízení se zapojuje především vrcholový management firmy, který rozhoduje na základě strategické analýzy okolního prostředí, vnitřních zdrojů a znalostí, a také konkurence. Strategické plánování je cyklický proces, který nikdy nekončí.

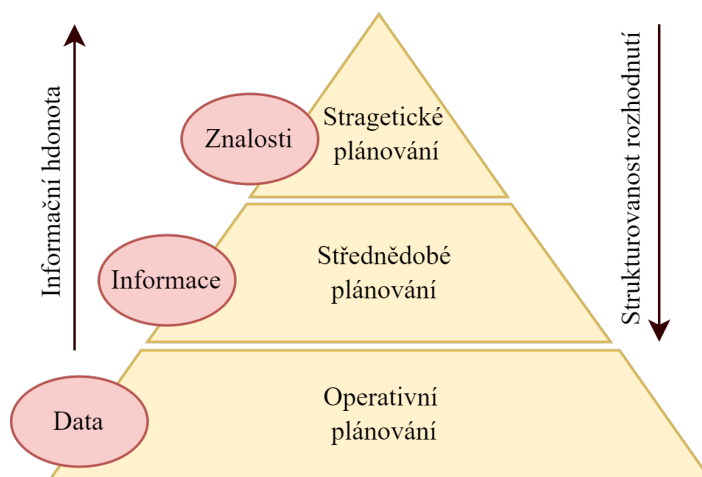
Výstupem uvedených analýz je dokument *Mission statement*, zachycující vizi společnosti. Podává informace o základním směru, kterým se společnost chce rozvíjet. Dále pak stanovuje segment trhu, na kterém chce působit a vymezuje vlastní produkty nebo služby a následně zvolí jejich způsob řízení. Při rozhodování o budoucí strategii podniku se využívají softwarové nástroje - *manažerské simulátory* poskytující možnost simulace scénářů a analýz typu *co se stane, když*. Strategická úroveň plánování a řízení výroby má úzké vazby i na další strategické oblasti v podniku, například obchod, ekonomiku a marketing.

Střednědobá úroveň plánování

Střednědobá úroveň plánování přichází na řadu po stanovení strategického plánu výrobního podniku. Výchozím bodem je dokument - *Business plan*. Zatímco *Mission statement* je spíše vizí, *Business plán* je již konkrétní plánovací dokument obsahující:

- **plánovaný počet, typ a kvalitu výrobků**, které chce podnik vyrobit,
- **segment trhu a cílová skupina zákazníků**, kterým budou výrobky určeny a
- **zdroje a kapacity**, které budou k realizaci výrobků zapotřebí.

Business plán by měl mít takovou strukturu, aby mohl být podkladem i pro následné *ekonomické vyhodnocování* (**controlling**), a tím i přímo odrážel plánované hospodářské výsledky podniku pro konkrétní období.



Obrázek 5.1: Hierarchie plánování a řízení výrobních operací

Informační podpora střednědobého plánování a řízení výroby zahrnuje celou řadu balíků typu *Material Requirements Planning* (MRP), *Advanced Planning and Scheduling* (APS) a *Supply Chain Management* (SCM). Jejich primárním úkolem je shromažďovat a uchovávat data související s business plánem, zprostředkovávat souhrnné a konzistentní informace o výrobě a zároveň optimalizovat podnikové výrobní zdroje. Popsané nástroje a jejich metodiky mohou fungovat buď jako součást podnikového informačního systému, nebo samostatně s vyšší či nižší mírou integrace s ostatními informačními systémy.

Operativní úroveň plánování

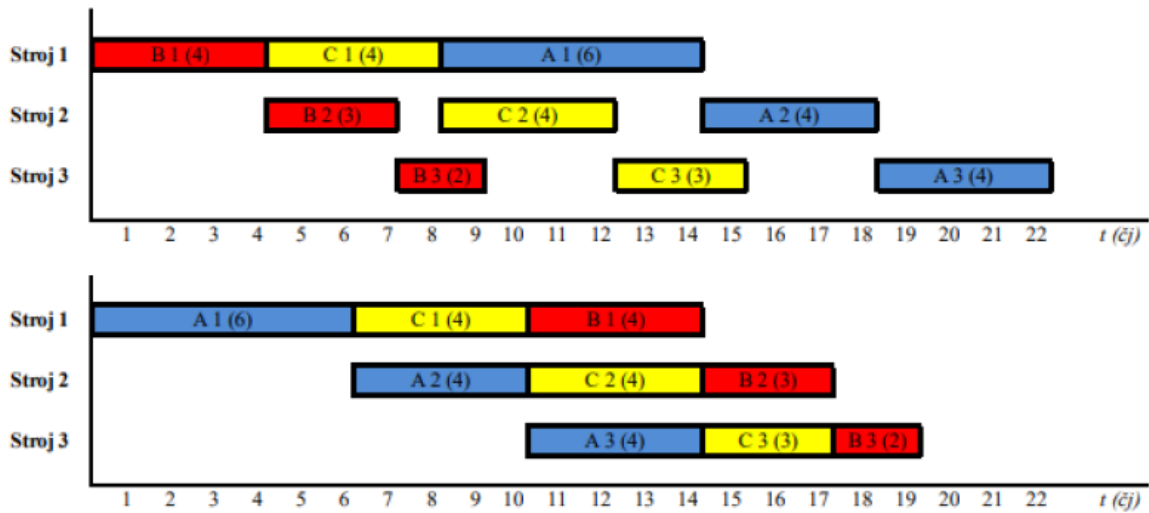
Operativní plánování je v hierarchii položeno nejnižší, ale rozhodně není nejméně důležité. Řeší totiž každodenní provozní problematiku výroby, která je vyjádřena pomocí tzv. *operativního plánu výroby*. Ten většinou zahrnuje *sledování plánovaných dodávek*, *vyhodnocování jednotlivých výrobních operací*, *operativní evidenci výroby*, *regulování absence a přesčasů dělníků*, *řešení poruch výrobních zařízení* apod.

Operativní plánování přináší maximalizaci produktivity zaměstnanců, lepší efektivitu využívání výrobních zařízení a zdrojů podniku, a také slouží jako nástroj pro stanovení a kontrolu cílů. Pro rychlé a správné rozhodování na této úrovni je samozřejmě potřeba, aby byly stále k dispozici aktuální a jednoznačně interpretovatelné informace z výroby i dalších oddělení.

Mezi nevýhody plánování řadíme vysoké vstupní finanční a časové náklady a omezování iniciativy zaměstnanců. V neposlední řadě může proces plánování vést k nadměrné byrokracii.

Rozsah domény plánování a rozvrhování výroby není dobré podceňovat, neboť celá tato problematika je typická protichůdnými požadavky, které jsou kladeny na výrobu. Na jedné straně panuje snaha o maximální využití výrobních kapacit při co nejkratších prostojích a minimálních zásobách. Na straně druhé stojí realita omezených

materiálových i personálních zdrojů. Firmy proto věnují značné úsilí zeštíhlování svých výrobních procesů a tvorbě detailních rozvrhů výrobních operací. Důležitost plánování ve výrobním procesu demonstruje obrázek 5.2.



Obrázek 5.2: Ganttův diagram pořadí zadaných operací do výroby a jejich vliv na využití zdrojů [53]

Operativní údaje o výrobě se většinou zadávají do modulů a plánovacích tabulí systémů MRP či APS. Často se tak děje pomocí speciálních aplikací, které bývají dodatečně vytvořeny tak, aby vyhovovaly speciálním potřebám konkrétní výroby či zákazníka. Jedná se například o výrobní linky přímo propojené s informačním systémem, podporu automatizovaného sběru dat pomocí snímání čárových kódů atd. *Snahou je minimalizovat možnost vstupu chybného údaje do systému, a přitom nezatěžovat pracovníky ve výrobě komplikovaným softwarem.*

Jednotlivé úrovně plánování a řízení vytvářejí pyramidu. Pro *strategické plánování* je charakteristické *volnější, spíše intuitivní rozhodování* na základě znalostí. *Střednědobé plánování* je již *přísně strukturované založené na logickém rozhodování*, přijatém systému metodik a informační podpoře podnikových systémů. V *operativním plánování* se využívají na první pohled logická a strukturovaná řešení, vycházející z *momentální situace*.

Informační systémy zde hrají stejně důležitou roli jako ve střednědobém plánování. Velmi důležitý je také zápis přesných a aktuálních dat do těchto systémů. *Data* se pak zpětně transformují na *informace* a *znalosti*, které se následně využijí na vyšších úrovních řízení. Na všech úrovních je však potřeba mít vždy na paměti, že není možné transformovat extrapolované cíle z minulosti do budoucích očekávání.

5.2 Potřeby výrobních firem v oblasti plánování

Různé strategie, možnosti firem a zákaznické potřeby znamenají jiné požadavky na výrobu jako takovou. Při rozlišování typu výrob je důležitá spojitost výrobního procesu a také místo, ve kterém dochází k rozpojení objednávky, tedy místo, kde se výroba stává závislá na poptávce konkrétního zákazníka [135].

Z hlediska spojitosti procesu je, jak již bylo zmíněno v kapitole 1.4, možné dělit výrobu na spojitou a nespojitou, která se dále dělí na kusovou (diskrétní) a dávkovou. Další možností je dělení dle bodu rozpojení objednávky, které určuje, kdy je vyráběno na základě predikce a kdy podle přání konkrétního zákazníka. Rozlišují se tyto čtyři možnosti [135, 103, 59]:

- **Výroba na sklad** (make-to-stock) – probíhá na základě *predikce* očekávaných objednávek zákazníků. K bodu rozpojení dochází na pozici *zásob hotových výrobků*. Zákazník pak nakupuje zboží, které je na skladu. V okamžiku prodeje zboží se nezávislá zákaznická poptávka stává závislou poptávkou pro výrobu.

Při tomto typu výroby je nutné se zaměřit na správu zásob hotových výrobků, přičemž nejdůležitějším aspektem řízení zásob se stává určení toho, kdy, kolik a kde budeme dané výrobky potřebovat. To je mnohdy složitý problém velkých korporací, vezmeme-li v úvahu, kolik poboček, distribučních center a skladů v různých zemích mají. Řízení takového dodavatelského řetězce potřebuje co nejpřesnější informace o stavu zásob a odhadu poptávky pro každou součást řetězce a každou položku.

Typickým případem výroby na sklad jsou např. konzervované potraviny, spotřební elektronika, knihy, koupelnová technika apod.

- **Montáž na zakázku** (assembly-to-order) – oproti výrobě na sklad vyžaduje montáž na zakázku komunikaci mezi zákazníkem a dodavatelem. Zákazník potřebuje být informován o možných konfiguracích výrobku a dodavatel musí být schopen potvrdit, že je možné jej vyrobit. Ověřováním, zda mohou být jednotlivé požadované části zkompletovány do funkčního výrobku se zabývá tzv. *configuration management*.

Základním úkolem v tomto typu výroby je odhad poptávky zákazníka (požadavků trhu) z hlediska požadavků na různé kombinace výrobků, které výroba může nabídnout. Design a funkčnost výrobků by proto měla být navrhována tak, aby umožňovala flexibilně kombinovat komponenty a moduly výrobku.

Příkladem montáže na zakázku a *configuration managementu* je montáž počítače dle požadavků zákazníka, kdy je na výběr z několika druhů procesorů, grafických karet, pevných disků, pamětí apod. Pokud by byly možné všechny kombinace těchto součástí a pro každou bychom uvažovali 4 možnosti, existoval bych celkový počet 256 kombinací. Je zřejmé, že je rozhodně snazší řídit a předpovídat poptávku 16 komponentů, než 256 hotových počítačů. V případě montáže na

zakázku se tedy nezávislá poptávka po výrobcích objevuje v bodě montáže dílů se závislou poptávkou. Podíváme-li se na dnešní výrobu osobních automobilů, uvidíme, že těch volitelných parametrů jsou desítky a kombinací pak miliony.

- **Výroba na zakázku** (production-to-order) – až dosud jsme se bavili o případech, kdy firmy dokázaly částečně predikovat, jaké výrobky si zákazníci budou chtít koupit, avšak neznaly dopředu požadované množství. V výrobě na zakázku je situace ještě horší, neboť výrobci zde vůbec netuší, o jaké produkty zákazníci budou mít zájem. Do hry navíc vstupuje ještě další komplikace, kterou je problematika samotné konstrukce výrobků. Projektanti musí rozhodovat, jaké materiály použít, jakými výrobními fázemi výrobek projde, a jaké budou vůbec náklady na něj. Bod rozpojení objednávky se tedy přesouvá k surovinám nebo až k dodavatelům materiálu.

Příkladem výroby na zakázku jsou drahé dopravní prostředky, jako např. letadla, lokomotivy nebo investiční celky v podobě celých výrobních linek.

- **Inženýrské práce na zakázku** (engineer-to-order) – jedná se o specifický typ výroby, kdy objednávky zákazníka nevedou k předem definovaným produktům či komponentám, ale k představám (více či méně specifickým) o výsledném produktu. Jinými slovy vytváří unikátní výrobek splňující požadavky konkrétního zákazníka. Velkou roli zde hrají projektanti, jejichž úlohou je jednak získat zákaznickou představu a specifikovat požadavky (v rádech týdnů až měsíců), vymyslet a zadokumentovat technické řešení do posledního detailu tak, aby bylo možné produkt následně realizovat. Je proto důležité co nejlépe odhadnout, kolik kapacity projektantů návrh a výroba požadovaného výrobku spotřebuje. Roční produkce u tohoto typu výroby bývají v rádech jednotek, maximálně desítek produktů.

Lokace zásob	Dodavatelé	Suroviny	Rozpracovaná výroba	Hotové výrobky
Bod rozpojení objednávky	↑	↑	↑	↑
Model řízení výroby	Engineer to Order	Make to Order	Assembly to Order	Make to Stock

Obrázek 5.3: Bod rozpojení v různých typech výroby, dle [135]

5.3 Metody pro plánování výrobních kapacit

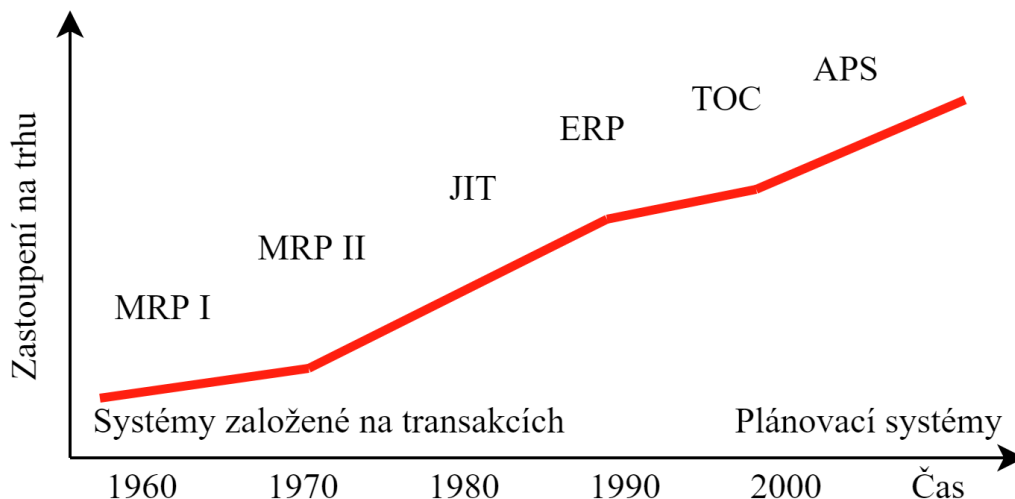
Jak již bylo řečeno, proces plánování a řízení výroby se rozvíjí od dob, kdy se začíná masivně prosazovat průmyslová výroba. Manuální rozvrhování bylo po nástupu počítačové éry a vzniku prvních plánovacích řešení vytlačováno, avšak zcela nezmizelo.

Své uplatnění nachází minimálně ve výrobních provozech, kde z pohledu uspořádání výroby použití pokročilejších plánovacích algoritmů nepřináší výraznou přidanou hodnotu (např. je-li více strojů využíváno postupně pro výrobu jediného typu produktu bez modifikací - *flow job*, viz dále). V takovém případě se místo rozvrhování dá mluvit spíše o jednoduchém monitorování a plánování průběhu výroby, často se také neuvažují úzká místa ve výrobním procesu. Řízení výroby probíhá s využitím prioritních pravidel a časy spouštění výroby jednotlivých dílů jsou určovány s ohledem na zkušenosti operátora výroby a dostupnosti materiálu. Nejčastěji je v takových případech využíváné pravidlo FIFO - první vstupující je také prvním vystupujícím ze systému. Při manuálním rozvrhování jsou priority určovány pouze na základě požadovaných termínů dokončení zakázky, tzn. že nejvyšší prioritu má zakázka, kterou je potřeba dokončit nejdříve (ať už v termínu nebo s co nejmenším zpožděním). Je zřejmé, že taková prioritizace může principiálně vést k lavinovému efektu, kdy kromě již zpožděných zakázek dochází postupně ke zpoždění dalších [135].

Do problematiky ručního plánování je možné zařadit také rozvrhování výroby s využitím výpočetních prostředků, jako jsou tabulkové kalkulátory či plánovací tabule. Rozsah použití těchto prostředků je však omezený pouze na dostatečně malé a jednoduché výrobní systémy. Pro složitější struktury není ruční plánování efektivně možné, neboť generování optimálního rozvrhu by vyžadovalo v takových systémech neúměrně dlouhý čas.

Rozvoj podnikové informatiky zpřístupnil plnou integraci pokročilých metod plánování a řízení do výrobních podniků. Jádrem tohoto řízení jsou ERP systémy, které umožňují koordinaci všech disponibilních podnikových zdrojů a obsahují širokou škálu modulů (jako jsou např. ekonomické či logistické funkce), mezi nimi i moduly pro rozvrhování a plánování. Ty se od sebe liší dle aplikovaného principu řízení logistického toku - jsou to *princip tlaku* a *princip tahu*. Tlačný princip říká, že termíny pro objednávání materiálu jsou stanoveny na základě struktury výrobku dopředu. Výrobek je *tlačen* na základě daného plánu až k zákazníkovi. Naopak pro tažný princip je charakteristická výroba iniciovaná zákazníkem prostřednictvím objednávky a materiálové požadavky jsou taženy v podobě plnění objednávek směrem od zákazníka k dodavateli [52].

Metody pro rozvrhování materiálových zásob v minulosti vycházely z koncepce řízení výroby dle minimálních zásob. Tento způsob řízení byl rigidní ke změnám a docházelo ke zbytečnému hromadění zásob, a tedy i neúměrnému zvyšování vázaného kapitálu. Pozdější nasazení metody *MRP*, která již má vazbu na řetězce zásobování, skladování a dopravy znamenalo viditelné přínosy v oblasti redukce materiálových zásob. Výstupem této metody byl seznam zakázek a potřebného materiálu spolu s reportem dostupných zásob či iniciací případného naskladnění. *MRP* samotná nebyla zaměřena na vytváření efektivních výrobních plánů, neboť nedovolovala rozvrhování výroby do omezených výrobních kapacit. Docházelo tedy k vytváření nereálných výrobních plánů. Tento problém bylo nutné dořešit dodatečným rozvrhováním [103, 135]. Doplněním *MRP* o kapacitní plánování, nákupy a prodeje v podobě zavedení zpětnovazebních smyček z reálné výroby vznikla metoda *MRP II*. Nejednalo se však o



Obrázek 5.4: Historie plánovacích metod, dle [135]

koncepční, ale pouze technologickou změnu. Zohledněním kapacitních omezení výroby spolu se zvýšením kvality a agregace toku dat bylo možné předcházet nereálnému plánování typickému pro jejího předchůdce. Metody MRP a MRP II využívají tlačný princip řízení výrobního podniku. Jejich výhodou je možnost integrace do celopodnikového systému plánování zdrojů implementovaného v rámci ERP. Postupem času začaly firmy vnímat nedostatky těchto konceptů a začaly se prosazovat nové metody, které jsou schopny efektivně plánovat nejen uvnitř podniku, ale také v rámci celého dodavatelského řetězce.

Na obrázku 5.4 je naznačena historie od již zmiňovaných metod MRP a MRP II po metody založené na tažném principu, např. *Just in Time* (JIT). Tato filozofie vynáší na popředí samotné řízení výrobního procesu s přihlédnutím ke všem článkům dodavatelského řetězce. Mezi hlavní charakteristiky konceptu JIT patří důraz na nulové zásoby, nulové ztráty a maximální efektivitu (tedy *štíhlá výroba*). Jde především o opakovanou výrobu stejných či podobných produktů většinou s konstantním odbytem [131].

Theory of Constraints (TOC) je kombinací tažného a tlačného principu. Při plánování touto metodou se bere v úvahu úzké místo. Před tímto místem je použit tažný způsob plánování, za ním pak tlačný princip. Systémy pokročilého plánování *Advanced Production Scheduling* (APS) eliminují nevýhody uváděných metod MRP a umožňují plánovat s omezenými kapacitami a řídit výrobu s přihlédnutím na omezení úzkých míst ve výrobě. Umožňují ucelené plánování celého dodavatelského řetězce s podporou rozhodování v rámci *Supply Chain Management* (SCM), tj. celého dodavatelského řetězce až k zákazníkovi. Výstupem APS systémů je realistický a optimalizovaný výrobní plán, který reflektuje reálná omezení a okolnosti ve výrobě. Tyto systémy jsou integrovány do podnikových informačních systémů a zprostředkovávají jednotlivým modulům plánovací funkce provádějící optimální výpočty na základě dat převzatých z jejich databází. APS integrované do ERP představují základní informační technologie

pro řízení výroby v podniku [135].

Optimalizace výrobního procesu zahrnuje celou řadu nástrojů z oblasti operační analýzy, statistiky a managementu. Cílem této operace je nalezení takových řešení, která budou ve vysoké míře optimální z pohledu časové náročnosti, kvality výroby, nákladů a servisu. Prostředkem pro nalezení nejlepšího řešení v úlohách rozvrhování je optimalizace hodnoty účelové funkce vyjádřené vhodnou matematickou formulací.

Konvenční metody pro rozvrhování a plánování je možné dle různých specifik rozdělit obecně do několika skupin [79].

- **Metody operačního výzkumu** pracující s exaktními matematickými modely a algoritmy jsou účinným nástrojem pro reprezentaci mnoha plánovacích / rozvrhovacích úloh. V rámci operačního výzkumu se rozlišují dvě skupiny matematických modelů, a to *rozhodovací* (zahrnují kriteriální funkci a optimalizují) a *technologické* (nezahrnují kriteriální funkci).
- **Heuristicky orientované přístupy** se využívají v případech, kdy není znám všeobecný algoritmus pro získání řešení problému, případně kvůli jeho složitosti není možné sestavit matematický model. Tento přístup sestává z analýzy jednotlivých kroků, přičemž po každém kroku se hodnotí jeho přínos ke konečnému výsledku. Taktéž se iterativně určuje, jakým směrem je nutné postupovat k dosažení požadovaného řešení. V problémech rozvrhování výroby se heuristické metody využívají nejčastěji při výběru z množiny úloh na základě zvolených pravidel, případně podle zkušenosti pracovníků.
- **Úplné metody** poskytují zaručeně optimální řešení rozvrhovacích úloh.
- **Neúplné metody** umožňují projít pouze část stavového prostoru, přičemž poskytnuté řešení je pouze suboptimální. S výhodou jsou využívány zejména pro rozsáhlé úkoly, ve kterých by úplné metody selhávaly na problému kombinatorické exploze.
- **Nestandardní metody**, jako jsou neuronové sítě, fuzzy logika, genetické algoritmy či strojové učení jsou zdrojem nových možností řešení problémů v oblasti výroby.

5.4 Matematické modely plánovacích problémů

V posledních několika desetiletích bylo výzkumu deterministického plánování věnováno značné množství úsilí. V odborné literatuře nalezneme široké spektrum deterministických modelů s větší, či menší mírou abstrakce. Mnoho z nich je také implementováno v různých softwarových protuktech pro plánování výroby, nejčastěji v APS systémech. Rozvrhovací problém je v literatuře často obecně označen jako *RCPSP* (Resource-Constrained Project Scheduling Problem).

Klasifikace úloh rozvrhování

Široké spektrum různých typů rozvrhovacích úloh vyžaduje jednotnou definici způsobu jejich klasifikace. Pro popis problému rozvrhování se proto využívá *Grahamova notace*, která se skládá ze tří částí označených α, β, γ [82].

První ze symbolů α označuje typ cesty operací (úloh) mezi stroji, a tedy určuje druh modelu rozvrhování výrobního vybavení, které můžeme rozdělit do tří kategorií: rozvrhování na *jednom stroji*, využití *paralelních strojů* a *vícetupňové rozvrhování*.

- **Jeden stroj (1)** - nejjednodušší výrobní prostředí, kdy rozvrhování výroby probíhá pouze na jednom stroji. Tato úloha je speciálním nejjednodušším případem pro všechny následující modely.
- **Paralelní výroba na identických strojích (P_m)** - označuje výrobní prostředí, ve kterém každá úloha j vyžaduje jednu operaci, přičemž může být provedena na kterémkoli z m strojů, nebo z jejich definované podmnožiny.
- **Paralelní stroje s různými rychlostmi (Q_m)** - označuje případ, kdy každý z identických strojů i má jinou rychlost v_i . Doba p_{ij} , kterou úloha j stráví na stroji i je pak rovna p_j/v_i . V případě, kdy bude platit $v_i = 1$ pro všechna i a $p_{ij} = p_j$, model se stane identickým s předchozím popsáním.
- **Nezávislé paralelní stroje s různými rychlostmi (R_m)** - označuje další, obecnější model m paralelních strojů. Každý stroj i zpracovává každou úlohu j různou rychlostí v_{ij} . Doba, kterou úloha j stráví na stroji i je pak rovna p_j/v_{ij} . Ve speciálním případě, kdy rychlost stroje je nezávislá na zpracované úloze se model stane identickým s předchozím případem.
- **Flow shop (F_m)** (proudový problém) - označuje model m strojů v sériovém uspořádání. Každá úloha pak musí být zpracována na prvním stroji, následně na druhém stroji, atd. Během vykonávání mohou mezi rozvrhovanými stroji vzniknout fronty, pro které se zpravidla uplatňuje princip FIFO (First-in-first-out). V případě využití FIFO principu pro zpracování front se tento problém nazývá *permutačním*.
- **Flexible flow shop (FF_c)** - označuje spojení proudového problému a paralelního zpracování. Místo m strojů v sérii existuje c sériově modelovaných stupňů, přičemž každý je složen z více identických strojů.
- **Job shop (J_m)** (sekvenční problém) - označuje model, ve kterém každá z úloh má individuálně definovanou cestu mezi výrobními stroji. Každá zakázka má svůj technologický postup, přičemž neplatí, že by všechny zakázky musely využít všechny stroje. Model se dále dělí na případy, kdy každá zakázka přejde přes každý definovaný stroj pouze jednou, a úlohy, kde nastává požadavek vícenásobného zpracování zakázky strojem.

- **Open shop** (O_m) (otevřený problém) - specifický model rozvrhování ve výrobě, ve kterém pořadí operací v zakázkách není dopředu definováno. Každá zakázka musí být zpracována na každém z m strojů, avšak některé z časů zpracování mohou být nulové. Model nedefinuje žádná omezení týkající se směrování zakázky výrobou, každá zakázka tedy může být plánována jinou cestou. Pro tento model existuje množství dalších variant [82], které jsou dále charakterizovány symbolem β .

Tabulka 5.1 shrnuje nejčastěji používané modely.

Typ rozvrhovacího modelu	Počet úloh v rámci zakázky	Pořadí úloh v rámci zakázky
Flow shop	Stejný pro všechny zakázky	Pevně dané, stejné pro všechny zakázky
Job shop	Různý pro jednotlivé zakázky	Pevně dané, různé pro jednotlivé zakázky
Open shop	Stejný pro všech zakázky	Libovolné

Tabulka 5.1: Přehled základních druhů modelů pro výrobní systémy

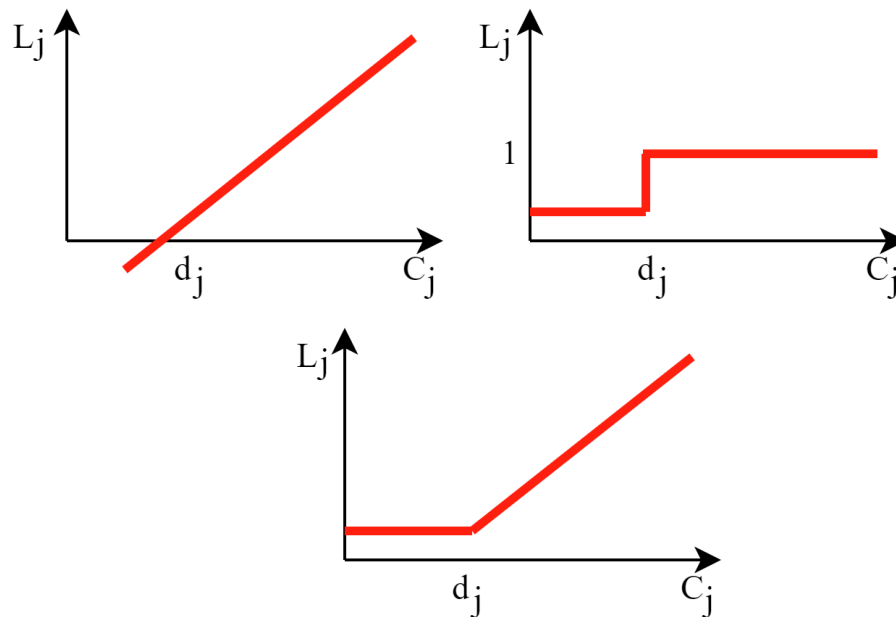
Symbol β v charakteristice modelu určuje další detaily zpracování, nejčastěji definuje omezení. Tento symbol může obsahovat jednu, či několik hodnot z následujících:

- **Preemptions** ($prmp$) (právo na přerušení) - reprezentuje stav, kdy není nutné, aby úloha byla na určitém stroji zpracována bez přerušení. Plánovač tedy může zastavit zpracování běžící úlohy s nižší prioritou, tu uložit tak, aby její výstup nebyl znehodnocena a využít zdroje a prostředky pro vykonání úlohy s vyšší prioritou.
- **Precedence constraints** ($prec$) (přednostní omezení) - jedno z nejčastěji se vyskytujících omezení v rozvrhovacích problémech. Reprezentuje podmínku, aby jedna, nebo více úloh byly řádně ukončené před začátkem zpracování konkrétní úlohy. Každá z úloh má tedy své předchůdce a nástupce. Pokud mají všechny úlohy nejvíce jednoho předchůdce a nejvíce jednoho nástupce, hovoříme o *řetězu* omezení.
- **Sequence dependent setup times** (s_{ijk}) (variabilní doby přestavování výroby) - s_{ijk} popisuje čas potřebný pro změnu výroby, pokud se na stroji i zpracovávala úloha j a dále se bude zpracovávat úloha k . Pokud doba změny je pro všechny stroje stejná, index i se vynechává.
- **Machine eligibility restrictions** (M_j) (omezení způsobilosti strojů) - při paralelním uspořádání strojů definuje takovou skupinu strojů, které jediné jsou schopny zpracovat danou úlohy.

- **Blocking** (*block*) (blokování) - výkonnost rozvrhování ve výrobě, nejčastěji v modelu *flow shop* je ovlivněna velikostí zásobníku mezi dvěma za sebou jeoucími stroji. Pokud se zásobník naplní (případně pokud vůbec neexistuje), předcházející stroj nemůže uvolnit dokončenou úlohu následujícímu, a tedy poskytnout své prostředky další úloze.
- **No-wait** (*nwt*) (bez čekání) - reprezentuje požadavek vykonání úloh bez prostojů, tedy úloha v žádném případě nesmí po dokončení na jednom stroji čekat na uvolnění druhého stroje.

Symbol γ obsahuje cílové hodnoty a účelové funkce, které mají být minimalizovány, tj. specifikuje optimalizační kritérium. Při sestavování optimálního rozvrhu jsou účelové funkce, pomocí nichž je definována logika přechodu úloh výrobním prostředím s cílem minimalizace hodnoty vybraného kritéria. Kritérium je vždy založeno na minimalizaci funkcí popisujících čas dokončení úloh.

Označme čas dokončení úlohy j na stroji i jako C_{ij} . Čas, po který úloha j existuje v systému, se značí jako C_j (jedná se o čas dokončení úlohy na posledním stroji, na kterém je vyžadováno zpracování).



Obrázek 5.5: Základní tvary nákladových funkcí, dle [137]

Vezmeme-li v úvahu čas dokončení úlohy d_j , pak *opožděnost úlohy* je definováno jako $L_j = C_j - d_j$ a jeho hodnota je kladná, pokud je úloha j dokončena pozdě, resp. záporná, pokud je úloha dokončena včas. *Nedochvilnost úlohy* je pak definováno jako $T_j = \max(C_j - d_j, 0) = \max(L_j, 0)$ ¹. *Jednotkový postih* je definován jako

¹Je zřejmé, že rozdíl mezi *opožděností* a *nedochvilností* je ten, že *nedochvilnost* nikdy nenabývá záporných hodnot.

$$U_j = \begin{cases} 1 & \text{pokud } C_j > d_j \\ 0 & \text{jinak} \end{cases}$$

Opožděnost, nedochvilnost a jednotkový postih (viz obr.) jsou tři základní typy chování používaných v nákladových funkcích. V následujícím přehledu je uvedeno několik kritériálních funkcí, jejichž minimalizací dostáváme optimální řešení plánovacích problémů:

- **Makespan** (C_{max}) (výrobní rozsah) - je definován jako $\max(C_1, \dots, C_n)$ a je ekvivalentní času ukončení poslední úlohy rozvrhované zakázky. Nejčastěji je při tvorbě optimálního rozvrhu předmětem minimalizace čas ukončení poslední z úloh.
- **Total weighted completion time** ($\sum w_j C_j$) - (celkový vážený čas ukončení) - dává informace o čase ukončení n úloh s ohledem na váhy, které definují náklady na skladování a tvorbu zásob potřebných pro realizaci rozvrhu.
- **Discounted total weighted completion time** ($\sum w_j(1 - e^{-rC_j})$) (diskontovaný celkový vážený čas ukončení) - reprezentuje obecnější nákladovou funkci oproti předchozímu případu, ve které jsou náklady diskontovány v rozsahu $0 < r < 1$ a vztaženy k jednotce času. To znamená, že pokud úloha j nebude dokončena v čase t , vzniknou v průběhu periody $[t, t + dt]$ dodatečné náklady $w_j r e^{-rt} dt$. Pokud bude úloha j dokončena v čase t , celkové náklady vzniklé v periodě $[0, t]$ jsou $w_j(1 - e^{-rt})$. Hodnota r je obvykle blízká 0.

Pro objasnění výše uvedených definic je možné uvést několik příkladů $\alpha|\beta|\gamma$ notace pro popis plánovacích modelů:

- $FF_c|r_j|\sum w_j C_j$ popisuje flexibilní proudový problém o c úlohách. Jednotlivé úlohy mají specifikován čas uvolnění do výroby a nejpozdější čas dokončení. Kritériální funkce pak odpovídá celkovému váženému času ukončení.
- $J_m|r_j, s_{ijk}, rcrcl|\sum w_j C_j$ popisuje sekvenční problém o m úlohách, které mají specifikován čas uvolnění do výroby, přičemž časy změny výroby mezi úlohami j a k nejsou nulové a jsou pro každý stroj i různé. Existuje možnost opakovaného zpracování jedné úlohy ve stroji. Cílem je minimalizace celkového váženého času ukončení.
- $1|s_{jk}|C_{max}$ popisuje systém s jedním strojem, ve kterém mají však úlohy nenulový čas pro změnu výroby. Cílem je minimalizace času dokončení poslední úlohy. Tento případ je ekvivalentní známému *problému obchodního cestujícího*.

Minimalizace účelových funkcí

Při vytváření výrobního rozvrhu je důležitá správná volba kritéria (či kritérií), na základě kterých se hodnotí, a následně optimalizuje výrobní rozvrh. Algoritmy rozvrhování výroby využívají při přiřazování úloh k výrobním zdrojům řadu heuristických

pravidel. Rozhodování může být realizováno např. podle délky operačního času, podle termínu dokončení zakázek, podle počtu zbývajících operací, či podle ceny. Mezi nej-používanější algoritmy rozvrhování výrobních zdrojů můžeme zařadit *Johnsonův*, *Jac-sonův*, *McNaughtonův*, *Min-max* algoritmus, či *Palmerovu heuristiku* [136].

Příklad rozvrhovacího modelu

Pro lepší představu je uveden příklad rozvrhování nad jedním strojem. Ačkoli není tato úloha v praxi příliš rozšířená - jde o jednoduchý příklad, na kterém je vhodné demon-strovat základ pro heuristiky používané v komplikovanějších modelových prostředích.

Při analýze problémů rozvrhování je možno komplexnější strojová prostředí rozložit na dílčí podproblémy. Existuje několik speciálních případů, které jsou tzv. triviální a jsou řešitelné v polynomiálním čase. Většina ostatních úloh však patří do skupiny nepolynomiálních problémů.

Mějme k dispozici jeden stroj, který zpracovává nejvíce jednu operaci z množiny $J = \{J_1, \dots, J_n\}$, přičemž jejich parametry jsou uvedeny v tabulce 5.2

	J_1	J_2	J_3	J_4	J_5	J_6
p_j	4	2	5	4	6	3
w_j	6	8	7	4	2	1

Tabulka 5.2: Časy potřebné pro zpracování úloh a jejich váhy

Předpokládejme, že pro optimalizaci rozvrhu modelové situace použijeme účelovou funkci celkového váženého času ukončení úloh. Tu je možné interpretovat také, jako *míru rozpracovanosti výroby*. Rozvrhovací problém by měl podobu: $1 || \sum w_j C_j \rightarrow \min$.

Algoritmus aplikuje pravidlo *wspt* (weighted shortest processing time), tedy výběr nejkratšího váženého času zpracování jako prvního. Podle tohoto pravidla se úlohy uspořádají jednoduše sestupně podle hodnoty w_j/p_j (výsledek viz obrázek 5.6).

Algorithm 1 Uspořádání dle nejkratšího váženého času zpracování ($1 || \sum w_j C_j \rightarrow \min$) [21]

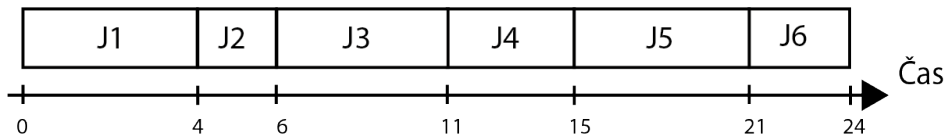
```

1: while  $w_1/p_1 \geq w_2/p_2 \geq \dots \geq w_n/p_n$  do
2:    $C_0 := 0$ 
3:   for  $i := 1$  to  $n$  do
4:      $C_i := C_{i-1} + p_i$ 
5:   end for
6: end while

```

Výpočetní náročnost rozvrhu se začne zvyšovat při uvažování precedenčního omezení mezi jednotlivými zakázkami. Je zřejmé, že uvedený algoritmus by při existenci precedenčních omezení generoval nepřijatelný rozvrh. Přírodným postupem by mohla

být úplná enumerace všech přípustných řešení a následný výběr toho s nejmenší hodnotou účelové funkce. Tento přístup by však vedl k výpočetně nerealizovatelnému exponenciálně rostoucímu prohledávání $n!$ permutací.



Obrázek 5.6: Gantův diagram pro rozvrh na jednom stroji

Další možné kritérium při hledání optimálního rozvrhu je minimalizovat vážený součet zpoždění úloh. Optimální rozvrh podle heuristik vzniká rozdělením úloh do dvou množin. Pro první množinu J platí podmínka pro všechny úlohy $d_j \leq \sum_{k=1}^n p_k$, tj. obsahuje jen úlohy, které mohou být v optimálním rozvrhu včas dokončeny. Tyto úlohy jsou řazené v pořadí neklesajících d_j tzn. $d_1 < d_2 < \dots < d_n$ podle pravidla *EDD* (nejpozdější termín dokončení jako první). Druhá množina úloh J obsahuje úlohy, které byly vyloučeny a nebude splněn jejich termín realizace. Tyto zpožděné úlohy již mohou být uspořádány v libovolném pořadí. V následující ukázce algoritmu vycházejme z předpokladu, že váhy všech zakázek jsou stejné, a tedy ve výpočtu nejsou uvažovány (Algoritmus s obecnými hodnotami vah je složitější a vede k úplné enumeraci). Seřazené úlohy podle termínu realizace d_j postupně přidáváme do množiny J až do chvíle, kdy $C_j > d_j$. Poté z množiny včas dokončených zakázek odstraníme tu s nejvyšší pracností.

Algorithm 2 Minimalizace váženého součtu zpoždění úloh ($1 \parallel \sum w_j U_j \rightarrow \min$) [21]

```

1: while  $d_1 < d_2 < \dots < d_n$  do
2:    $J := \emptyset; t := 0$ 
3:   for  $i := 1$  to  $n$  do
4:      $J := J \cup \{i\}$ 
5:      $t := t + p_i$ 
6:     if  $t > d_i$  then
7:       Find job  $j$  in  $J$  with a largest  $p_j$  value
8:     end if
9:      $J := J \setminus \{j\}$ 
10:     $J^d := J^d \cup \{j\}$ 
11:     $t := t - p_j$ 
12:   end for
13: end while

```

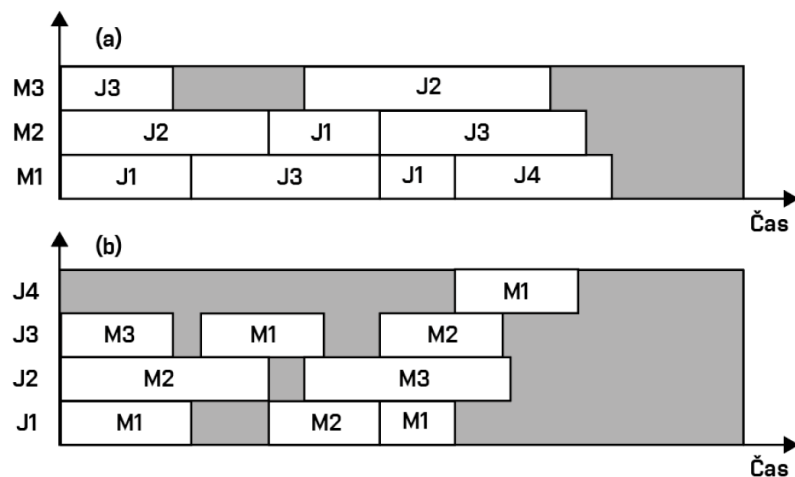
V praxi představuje toto kritérium míru výkonu rozvrhu, která se vyhodnocuje, avšak nemůže být jediným prostředkem měření splnění časů platnosti. Jeho nedostatkem je, že některé úlohy musí čekat na vykonání nepřiměřeně dlouhý čas. Z tohoto pohledu je vhodnější optimalizační kritérium celkového zpoždění $1 \parallel \sum w_j T_j$ [21].

5.5 Deterministické metody rozvrhování ve výrobě

Při rozvrhování výroby je každá zakázka rozložena na množinu uspořádaných dílčích operací (úloh) s definovaným termínem ukončení. Každá z těchto úloh vyžaduje zpracování na určitém stroji po stanovenou dobu. Všechny modely, které budou v této kapitole popsány vycházejí z předpokladů, že všechny údaje o problému (t.j. počet úloh, časy zpracování, časy zadání, termíny ukončení, váhy) jsou známy v čase *nula*. Plánovač má k dispozici všechny informace a může v daném čase určit celý plán. Toto paradigma se označuje jako offline plánování. Za jeho protipól pak můžeme označovat online programování, při kterém plánovač rozhoduje o pořadí vykonávaných úloh pouze na základě znalosti počtu pracujících strojů, nezná však počet ještě nezpracovaných úloh a nedisponuje informací o čase jejich trvání. V tomto případě se pak používají algoritmy LIST, případně Round Robin [82].

Cílem deterministického rozvrhování výroby je nalezení rozvrhu, který určuje kdy a která zakázka bude zpracovávána a na kterém konkrétním stroji, přičemž je kladen důraz na optimalizaci určité účelové funkce vycházející právě z konkrétního modelu rozvrhování.

Ve všech následujících problémech plánování se explicitně předpokládá konečný počet úloh (označujeme n s indexem j) a strojů (označujeme m s indexem i). Pokud úloha vyžaduje určitý počet kroků zpracování nebo operací, pak dvojice (i, j) označuje rozvrhovaný krok zpracování, respektive operaci úlohy j na stroji i . Rozvrhy mohou být obecně reprezentovány Ganttovými diagramy podobně jako na obrázku 5.7. Ganttovy diagramy mohou být konstruovány z pohledu využití jednotlivých strojů (5.7-a) či z pohledu realizace jednotlivých úloh (5.7-b).



Obrázek 5.7: Ganttův diagram: (a) orientovaný na stroje (b) orientovaný na úlohy

5.5.1 Lineární programování

Lineární programování² patří mezi úplné (přesné) metody operačního výzkumu, které optimalizují (minimalizují či maximalizují) kriteriální funkci vhodnou volbou koeficientů omezujících proměnných. Základním předpokladem pro použití této optimalizační metody je linearita kriteriální funkce a všech definovaných omezení. Úloha lineárního programování (*lineární program*) předpokládá dělitelnost výpočetních proměnných, jinými slovy proměnné mohou nabývat i neceločíselných hodnot. V případě, že existuje požadavek na nedělitelnost proměnných plynoucí z definice problému, mění se tento na tzv. *celočíselný lineární problém (LP)*. Může rovněž nastat situace, kdy jsou takto omezeny pouze některé proměnné. V takovém případě se problém nazývá *smíšený celočíselný LP (MILP)*. Problémy s celočíselnými proměnnými jsou obecně mnohem náročnější na výpočet.

5.5.2 Lineární modely pro plánování výroby

V literatuře jsou obecně uváděny tři různé přístupy, jak lze konstruovat lineární problém. Jedná se o *disjunktivní model*, který byl formulován již v roce 1960 [68], *model s časovou indexací* (time-indexed formulation) zmiňovaný ve stejném období [20] a [54] a zhruba stejně starý *model založený na hodnocení* (rank-based formulation) [119]. Další dohledatelné modely jsou pak většinou jen modifikacemi původních tří, resp. jejich kombinací.

Z teoretického porovnání těchto tří modelů, které nabízí práce [56], se nejefektivnější jeví první z modelů, tedy model disjunktivní. Důvodem je skutečnost, že obsahuje nejmenší počet celočíselných proměnných. Na druhém místě se logicky umístil model založený na hodnocení. Jako nejméně efektivní je hodnocen model s časovou indexací. Počet celočíselných proměnných je jedním, nikoli však jediným ukazatelem efektivity modelu. Mezi další patří např. možnosti relaxace celočíselného problému (doplnění o minimalizovanou neceločíselnou proměnnou, která zmírňuje celočíselné pravidlo). Je také prokázáno, že vhodnou volbou relaxace lze se stejným výpočetním výkonem řešit problémy s více než dvojnásobným počtem celočíselných proměnných. V následujícím textu jsou tyto problémy stručně popsány společně s výsledky výkonnostních experimentů.

²Lineární programování představil v roce 1937 Leonid Vitaljevič Kantorovič. Během 2. světové války jej dále rozvíjel jako prostředek pro plánování komplexních armádních problémů. Po válce se na dalším vývoji podíleli George Bernard Dantzig, který definoval simplexovou metodu, a John von Neumann, který popsal princip duální úlohy. V roce 1975 obdržel Kantorovič za svůj přínos k řešení problémů optimálního přidělování zdrojů Nobelovu cenu za ekonomii.

Disjunktivní model

Následující disjunktivní model byl převzat a upraven z [56]. Jsou v něm použity následující proměnné:

- $r_{i,j}$ je celočíselná hodnota udávající čas startu zpracování úlohy j na stroji i .
- $z_{i,j,k}$ je rovno 1, pokud je na stroji i úloha j zpracována dříve než úloha k .

Model je možné definovat takto:

$$\mathbf{min} \quad C_{max} \quad (5.1)$$

$$s.t. \quad r_{i,j} \geq 0 \quad \forall j \in J, i \in M \quad (5.2)$$

$$r_{\sigma_h^j,j} \geq r_{\sigma_{h-1}^j,j} + p_{\sigma_{h-1}^j,j} \quad \forall j \in J, h = 2, \dots, m \quad (5.3)$$

$$r_{i,j} \geq r_{i,k} + p_{i,k} - V \cdot z_{i,j,k} \quad \forall j, k \in J, j < k, i \in M \quad (5.4)$$

$$r_{i,k} \geq r_{i,j} + p_{i,j} - V \cdot z_{i,j,k} \quad \forall j, k \in J, j < k, i \in M \quad (5.5)$$

$$C_{max} \geq r_{\sigma_m^j,j} + p_{\sigma_m^j,j} \quad \forall j \in J \quad (5.6)$$

$$z_{i,j,k} \in \{0, 1\} \quad \forall j, k \in J, i \in M \quad (5.7)$$

Jako první (5.1) je uvedena kritériální funkce. Množina omezení (5.2) zajišťuje, že počáteční čas zpracování každé úlohy bude větší nebo roven 0, tedy že nebudou uvažovány záporné časy. Precedenční omezení (5.3) zajišťuje, že všechny operace úlohy jsou plánovány ve správném pořadí. Omezení (5.4) a (5.5) vylučují, aby na jednom stroji v jednom čase byla naplánována více než jedna úloha. Za V je nutno dosadit dostatečně velké číslo, což může být např. $V = \sum_{j \in J} \sum_{i \in M} p_{ij}$. Toto rozhodnutí lze vyjádřit tak, že doba dokončení kterékoli operace nemůže přesáhnout součet času zpracování všech operací. Omezení (5.6) definuje, že celkový čas realizace musí být nejméně tak dlouhý, aby každý stroj zvládl zpracovat všechny požadované úlohy. Vzhledem k tomu, že proměnná z může nabývat pouze hodnot 0 nebo 1 (omezení 5.7), jedná se tedy o model, v němž se vyskytují celočíselné proměnné. Vzhledem k disjunktivním omezením 5.4 a 5.5 neodpovídá model definici jednoduchého celočíselného problému, ale tzv. *disjunktivnímu problému*. Moderní prostředky pro řešení MILP úloh však takový typ problémů dovedou řešit.

Model s časovou indexací

Původní myšlenka modelu s časovou indexací byla publikována již v roce 1959 v [20]. V [54] a [56] je popsána jeho modifikovaná efektivnější varianta. Pro každý stroj, každou úlohu a každé časové okno v tomto modelu existuje jedna optimalizační proměnná r , přičemž $r_{i,j,t}$ je rovno 1, pokud na stroji i byla úloha j spuštěna v čase t .

$$\min C_{max} \quad (5.8)$$

$$s.t. \sum_{t \in H} r_{i,j,t} = 1 \quad \forall j \in J, i \in M \quad (5.9)$$

$$\sum_{t \in H} (t + p_{ij}) \cdot r_{i,j,t} \leq C_{max} \quad \forall j \in J, i \in M \quad (5.10)$$

$$\sum_{j \in J} \sum_{t' \in T_{i,j,t}} r_{i,j,t'} \leq 1 \quad \forall i \in M, t \in H, kde$$

$$T_{ijt} = \{t - p_{ij} + 1, \dots, t\} \quad (5.11)$$

$$\sum_{t \in H} (t + p_{\sigma_{h-1}^j}) \cdot r_{\sigma_{h-1}^j, j, t} \leq \sum_{t \in H} t \cdot r_{\sigma_h^j, j, t} \quad \forall j \in J, h = 2, \dots, m \quad (5.12)$$

$$r_{i,j,t} \in 0, 1 \quad \forall j \in J, i \in M, t \in H \quad (5.13)$$

Kriteriální funkce je uvedena v (5.8). Omezení (5.9) specifikuje, že každá úloha j může být na každém stroji i spuštěna právě jednou v intervalu odpovídajícím délce plánovaného rozvrhu. Omezení (5.10) pak definuje, že celkový čas realizace musí být nejméně tak dlouhý, aby každý stroj zvládl zpracovat všechny požadované úlohy. Omezení (5.11) zajišťuje, že žádný stroj j není v žádném časovém okamžiku t nadměrně vytížen, tj. pravidlo zamezuje naplánování více než jedné úlohy jednomu stroji do stejného časového intervalu. (5.12) je precedenční omezení, které vynutí, že všechny operace dané úlohy jsou plánovány ve správném pořadí. (5.13) pak zajišťuje důležitou celočíselnost výsledného plánu.

Jednoduché příklady lineárních problémů lze řešit grafickou metodou. Do prostoru se zakreslí jednotlivé poloprostory vzniklé lineárními omezeními a vznikne mnohostěn, v jehož středu jsou přípustná řešení. Dále je nutné zakreslit vektor kriteriální funkce nebo jeho normály a na základě jeho směru na mnohostěnu zjistíme bod (nebo více bodů), který je maximálním nebo minimálním řešením daného problému. Jednoduché řešení má však tu nevýhodu, že lze rozumně použít maximálně pro 3 proměnné (v takovém případě uvedené modelování používá trojrozměrný prostor). Možnosti užití této metody se tak omezují pouze na demonstraci vztahů platných v lineárním modelu a naznačení možných univerzálních principů řešení.

Řešení simplexovou metodou

Nejznámější a často používanou metodou pro řešení lineárních problémů je simplexová metoda, kterou představil v roce 1947 americký matematik George Bernard Dantzig. Jde o iterační metodu, tedy metodu řešení po krocích. Tyto kroky postupují od výchozího základního řešení optimalizační úlohy k dalšímu základnímu řešení s lepší (v případě minimalizace nižší) hodnotou účelové funkce. Při těchto krocích se vynechávají nepřípustná řešení a také řešení, která (v případě minimalizace) nesnižují hodnotu účelové funkce. Ve výsledku to znamená, že podobně jako při řešení optimalizačních úloh pomocí grafického znázornění se zkoumají pouze vrcholy množiny přípustných

řešení a to tak, že s každým krokem (iterací) se výpočet dostává blíže k optimálnímu řešení.

V porovnání s grafickou metodou řešení je simplexová metoda takřka neomezená v počtu neznámých, avšak s jejich vzrůstajícím počtem vzrůstá i časová náročnost na její výpočet. Stále však patří k nejefektivnějším způsobům řešení lineárních optimalizačních úloh. Výpočetní metody spolu s dnešními technickými prostředky zvládnou bez potíží řešit rozsáhlé lineární problémy s mnoha stovkami tisíc proměnných [111]. Základní překážkou pro použití této metody v řešení rozvrhovacích problémů je fakt, že metoda je použitelná pouze na čistě lineární problémy, zatímco rozvrhovací problémy jsou vždy celočíselné (ILP), případně smíšené (MILP).

5.5.3 Řešení metodami Branch and Bound

Branch and bound metody (tzv. metody větví a mezí) jsou jedny z nejrozšířenějších pro řešení úloh celočíselného programování. Metody jsou založeny na paradigmatu *rozděl a panuj*, tedy na rozkládání komplexního problému na podproblémy a jejich řešení (případně dalšího rozkladu). Během řešení vzniká stromová struktura popisující stavový prostor problému, který se však neprochází celý, neboť pro reálné problémy to kvůli expanzi stavů není možné. Namísto toho se určuje, které uzly stromu jsou nejslibnějšími kandidáty tak, že se odhaduje nejlepší hodnota řešení pro všechny následníky každého uzlu. V případě, že tato hodnota je horší (tedy v případě minimalizační úlohy větší), než nejlepší dosud nalezené řešení, je celý uzel i se svými následovníky odstraněn. V nejhorsím případě má sice metoda exponenciální náročnost (naskytne-li se nutnost opravdu projít celý stavový prostor), v naprosté většině případů je ale řešení nalezeno mnohem rychleji a je nutno skutečně sestrojít a prohledat pouze malou část stavového prostoru. Název je odvozen od větvení (*branching*), tedy vytvoření následovníků konkrétního uzlu, a hledání mezní hodnoty (*bounding*) řešení pro všechny následovníky příslušného uzlu. Posledním důležitým mechanismem typickým pro tyto metody je tzv. prořezávání (*pruning*), kdy jsou odstraněny uzly, jejichž žádný následovník nepředstavuje optimální řešení problému.

Princip činnosti

V každém okamžiku při běhu algoritmu lze průběh řešení popsat pomocí podmnožiny *Live* definující část stavového prostoru problému, který dosud nebyl prohledán, a hodnoty minimálního dosud nalezeného celočíselného řešení. Na začátku tato podmnožina obsahuje celý stavový prostor, nejlepší nalezené řešení minimalizačního problému z_{MIP} je nastaveno na ∞^3 a strom řešení obsahuje pouze kořenový prvek. Při každé iteraci metody je z *Live* odstraněna jedna úloha – *Current*, se kterou se dále pracuje (viz řádek 5 v algoritmu 3).

³Alternativně je počáteční řešení možno získat pomocí heuristických metod.

Pro úlohu *Current* se vypočte optimální relaxace $\mathbf{x}_{Current}$ (7) a je-li to možné, rovněž hodnota celočíselného řešení $x_{Current}$ (8). Pokud je hodnota kritéria vypočtené relaxace větší či rovna nalezenému nejlepšímu celočíselnému řešení ($\mathbf{c}^T \mathbf{x}_{Current} \geq z_{MIP}$, 9), neexistuje již dále možnost, aby v kterémkoli podprostoru bylo nalezeno lepší řešení, celá úloha může být zapomenuta a zpracování pokračuje výběrem další úlohy *Current*.

Byla-li nalezená hodnota celočíselného řešení menší než nejlepší dosud nalezené celočíselné řešení ($\mathbf{c}^T \mathbf{x}_{Current} < z_{MIP}$, 13), stává se tato nová hodnota nejlepším dosud nalezeným řešením. Navíc jsou z množiny *Live* odstraněny všechny úlohy q , pro které nově platí, že $z_q \geq z_{MIP}$ (16).

Jakmile však hodnota celočíselného řešení pro daný uzel nebyla nalezena, nelze v aktuálním kroku rozhodnout o existenci možného optimálního řešení na zkoumaném podprostoru. Dochází tedy k dělení – úloha *Current* je přidáním dalších pravidel rozdělena na vzájemně disjunktní podúlohy $Current_1 \dots Current_k$ (19) a všechny tyto jsou přidány do *Live*.

Jakmile se množina *Live* vyprázdní, stává se nejlepší nalezené celočíselné řešení \mathbf{x}_{MIP} optimálním globálním řešením. Pro hodnotu kritéria pak platí $z_{MIP} = \mathbf{c}^T \mathbf{x}_{MIP}$. Nebylo-li však nalezeno ani jedno celočíselné řešení ($z_{MIP} = \infty$), problém nemá řešení.

Efektivita řešení modelů rozvrhování

Míra úsilí, které je potřeba vynaložit při hledání optimálního řešení, přirozeně vzrůstá s komplexností úlohy. V případě modelu s časovou indexací si lze povšimnout, že počet binárních proměnných prudce stoupá se zvětšující se velikostí problému (pro 20 strojů, 50 úloh a např. 120 časových intervalů se jedná o 120000 proměnných). Ve výsledku je efektivně možné pomocí tohoto modelu řešit pouze rozvrhy méně komplexních problémů. Výpočetní čas daného algoritmu je definován jako horní mez počtu iterací vzhledem k velikosti vstupu.

5.6 Stochastické metody pro plánování výroby

5.6.1 Konstruktivní metody

Jedná se zpravidla o jednoduché, zřetelné, a tudíž jednoduše vysvětlitelné metody. Díky těmto vlastnostem se řadí mezi nejpoužívanější v praxi [53]. Plánování začíná s *prázdným řešením* (plánem) a použitím metod, jako jsou *prioritní pravidla*, *rozvrhování dle úzkého místa*, či *dle zakázky s největší sumou procesních časů*. Konstruktivní metody řeší rozvrhovací problémy při respektování daných omezení (technologickým postupem, materiálovými požadavky, požadavky na seřízení stroje), a to v konečném počtu kroků.

Algorithm 3 Příklad branch-and-bound algoritmu

```
1:  $z_{MIP} \leftarrow \infty$ 
2:  $\mathbf{x}_{MIP} \leftarrow 0$ 
3:  $Live \leftarrow$  celá MIP úloha
4: while  $Live \neq \emptyset$  do
5:   Vyber úlohu  $Current | Current \in Live$ 
6:    $Live \leftarrow Live \setminus Current$ 
7:   Vypočti relaxaci  $\underline{\mathbf{x}}_{Current}$ 
8:   Vypočti celočíselné řešení  $\mathbf{x}_{Current}$ 
9:   if  $\mathbf{c}^\top \underline{\mathbf{x}}_{Current} \geq z_{MIP}$  then
10:     Odstraň úlohu  $Current$ 
11:     Skoč na řádek 5
12:   end if
13:   if  $\mathbf{c}^\top \mathbf{x}_{Current} < z_{MIP}$  then
14:      $z_{MIP} \leftarrow \mathbf{c}^\top \mathbf{x}_{Current}$ 
15:      $\mathbf{x}_{MIP} \leftarrow \mathbf{x}_{Current}$ 
16:      $\forall q | q \in z_q \geq z_{MIP} : Live \leftarrow Live \setminus q$ 
17:     Skoč na řádek 5
18:   end if
19:   Rozvětvi  $Current$  na  $Current_1, \dots, Current_k$ 
20:    $Live \leftarrow Live \cup Current_i, \forall i \in 1 \dots k$ 
21: end while
22: if  $z_{MIP} < \infty$  then
23:   Optimální řešení je  $\mathbf{x}_{MIP}$ 
24:   Hodnota kritéria je  $z_{MIP}$ 
25: end if
```

Prioritní pravidla

Rozvrhování podle prioritních pravidel je založeno nejčastěji na generování rozvrhů pomocí algoritmu Giffler&Thompson [109]. Tento algoritmus generuje tzv. *aktivní rozvrh* (tedy takový, ve kterém není možné vytvořit jiný rozvrh změnou sledu alespoň jedné operace na stroji tak, aby byla odvedena dříve, aniž by došlo k opoždění jiné operace) tím způsobem, že rozvrhuje jednu operaci po druhé na základě výběru právě dostupných operací. Právě dostupná operace je taková, která je nerozvrhnutá a je jako první v technologickém postupu. Dostupnost dále ovlivňují další omezení, jako např. čas uvolnění materiálu do výroby, směnnost, připravenost pracovníků, aj. Pro takto dostupné operace se vybírá stroj, který je může zpracovat, a u kterého se předpokládá nejdřívejší čas dokončení. Na tomto stroji je pak nutné rozhodnout, jaké z dostupných operací dát přednost. Tato rozhodnutí probíhají na základě prioritních pravidel vyplývajících z ① obecných požadavků na výrobu a ② vlastností technologického postupu.

V praktických aplikacích se často vyskytují prioritní pravidla dle požadavků na výrobu. Systémy APS často umožňují ruční přiřazení priorit zakázkám na základě např: požadovaného termínu odvedení zakázky zákazníkovi, termínu přijetí zakázky, penále z nedodržení termínů, či druhu zakázky.

Přestože tento způsob rozvrhování nejčastěji odpovídá požadavkům podniku, není příliš vhodný pro dílenské řízení výroby. Pokud je např. nutné vyrábět na jednom stroji dvě zakázky, které mají stejné datum dodání zákazníkovi, a tudíž stejnou prioritu, nelze rozhodnout pořadí na výrobním stroji. Často se tak používají prioritní pravidla vycházející z vlastností daného technologického postupu. Mezi základní patří:

- **Shortest Processing Time (SPT)** - zakázka s nejkratším procesním časem;
- **Most Work Remaining (MWKR)** - zakázka, na které zbývá nejvíce práce;
- **Least Work Remaining (LWKR)** - zakázka, na které zbývá nejméně práce;
- **First Come First Served (FCFS)** - zakázka s nejdřívejším časem příchodu na pracoviště;
- **Random Select (RS)** - náhodný výběr zakázky.

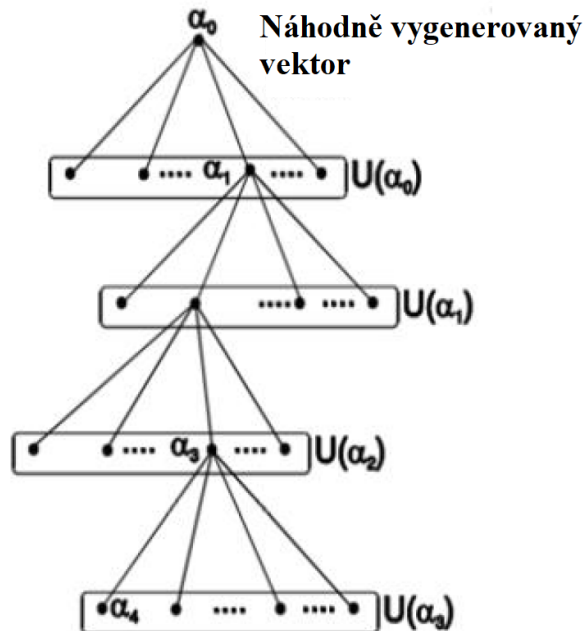
Výhodou tohoto způsobu rozvrhování je rychlost a jednoduchost generování rozvrhů, nevýhodou pak skutečnost, že nelze zaručit nalezení optimálního řešení a možnost, že lze v případě generování rozvrhů pomocí různých prioritních pravidel dosáhnout stejného řešení. Nejvhodnější nasazení této metody je v případě, kdy se vyžaduje řešení v krátkém čase nebo pro generování počátečních řešení pro prohledávací heuristické metody [53, 109, 82].

5.6.2 Lokální prohledávání

Tato metoda patří mezi neúplné metody založené na heuristickém principu. Algoritmus prohledává stavový prostor optimalizačních úloh pouze lokálně, proto z principu přináší pouze sub-optimální řešení. I tak jsou v praxi často dobře použitelná. Metoda nalézá uplatnění zejména v komplexních úlohách, které zcela selhávají kvůli enormnímu nárůstu potřeby výpočetního výkonu. Kritický parametr úspěšnosti této metody je silně závislý na kvalitě zvolené funkce sousednosti tak, aby mohl být efektivně prohledáván stavový prostor.

Tato funkce přiřazuje každému stavu skupinu dalších (sousedních) stavů v prohledávaném prostoru. Algoritmus začíná na náhodně vygenerovaném místě (případně může počáteční místo generovat heuristika). Z tohoto místa se přesouvá do sousedního stavu, který je vzhledem k hodnotě optimalizační funkce výhodnější (tedy stavu s minimální hodnotou funkce mezi všemi sousedními). Nový stav se použije v následujícím iteračním kroku jako střed nové oblasti. Proces se opakuje po konečný počet kroků s tím, že nejlepší nalezené řešení se uchovává. Na obrázku 5.8 jsou znázorněny iterace horolezeckého algoritmu. α_n označuje lokální optimum a $U(\alpha)$ zkoumané okolí.

Zásadním nedostatkem tohoto algoritmu je riziko, že uvázne v lokálním optimu z pohledu kvality sousedních řešení. Tento problém je možné částečně eliminovat zvětšením množiny sousedů (počtu elementů v $U(\alpha)$), případně opakovat metodu pro různě vygenerované počáteční řešení. V literatuře je možné nalézt i pokročilejší formy tohoto algoritmu, např. *horolezecký algoritmus s učením*, který přidává parametr určující rychlost konvergence při zlepšování optimality.



Obrázek 5.8: Schématické znázornění generování lokálních okolí [137]

Na podobném principu fungují i další stochastické metody optimalizace, např. *tabu search* (metoda zakázaného prohledávání) a *simulated annealing* (metodu simulovaného

žihání) [53, 112].

5.6.3 Meta-heuristické algoritmy

Evoluční algoritmy jsou meta-heuristické optimalizační algoritmy založené na optimalizaci výsledné populace (tedy množiny řešení) pomocí principů známých z biologie, jako jsou mutace, křížení, přirozený výběr a přežití nejsilnějších jedinců za účelem nalezení nejlepších jedinců skrze generace (iterace algoritmu).

Tyto algoritmy jsou založeny na zjednodušeném praktikování biologických principů a principu cílené optimalizace, tedy zlepšování určitého znaku jedinců - účelové funkce. Prohledávaný prostor G je abstrakce množiny všech možných DNA řetězců v přírodě a jejich znaky $g \in G$ (v rozvrhování pořadí operace) hrají roli přírodního genotypu. Populace jedinců x (řešení problému) se vyvíjí (prohledává prostor X) pomocí mutace a křížení předem definovaným směrem daným silou nutnou pro přežití (účelová funkce).

Do kategorie evolučních algoritmů se řadí *genetické algoritmy*, *genetické programování* a *evoluční strategie* [53, 137].

5.7 Kritéria výběru vhodné matematické metody

Volba metody pro výpočet optimálního řešení je vždy kompromisem. K tomu, abychom mohli správně rozhodnout, je potřeba poznat do hloubky jak řešený problém, tak samozřejmě jednotlivé metody. Během výběru je nutné odpovědět si na základní otázky a z odpovědí na ně vyplyne použitelnost jednotlivých metod. Existují čtyři základní kritéria [80]:

- **Splnitelnost či optimalizace** - postačuje nalezení jakéhokoli řešení splňujícího všechny zadané ohraničení nebo je potřebné nalézt řešení, které je optimální na základě zadaného kritéria? Lineární programování spolu s metodou Branch and Bound umožňuje nalézt optimální řešení problémů, které je možné popsat lineárními rovnicemi i kriteriální funkcí. Neúplné metody, jako horolezecký algoritmus či simulované žihání a další jsou určeny zejména pro nalezení existujícího přípustného řešení.

V předchozím odstavci zmíněná ohraničení mohou být dvou druhů - tzv. tvrdá, neboli technologická ohraničení (např. omezení stanovující, že v jednom čase může být na stroji zpracovávána pouze jedna úloha) a měkká, neboli preferenční ohraničení. Zatímco tvrdá ohraničení z podstaty nemohou být ve výsledném optimálním řešení porušena, měkká ohraničení je možné v optimalizačním problému (např. lineárním programu) započítat do nákladů, tedy do kriteriální funkce. Pak se optimalizuje mimo hlavní kritérium také vážená suma počtu či doby porušení těchto preferenčních ohraničení.

- **Výpočetní čas či optimalita řešení** - volba vhodného algoritmu bude závislá na rozhodnutí, který požadavek je v konkrétním případě důležitější. Pokud klademe prioritu na zaručené nalezení optimálního řešení bez ohledu na časové ohraničení doby výpočtu, je vhodné použít některou z úplných metod. Problémem však zůstává, že časová náročnost těchto metod exponenciálně narůstá s velikostí rozvrhovací úlohy. Při dlouhodobém plánování není časová náročnost výpočtu až tak kritický parametr, důraz je kladen zejména na nalezení globálně optimálního řešení. Naopak, v časově kritických situacích rozvrhování výroby (např. při vzniku mimořádné události) je použití některé ze stochastických (neúplných) metod plně postačující i za cenu nalezení pouze sub-optimálního řešení.
- **Specifikace problému** - volba vhodné metody je také závislá na tom, jakým způsobem je problém reprezentován. Lineární programování pracuje s množinou lineárních nerovnic a lineární kriteriální funkcí. Metodu branch and bound je možné použít na každý optimalizační problém, je-li k dispozici způsob, jakým ohodnotit kvalitu částečného řešení. Stochastické metody, např. Hill climbing či simulované žíhání je možné uplatnit na široké spektrum úloh, avšak jejich kvalita je závislá na funkci sousednosti, kterou metody využívají pro nalezení nových potenciálních řešení.
- **Volba algoritmu a implementace** - algoritmy lineárního programování a branch and bound jsou v literatuře dobře definovány a jejich implementace je poměrně přímočará (ačkoli není vždy jednoduchá). Stochastické metody pak již vyžadují větší odborný vklad od řešitele, neboť např. v případě Hill climbing je nutno nalézt kvalitní funkci sousednosti (najít nějakou funkci většinou není problém, avšak nalézt takovou, která bude efektivně prohledávat stavový prostor je náročná úloha).

Výpočetní náročnost rozvrhovacích úloh

Míra úsilí, kterou je potřeba vynaložit pro hledání optimálního řešení přirozeně roste s mírou komplexnosti úlohy. Výpočetní čas daného algoritmu je definován jako horní mez počtu iteračních průchodů algoritmu vzhledem k velikosti vstupu. Z hlediska výpočetní náročnosti rozlišujeme dvě třídy algoritmů - **P** a **NP**. Třída P obsahuje všechny rozhodovací problémy, které jsou řešitelné v polynomiálním čas. To znamená, že na řešený problém existuje algoritmus, jehož výpočetní čas je shora ohraničen polynomiální funkcí. Např. uváděný algoritmus pro minimalizace kriteriální funkce $1 || \sum w_j C_j$ je polynomiální. Třída NP problémů obsahuje takové problémy, pro které nebyl algoritmus s polynomiální náročností nalezen. Problém $1 || (\sum w_j U_j \rightarrow \min)$ je problémem třídy NP. Tyto algoritmy jsou výpočetně náročné, a proto je žádoucí vyhnout se úplné enumeraci, čímž dojde ke snížení časové a paměťové náročnosti algoritmu. K tomuto účelu jsou využívány další metody založené na heuristickém přístupu, např. již popisovaná metoda větví a mezí (Branch and Bound) [82].

5.8 Shrnutí kapitoly

Pro zajištění konkurenceschopnosti musí být podniky plánovat svou výrobu na různých úrovních. Dlouhodobé strategické a střednědobé plánování jsou základem pro určení směru rozvoje podniku. Operativní plánování je každodenní, opakující se činnost, při které vzniká plán popisující výrobu zakázek na jednotlivých strojích či pracovištích. První plánovací metody začaly vznikat v polovině minulého století. S postupným vývojem prostředků výpočetní techniky a vznikem nových metod mění také způsoby a procesy plánování.

Základem řešení plánovacího problému je tvorba modelu a následně volba a nasazení vhodné výpočetní metody, ať už deterministické (např. lineární programování), nebo stochastické (např. konstruktivní metody). Vzhledem k tomu, že algoritmy pro výpočet optimálních plánů patří mezi tzv. NP problémy, je nutné klást důraz na využití vhodné metody, ale také na způsob konstrukce modelu.

I při správné volbě metody a modelu může nastat situace, kdy optimální plán není možné kvůli omezení výpočetního výkonu získat. V takovém případě přichází ke slovu zcela jiný - distribuovaný přístup popsany v následující kapitole.

Kapitola 6

Inteligentní výrobní systémy

Všechny algoritmy pro vytváření výrobních rozvrhů, zmiňované v předchozí kapitole mají jedno společné - výrobní plán vzniká nalezením *řešení* (optimálního, či sub-optimálního) problému. Výpočet tohoto řešení je vždy realizován v jediném - ústředním (řídícím) členu s dostatečným výpočetním výkonem. Pokud však centrální člen nemá dostatečný výkon, či v rámci architektury systému žádný takový člen neexistuje, je možné využít jinou skupiny algoritmů. Popisem takových архитектур a algoritmů se zabývá právě tato kapitola.

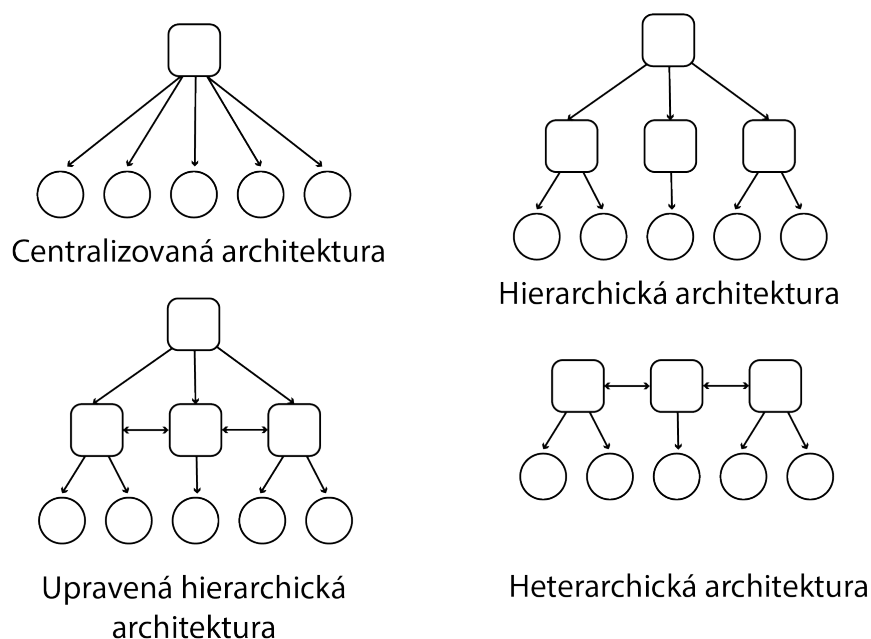
6.1 Typy řídicích архитектур výrobních systémů

Co vlastně označuje pojem *výrobní architektura*? Dle [64] je architektura systému definována jako *rámeček nebo sada pravidel a pokynů pro management vývoje a provozu komplexních systémů*. Tuto definici je možné převzít i pro výrobní řídicí architektury. V posledních letech se uskutečnilo velké množství výzkumů v akademické i průmyslové sféře s cílem zvýšit výkonnost výrobních systémů a schopnost jejich reakce na rychle se měnící požadavky zákazníků [132, 137].

Kromě základních požadavků na zvýšení produktivity a kvality při co nejnižších nákladech je dnes rovněž kladen důraz na rychlou reakci systému při výskytu různých výjimečných stavů systému, odolnost vůči poruchám a maximální přizpůsobitelnost změny konfigurace systému a prostředí. V oblasti řízení výroby se běžně rozlišují čtyři typy řídicích архитектур:

- Centralizovaná architektura
- Hierarchická architektura
- Upravená hierarchická (hybridní) architektura
- Heterarchická architektura

Na obrázku 6.1 jsou tyto čtyři typy architektur znázorněny. Čtverce zastupují řídicí komponentu/komponenty a kruhy označují řízené entity. Šipky znázorňují směry informačních toků.



Obrázek 6.1: Typy řídicích architektur

6.1.1 Centralizovaná architektura

Evoluci řídicích architektur přinesla centralizovaná architektura (obr. 6.1 vlevo nahoře). Při využití tohoto typu architektury je informační tok (tok příkazů) směřován z jediné řídicí jednotky do všech přidružených řízených jednotek (např. výrobních strojů a zařízení). Centralizovaná architektura je implementačně jednoduchá a v případě malého množství řízených zařízení také efektivní. Její výhodou je centrální úložiště relevantních informací a dat, a také možnost provádět relativně jednoduché optimalizace celého řízeného procesu.

Pokrok v oblasti automatizované výroby klade stále větší důraz na výkonnost a flexibilitu, což odhaluje nevýhody centralizovaných řešení. Při zvyšování počtu řízených výrobních strojů a zařízení dochází ke zvyšování komplexnosti celého systému, což se projevuje zejména zvyšováním časové odezvy řídicí jednotky. Jednotka za účelem optimalizace celého procesu dopředu plánuje na určitém časovém horizontu operace pro jednotlivé systémy. Při zvyšujícím se počtu systémů ale dochází k růstu komplexnosti optimalizačního problému, který se stává jen obtížně řešitelný v požadovaném omezeném čase. Jelikož je řídicí jednotka kritickým bodem celého systému, je problematické provádět její hardwarové či softwarové změny za účelem škálování systému. S tím také souvisí nízká odolnost systému vůči poruchám, neboť v případě selhání řídicí jednotky se stane nefunkčním také celý řízený systém.

6.1.2 Hierarchická architektura

Hierarchická architektura (obr. 6.1 vpravo nahoře) se vyznačuje dekompozicí jediné složené řídicí jednotky do několika menších (podřízených) jednotek. Snahou při návrhu této architektury bylo distribuovat rozhodovací proces do více řídicích úrovní. Výhodou tohoto uspořádání je lepší rozložení výpočetní zátěže do více řídicích jednotek oproti centralizované architektuře¹. Hierarchická architektura je založena na principu *top-down*, tedy že řídicí informace jsou předávány řídicími jednotkami na vyšších úrovních k jednotkám na nižších úrovních, které následně vracejí zpětnovazební informace.

Čím výše v hierarchii se daná řídicí jednotka nachází, tím abstraktnější bývají příkazy, které odesílá a přijímá. Vzhledem k typu relace *master-slave* mezi jednotlivými úrovněmi řídicích jednotek je architektura robustní, lehce pochopitelná a předvídatelná. Čas odezvy systému je nižší, než v případě centralizované architektury.

Nevýhodou architektury je, že jednotlivé jednotky mají malou či žádnou autonomii. Neumějí reagovat na neurčitosti, které přicházejí z vnějšího prostředí a v případě jejich výskytu se výkonnost systému výrazně snižuje. V případě požadavku na hardwarové či softwarové změny je nutné zastavit provoz celého systému a aktualizovat řídicí jednotky na všech úrovních řízení novými informacemi. Při výpadku některé z podřízených jednotek na nižší úrovni pak nemusí dojít k výpadku celého systému. Odolnost vůči poruchám je tedy částečně zvýšená. Pokud je mimo provoz centrální jednotka, je vždy zasažen celý systém.

Typickým představitelem tohoto druhu architektury jsou klasické výrobní systémy, které jsou na každé úrovni (řízení podniku, řízení výroby, apod.) řízeny samostatně centrální jednotkou, avšak jsou koordinovány nadřazenou řídicí jednotkou (PLC je koordinováno MES, který je koordinován ERP, atd.)

6.1.3 Upravená hierarchická architektura

Dalším krokem ve vývoji řídicích architektur je upravená hierarchická řídicí architektura (resp. hybridní architektura), viz na obr. 6.1. Od čistě hierarchické architektury se tato liší přidáním horizontálního toku informací. Ten umožňuje realizovat synchronizaci a kooperaci mezi jednotkami na stejné úrovni, které mezi sebou mohou sdílet data. To vede ke zvýšení agility, flexibility a odolnosti vůči poruchám. Celková efektivita systému je formována jeho horními úrovněmi, neboť ty disponují globálním nadhledem, zatímco robustnost a rychlá odezva na poruchy je zabezpečena nižšími řídicími úrovněmi.

Upravená architektura sdílí výhody a nevýhody hierarchické architektury s několika rozdíly. Zde je povolen jistý stupeň lokální autonomie podřízených jednotek, čímž se snižuje výpočetní zátěž nadřazené jednotky. Naproti tomu je tato organizační struktura po návrhové a implementační stránce složitější než hierarchická architektura.

¹Centralizovaná architektura je speciálním případem hierarchické architektury s jedinou úrovní řízení

6.1.4 Heterarchická architektura

Centralizované a hierarchické formy řídicích architektur dokáží jen velmi neefektivně reagovat nejen na vnější podněty či změny ve výrobě, ale také na vnitřní podněty systému, např. poruchy. Jejich řídicí algoritmy bývají navrženy pro vypořádání se s dopředu stanovenými situacemi, a jakákoli změna výrobních podmínek může znamenat dlouhé časové odezvy, či dokonce selhání celého systému.

Důležitým krokem ve vývoji flexibilnějšího řídicího systému vzhledem k měnícím se podmínkám ve výrobě je heterarchická řídicí architektura (6.1). Ta umožňuje pouze horizontální tok informací, a vůbec nenabízí vertikální hierarchické uspořádání. Tato architektura je plně decentralizovaná, je tvořena skupinou na sobě nezávislých entit, které disponují vlastními vnitřními algoritmy ve svých řídicích jednotkách. Dekompozice systému na soubor na sobě nezávislých komponent je řešena přirozeně podle funkční příslušnosti či lokality.

Velkou výhodou této architektury je, že rozhodnutí se vykonávají lokálně v místě sběru informací. V systému nefiguruje globální databáze, a pracuje se pouze s takovými daty, kterými jednotlivé moduly architektury disponují.

Spolupráce mezi subjekty je zabezpečena prostřednictvím vyjednávacího postupu. Výrobní buňky navzájem vyjednávají o vytvoření výrobního plánu, přičemž zohledňují vnější podněty, změny ve výrobě či poruchy. Mezi hlavní výhody tohoto uspořádání patří vysoká odolnost vůči poruchám, flexibilita a vyšší míra případné redundance. Odolnost vůči poruchám je určující vlastností této architektury a je jí dosahováno implicitně díky lokální autonomii komponent. Selhání některé z nich pak nevede k selhání celého systému, ale k rekonfiguraci zbývajících výrobních kapacit. Systém tedy degraduje postupně.

Heterarchická architektura má také své nedostatky. Mezi hlavní patří nedeterministické chování systému. To je nepředvídatelné a může vést k selhání systému. Navíc je matematicky nemožné předpovídat budoucí stavy systému. Absence řídicího členu také znamená, že není možné provádět globální optimalizaci chování systému, ale je možná pouze lokální forma optimalizace.

Další problémy souvisí s tím, že heterarchicky organizované systémy jsou často heterogenní a v nich použité komponenty nemusí být vzájemně kompatibilní. Problémem je také nedostatečné vymezení postupů a norem pro koordinaci a vývoj distribuovaných konceptů řízení [30].

Při syntéze heterarchických výrobních systémů je velmi důležité kombinovat důkladnou funkční analýzu ve směru shora-dolů a uspořádání toku implementačních rozhodnutí ve směru zdola nahoru. Důraz je kladen na vytvoření kooperativní heterarchie, která splňuje následující pravidla:

- Subjekty mají stejná práva přístupu ke zdrojům

- Subjekty mají vzájemně nezávislé způsoby fungování
- Všechny subjekty si mohou vzájemně předávat informace
- Všechny subjekty přísně dodržují stanovená pravidla.

Klíčovou podmínkou pro každou entitu v systému je podílet se na maximalizaci výkonnosti celého systému a zabezpečovat jeho optimální provoz a přežití v havarijních stavech [30].

6.2 Definice inteligentního výrobního systému

Vymezení termínu *inteligentního výrobního systému* (IMS, Intelligent Manufacturing System) není napříč literaturou jednotná. Za inteligentní se dá považovat systém s vlastnostmi, pomocí nichž je schopen pochopit příčiny změn na svém vstupu a tyto informace využít pro učení, a jehož vlastnosti se s opakujícím se učením prohlubují. Protože možné změny na vstupech systému nelze exaktně specifikovat ani předpokládat, inteligentní systém si klade za cíl napodobovat schéma lidského chování a při vyhodnocování podnětů uplatňovat zejména tvořivost. Na inteligentní systém je možné nahlížet také z pohledu umělé inteligence. Vzhledem k tomu, že ta v sobě zahrnuje množství vědních oborů, budou se také lišit různé inteligentního chování. Obecně lze říci, že systém je inteligentní, pokud by bylo jeho chování označeno jako inteligentní v případě, že se takto bude chovat člověk [132].

V návaznosti na výše uvedenou definici inteligentního systému lze uvést, že inteligentní výrobní systém je takový systém, který vykonává výrobní funkce stejně, jako by je vykonával lidský operátor. Inteligentní výrobní systémy by měly mít tyto vlastnosti [73]):

- Schopnost využití distribuovaného přístupu, ve kterém mohou být komplexní problémy rozdělené na množství menších problémů a dále distribuovány k vyřešení na jednotlivé inteligentní části systému (řídící jednotky).
- Každá řídící jednotka je autonomní a sleduje své vlastní cíle s využitím svých znalostí a schopností. Každá jednotka tedy zapouzdřuje inteligentní funkce. Žádná z nich nedisponuje globálním pohledem na systém.
- Globální řídící rozhodnutí (rozvrhování, diagnostika) jsou realizovány kooperací více řídících jednotek. Jednotky musí kooperovat, aby dosáhly realizace výroby.
- Některé řídící jednotky jsou připojené k automatizovanému fyzickému stroji.
- Řízení jednotek by mělo vykazovat důležité vlastnosti, jako jsou rekonfigurabilita, robustnost, modularita, učení a znovupoužitelnost.

- Systém by měl poskytovat dostatečnou rychlost reakce na různé formy selhání tak, aby mohly tyto být řešeny v reálném čase.

Vlastnosti, které popisují chování inteligentních výrobních systémů, reflektují jejich hlavní myšlenku, totiž *jak zvýšit výkonnost výrobních systémů a jejich schopnost adaptace na rychle se měnící požadavky (zákazníků, produktivity výroby, zkrácení Time-to-Market) vstupující do výrobního procesu (viz kapitola 2.2.3).*

Návrh IMS by měl počítat také s neurčitostmi a nekompletními informacemi, které jsou vždy v reálném světě přítomny. Heterarchická struktura inteligentního výrobního systému může být inspirována např. fungováním živých organismů, či mechanismem fungování lidské společnosti. Inspiraci v přírodě je možné nalézt např. při vyhledávání potravy pomocí přístupu založeného na feromonech. Hmyz, zejména kolonie mravenců, fungují na stejném principu. Každý mravenec se řídí pouze jednoduchým algoritmem, avšak kolonie jako celek je schopná řešit a přizpůsobit se široké škále okolností. Např. v kolonii je důležité, aby mravenci našli blízké zdroje potravy a vytyčili krátké trasy propojující hnízdo a zdroje. Kolonie je schopná vybudovat minimální strukturu takovýchto cest. Funguje v tzv. konceptu kolektivní inteligence - ukazuje se, že systém začne takovou kolektivní inteligenci vykazovat při řešení problému většího rozsahu, tedy takového, který zjevně přesahuje rámec individuální inteligence. Žádný mravenec nedrží *rozvrh* dodávky potravy, neexistuje ani žádný centrální plánovač. Řešení problému zde není bráno jako konečný výsledek, ale jako kontinuální proces. Ve výrobě by se tento přístup projevil větší stabilitou vůči poruchám, lepší adaptabilitou a flexibilitou vůči změnám či efektivním využíváním dostupných zdrojů [115].

6.3 IMS založené na agentním přístupu

Výše uvedené požadavky kladené na inteligentní výrobní systémy jsou v souladu s možnostmi, které v heterarchické struktuře nabízí multiagentní systémy. Právě heterarchická architektura umožňuje plně decentralizovat rozhodovací a plánovací kompetence. Teorie multiagentních systémů svou koncepcí spadá do několika vědních disciplin, mezi jinými také do oblasti výzkumu umělé inteligence. Zaměřuje se zejména na systémy složené z více na sobě nezávislých prvků umístěných v prostředí. Tyto prvky disponují distribuovanou inteligencí a jsou ve vztahu kooperativní heterarchie. Agentní koncept je rovněž v souladu s definicí komponenty Industry 4.0 definované v kapitole 4.2.

Na počátku vzniku a vývoje agentních systémů stojí distribuovaná umělá inteligence, v níž se autonomní jednotky schopné řešit určité problémy nazývají *aktéři*. Vznik tohoto oboru lze datovat do první poloviny 90. let, jeho kořeny však lze sledovat ještě i mnohem hlouběji, s trochou nadsázky dokonce v samotné myšlence objektově orientovaného programování, kdy jednotlivé objekty mohou být označeny jako *agenty*. Princip autonomního agenta (tzv. reaktivního agenta) zpracoval Rodney Brooks, pracovník AI laboratoří MIT. Principu inteligentních agentů (Intelligent Agents) se věnoval M. J. Wooldridge [74].

Agent jako autonomní výpočetní systém plní předdefinované cíle a svým chováním zvyšuje výkonnost systémů v závislosti na svých zdrojích, schopnostech, vnímání a možnostech komunikace. Pro úspěšné splnění cílů agenti využívají principy kooperace, vzájemné koordinace a vyjednávací mechanismy. Autonomní agentní entity mohou být různých druhů. Nejjednodušším je *reaktivní* agent, který dokáže vnímat okolní prostředí a reagovat na případné změny. Pokročilejší jsou *proaktivní* či *sociální* agenti, kteří dokáží převzít iniciativu a cíleně upravit své reakce, potažmo dokonce realizovat interakci pomocí vhodného komunikačního jazyka, ontologie a interakčních protokolů.

Chování agentů je určeno množstvím aspektů - od jejich vnitřní struktury samotné, až po architekturu prostředí, ve kterém jsou agenti nasazeni. Z pohledu složitosti a míry inteligence (racionality) můžeme agenty dělit na [57]:

- **Reaktivní agent** (alternativní přístup - nemysli, konej). Jedná se o typ agenta s nejjednodušší vnitřní architekturou. Vyznačuje se tím, že nemá kapacitu modelovat obraz svého prostředí, tzv. nemá modul vnitřní reprezentace. Následkem toho není schopen předvídat, jakou akci bude nutné vykonat. Agent pouze bezprostředně reaguje na podněty vycházející z prostředí, ve kterém operuje. Součástí reaktivního agenta jsou kompetenční moduly, většinou ve formě funkce, či procedury obsahující invokační podmínku a důsledkovou část (specifikace chování agenta po přijetí příslušného stimulu z prostředí). Reaktivní agenty dělíme na ty s pamětí a bez paměti. Agent s pamětí si pamatuje stav, do kterého se v minulosti vlivem působení prostředí dostal, a pokud se stejná situace opakuje, zachová se stejně. Naopak agent bez paměti nemá žádné vnitřní stavy, proto vždy vykoná první akci, která je v dané situaci možná.
- **Deliberativní agent** (klasický přístup - nejprve myslí, potom konej). Agent udržuje obraz okolního prostředí, dokáže cíleně vytvářet plán svých reakcí, přičemž udržuje reprezentaci svých vnitřních vztahů. Oproti reaktivnímu agentovi disponuje modulem vnitřní reprezentace, je schopen deduktivního myšlení, s jehož pomocí dokáže stanovit posloupnost akcí vedoucích k cíli. Vlastnosti předurčují tohoto agenta do prostředí, kde je deterministicky specifikované řešení daného problému.

Podmnožinou deliberativních agentů jsou sociální agenti. Ti rozšiřují popisované vlastnosti o možnost kooperace s okolními agenty, tedy komunikace za účelem optimalizace svých činností ve prospěch okolních agentů. Díky tomu mohou sociální agenti dosáhnout společného cíle systému.
- **Hybridní agent** (alternativní přístup - myslí a konej nezávisle). Tento typ vznikl kombinací vlastností reaktivního a deliberativního agenta. Hybridní agent operativně přizpůsobuje své vlastnosti a dokáže se chovat jako jednoduchý reaktivní agent, avšak pokud to situace vyžaduje, umí si budovat také vlastní plán.

Komunikace agentů

Jednotlivé entity multiagentních struktur mají v rámci výrobního procesu omezené informace a kompetence, proto je ke splnění společných cílů nutné, aby vzájemně sdílely informace o výsledcích svého úsilí. Je také nevyhnutelné, aby každý z agentů měl možnost komunikovat se zbytkem agentního světa, tzn. aby byli jednotliví agenti propojeni. Kooperativní uspořádání, v němž se agenti často nacházejí, a s ním spojené dosažení vyššího stupně racionality s sebou přináší mnoho implementačních nároků. Interakční mechanismy musí implementovat specifické vlastnosti rolí, vazeb a vzájemných závislostí mezi agenty. Tato implementace vyžaduje standardizované komunikační protokoly a procesy [57].

V prostředích s různými platformami a heterogenními agenty je kladen velký důraz na standardizovaný rámec komunikace s cílem zaručit syntaktickou interoperabilitu mezi různými agentními systémy. Standardy *FIPA-ACL* či *KQML* popisují manipulační jazyky založené na komunikačních aktech. To zjednodušeně znamená, že nedefinují formu samotného významu přenášené zprávy, ale pouze od koho, komu a kdy byla zpráva doručena a v jaké formě odesílatel očekává odpověď. Implementace těchto jazyků definují dvě úrovně. Tou první je úroveň rozlišení řečového aktu, druhou je pak úroveň samotného komunikovaného obsahu. Obsah zprávy může být v takovém jazyce, který je adekvátní pro vyjádření obsahu a rozumějí mu oba agenti [84]. Komunikaci zastřešuje přenosová vrstva založená většinou na protokolu TCP.

Pro ilustraci KQML lze uvést příklad komunikačního aktu s žádostí agenta B o informace od agenta A (viz obr. 6.2). Agent B může být v kontextu výrobního podniku agent, který prostřednictvím vytvořeného kanálu periodicky vyjednává svou výrobu s ostatními agenty. Agent A v pozici agenta výrobního zdroje odpovídá na požadavek agenta B poskytnutím reference na svou rezervační frontu, na základě které plánuje svou činnost.

Každou zprávu tvoří dvojice parametrů a hodnot. Pro jednotlivé typy řečového aktu jsou definovány tzv. *performativy* (na obr. 6.2 tučně zvýrazněné), které určují typ obsahu - tvrzení, dotaz, příkaz, apod. Zároveň popisují také formu, jakou by měla být doručena odpověď. KQML implementačně podporuje služby, umožňující agentům se navzájem lokalizovat v distribuovaném prostředí pomocí nepovinných vlastností ve zprávách (např. jedinečných identifikátorů, či jmen). Mezi výhody komunikačních jazyků KQML patří:

- nezávislost komunikačního jazyka na jazyku obsahu zprávy,
- oddělení obsahové vrstvy od vrstvy komunikačního aktu a
- deklarativní forma.

Samotnou komunikaci agentů můžeme rozdělit z pohledu centralizace hledání požadovaných agentů na *přímou* a *nepřímou*. Pokud spolu agenti nekomunikují přímo,

```

(subscribe
  :sender B
  :receiver A
  :timestamp 1113340454
  :reply-with query_1
  :language KQML
  :ontology Manufacturing system
  :content (ask if
    :sender B
    :receiver A
    :in-reply-to query_1
    :language Prolog
    :ontology Manufacturing_system
    :content reservation request(time)
      state(in progress)
  )
)

(tell
  :sender A
  :receiver B
  :timestamp 1113341454
  :reply-with query_1
  :language Prolog
  :ontology Manufacturing system
  :content reservation queue(array[X]);
    state(in operation) ;
)

```

Obrázek 6.2: Příklad dotazu a odpovědi v jazyku KQML [84]

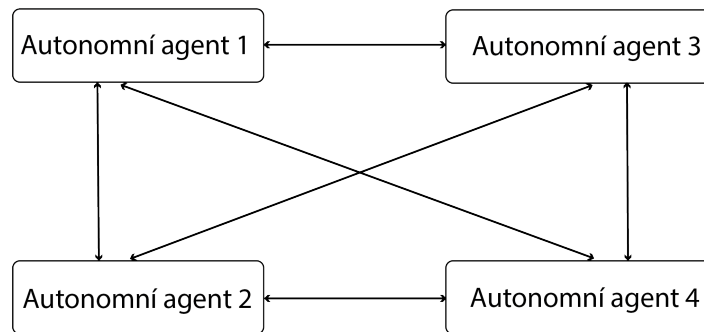
ale prostřednictvím jiného (zprostředkovatelského) agenta (viz obr. 6.3), který vyřizuje jejich požadavky, hovoříme o nepřímé komunikaci. Takové MAS se nazývají *federované* a tvoří předstupu decentralizovaných MAS. Jednotliví agenti nemusí znát adresy a poskytované služby ostatních agentů, stačí jim znát pouze adresu mediátorského agenta. Ten má k dispozici databázi údajů, adres a schopností, přičemž se stará o pravidelnou aktualizaci těchto dat. V systému vystupuje jako komunikační most, což umožňuje implementační volnost, a zjednodušuje vztahy mezi ostatními agenty. Selhání zprostředkovatelských agentů však přirozeně ohrožuje stabilitu celého systému. Je proto nutné dodržet jejich funkční redundanci. Mezi výhody nepřímé komunikace patří redukce zátěže komunikačních rozhraní v systému. Praktickou implementací zprostředkovatelského agenta je např. zařízení se spuštěným Discovery serverem OPC UA popsaným v kapitole 4.6.2.



Obrázek 6.3: Architektura federovaného multiagentního systému

Decentralizované multiagentní systémy

Decentralizované (autonomní) MAS neobsahují centrálního agenta, ani žádné zprostředkovatelské agenty. Jednotliví agenti vzájemně komunikují pouze na základě svých sociálních poznatků. Každý agent je tedy autonomní entitou, využívající přímou komunikaci napříč systémem. Vlastnosti decentralizovaného MAS určuje povaha interakcí mezi agenty, které mohou být statické a dynamické. Statické interakce charakterizuje vyšší míra centralizace, dynamické pak vyšší robustnost vyplývající z využívání negociačních protokolů ke koordinaci aktivit. Z principu vysoké míry autonomie agentů vyplývají také vyšší nároky na komunikační prostředky (pro systém s N agenty musí existovat až $N \cdot (N - 1)/2$ komunikačních kanálů typu *bod-bod*). Je nutné počítat s vysokou režii komunikace, zejména kvůli nutnosti vyhledávání agentů (jednotliví agenti nedisponují informacemi o ostatních entitách v systému). Potenciálním rizikem této koncepce uspořádání agentů je možnost vzniku nepředvídatelného chování systému (viz následující kapitola).



Obrázek 6.4: Architektura decentralizovaného multiagentního systému

6.4 Potenciál a využitelnost MAS v průmyslovém prostředí

Koncepce multiagentních systémů je implementačně relativně mladá, a proto jsou struktury založené na agentním principu často spojovány s nedostatečnou, či zcela chybějící metodikou vývoje, z čehož plyne také vlažné přijetí ze strany průmyslu. V současnosti se nejčastěji využívá centralizované řízení výroby spoléhající se na MES (MOM) systémy, do kterých je potřebné agregovat veškeré relevantní údaje. Takový přenos je strukturován vertikálně a podléhá hierarchickým principům. Paradigma multiagentních systémů však nahrazuje centralizované řízení distribuovaným, při kterém vedou interakce mezi jednotlivými agenty ke vzniku *inteligentního* globálního chování. Interakce jsou integrovány jak horizontálně, tak i vertikálně, což umožňuje dosažení vysokého stupně autonomie a spolupráce bez potřeby rigidní struktury a následně poskytuje modularitu, flexibilitu, robustnost a agilitu chování.

Multiagentní přístup má vysoký potenciál přidané hodnoty zejména v komplexně propojených systémech s vysokou mírou složitosti, kde tradiční centralizovaný přístup (monoliticky či hierarchicky řízený) je nejen neefektivní, ale častokrát z důvodu omezení

množství zdrojů centrálního agenta nereálný. Z tohoto pohledu je snaha o dekompozici do více funkčních jednotek (agentů) zcela přirozená. Ve výrobní doméně umožňuje takový přístup efektivnější využití výpočetního výkonu a tedy zkrácení času splnění úlohy a zvýšení kvality výsledného řešení [132].

Na teoretické úrovni existují nejméně tři kritéria, po jejichž splnění lze použít agentně orientovanou architekturu v průmyslovém prostředí:

- systém je přirozeně distribuován (např. autonomní subjekty, případně geograficky rozdělené),
- mezi jednotlivými prvky systému existuje potřeba flexibilní interakce (tedy nesmí existovat apriorní přiřazení úloh jednotlivým aktérům, tj. žádné pevně stanovené procesy),
- technologie je zasazena do dynamického prostředí s proměnnými vlastnostmi, což vyžaduje časté a bezprostřední intervence systému v reálném čase, např. pomocí protokolů vyjednávání/spolupráce.

Výrobní systémy, které jsou přirozeně velmi dynamické při vytváření vztahů nejen v rámci systému, ale také se zákazníky či dodavateli mohou nalézt účelná řešení řízení a rozvrhování právě s využitím MAS. Vhodným nasazením MAS je např. řízení v podniku s více závody, který vyrábí produkty na objednávku (strategie make-to-order), přičemž složitost výrobku vyžaduje kooperaci různých oddělení (splňuje všechna tři uvedená kritéria). Naopak, pro výrobní podnik tvořený jediným závodem s malým konsolidovaným počtem dodavatelů, který se vyznačuje poměrně stabilní výrobou několika typů jednoduchých výrobků a využívá strategii make-to-stock může být uspokojivé offline centralizované plánování a rozvrhování s využitím MAS není účelné (nesplňuje ani jedno z uvedených kritérií).

Decentralizovaný přístup k řízení výroby přináší samozřejmě i nevýhody. Jedním z jeho hlavních nedostatků, který brání plnému využití v průmyslovém prostředí, je neterministický charakter a nepředvídatelnost výsledku rozhodovacích procesů. Predikce je prakticky nemožná z toho důvodu, že budoucí stavy jsou vždy výsledkem vzájemné interakce agentů. Ta může samozřejmě vést k nežádoucímu, či dokonce nebezpečnému chování celého systému. Tento jev se nazývá *emergence* a při vzniku takového emergentního chování se dá říci, že neplatí princip superpozice. To znamená, že výsledné vlastnosti celku nejsou součtem vlastností jeho hierarchicky nižších částí. Při kooperaci autonomních agentů v heterarchické struktuře mohou vznikat vlastnosti, které nejsou definované ani v jednom z nich. Odpovědí na toto nebezpečí je rozšíření agentního přístupu o využití *holonického konceptu*, ve kterém jsou některé entity částmi větších celků a dodržují principy hierarchie. Každý holon může být součástí jiných holonů a společně tvoří *holonický systém*. Vzniklá architektura se přibližuje výše popsané upravené hierarchické struktuře [123]. Pro zvýšení potenciálu využití MAS v organizaci výroby je nutné zaměřit úsilí na tyto oblasti [97]:

- Mechanizmy a protokoly vyjednávání - cílem výzkumu jsou sofistikovanější vyjednávací mechanismy a protokoly [55].
- Integrace agentních a tradičních přístupů - výzkumné cesty vycházejí z kombinací vyžadujících offline plánování společně s reaktivním (online) plánováním.
- Teoretické zkoumání metodiky - konsolidace současných výsledků výzkumu a usnadnění implementace reálných průmyslových aplikací s prioritním zaměřením na referenční rámce semi-heterarchické architektury.
- Hodnocení a porovnávání výkonnosti konkrétních implementací MAS - neexistuje žádná standardní metodika založená na společném porovnání výkonnosti s dostatečnou mírou komplexity.

6.5 Související technologie přístupu Průmyslu 4.0

Využití výše popsaných architektur inteligentních výrobních systémů přináší násobně vyšší nároky na konektivitu a autonomii jednotlivých agentů. Mezi relevantní technologie využívané v konceptu Průmyslu 4.0 pro tyto architektury můžeme zařadit *Průmyslový Internet věcí*, *Big-Data*, *RFID identifikaci*, *Cloud computing*, *Architekturu orientovanou na služby* a konečně *Industry 4.0 komponentu* uvedenou v kapitole 4.2.

Sběr dat v reálném čase umožní technologie Průmyslového Internetu věcí. Na velké objemy takto získaných dat se budou vztahovat techniky Big-Data. Vzhledem k dynamické povaze inteligentního výrobního systému a způsobu jeho práce je vzájemná identifikace jednotlivých fyzických entit klíčová. Vzájemná interakce agentů ve výrobním systému často začíná ve chvíli, kdy jsou fyzické entity, které tyto agenti zastupují, k sobě přiblíženy - např. vznikající produkt je založen do stroje (agent produktu a agent stroje spolu potřebují začít komunikovat, přestože jsou oba vzdáleni klidně stovky kilometrů od fyzických entit). Identifikace fyzických entit tak může být řešena např. pomocí RFID technologie.

Jednou z integrálních součástí modelování multiagentních struktur v průmyslu je architektura orientovaná na služby (SOA). Tato koncepce vznikla s cílem ulehčit implementaci distribuovaných systémů. Definiuje přístup výrobních řídicích algoritmů k dynamické rekonfiguraci založené na standardu IEC 61499 s ohledem na rozšiřitelnost a interoperabilitu. Při správném návrhu sjednocuje SOA technologii na všech podnikových informačních úrovních, od senzorů a akčních členů až po úroveň ERP. V multiagentním světě jsou koncepty SOA agenty využity pro zprostředkování a konzumaci služeb mezi agenty [120].

Agentní koncepty v oblasti automatizace reprezentují kyber-fyzické výrobní systémy popsané v kapitole 3.2.2. Jak již bylo uvedeno, CPPS, jejichž základním principem je propojení vestavěného systému, tedy fyzické části reprezentované prvky procesní automatizace a kybernetické části, zastupující softwarovou výbavu entity (agenta),

jsou inovativním konceptem řízení výroby. Využití tohoto konceptu spolu s agentním přístupem představuje cestu změny organizace řízení výroby z tradiční (statické, hierarchické, uzavřené) na dynamickou (otevřenou, samoorganizující) síť I4.0 komponent. Základním rysem každé I4.0 komponenty je přítomnost komunikační obálky (AAS). Zatímco pasivní část může sloužit k identifikaci vlastností komponenty, konfiguraci, či poskytování dat submodelů (viz kapitola 4.3), aktivní část může implementovat negociační protokoly pro koordinaci aktivit agenta. Výhodou tohoto přístupu je již možnost automatické rekonfigurace v případech, kdy dojde k jakékoli změně podmínek výroby. V případě vzniku změny, ať negativní (např. výpadek některého z agentů či snížení jeho výkonu) či pozitivní (přidání zdrojů či navýšení výkonu agenta), iniciují negociační mechanismy proceduru vyjednávání s jinými stroji či výrobky [96].

6.6 Referenční architektura CPPS

Při návrhu architektur inteligentních výrobních systémů založených na multiagentním nebo holonickém přístupu je nutné formalizovat základní typy agentů, které reprezentují entity ve výrobní oblasti. Jednou z takových architektur může být architektura *PROSA*, která definuje čtyři hlavní holony, z jejichž počátečních jmen je odvozen její název. Jsou to *Product* (výrobek), *Resource* (zdroj), *Order* (objednávka) a *Staff* (personál). Další architekturou je *ADACOR*, který vychází z konceptu *PROSA* a definuje supervizní holon a jeho roli koordinátora rozvrhování a plánování operací [132].

Ve většině současných agentních architektur jsou na makroúrovni specifikovány dva nejdůležitější typy agentů - *agent zastupující fyzickou entitu* a *agent zastupující organizační entitu*. Zatímco první zmiňovaný připojuje do virtuálního světa např. výrobní zařízení (měřicí zařízení, aktuátor, stroj, ale také např. produkt), druhý poskytuje čistě softwarové služby (diagnostické, statistické, či např. službu transformace objednávek zařazených do výroby na výrobní agenty) [115]. V odborné literatuře je uvedeno více různých konceptů agentních CPPS s různými typy agentů a různou filozofií [121, 114, 92, 89]. V rámci této kapitoly je popsán přístup založený na studii [121], specifikující dva základní typy entit:

- *Resource agent* zastupující fyzické zařízení.
- *Product agent* zastupující vyráběný kus.

Na mikroúrovni konkrétního agenta se topologie zjednodušeně skládá z komunikačního modulu, znalostní báze a argumentačního modulu. Specifikace komunikačního modulu by měla definovat komunikační kanál a příslušnou ontologii (viz kapitola 6.3). Vzhledem k tomu, že agent je softwarová aplikace, měla by jasně odkazovat na architekturu ISO/OSI, která specifikuje fyzické a softwarové vlastnosti spojení. Báze znalostí implementuje know-how agenta v určité oblasti. V kontextu výrobního podniku zahrnuje informace v poli působnosti konkrétního agenta - technologický postup, seznam

podporovaných výrobních operací, pravidla či omezení, apod. Argumentační modul využívá báze znalostí k rozhodování o činnosti agenta, ale stejně tak dokáže obsah báze modifikovat (přidávat pravidla, upravovat parametry apod). Modul by měl rovněž disponovat určitou formou inteligence, a dokázat se učit z předchozích zkušeností. Tato vlastnost argumentačního modulu představuje důležitý aspekt výzkumu v oblasti multiagentních systémů [93].

Resource Agent

Resource agent (RA) představuje inteligentní softwarovou obálku fyzického výrobního zařízení. Agent výrobního zařízení může reprezentovat všechny zdroje napříč celým výrobním procesem - od dopravníků, jeřábů, AGV systémů, obráběcích či CNC strojů, až po lidské zdroje. Agent transparentně reprezentuje a udržuje aktuální seznam služeb, které je schopen poskytnout (on sám či jeho fyzická část). Skládá se z více modulů, které díky vzájemnému propojení plní diagnostickou, řídicí a plánovací funkci. Každý agent výrobního zdroje dokáže kooperovat s ostatními agenty a v případě změn ve výrobě převzít naplánovaný harmonogram na sebe.

Resource Agent společně s fyzickým zařízením tvoří I4.0 komponentu a pro jeho implementaci platí vše, co bylo uvedeno v kapitole 4.2. Inteligence agenta může být implementována jako softwarový modul v řídicím členu zařízení, nebo se může jednat o samostatnou softwarovou (či také hardwarovou) entitu. Agent (komponenta) může reprezentovat pouze jedno zařízení, stejně tak může agent agregovat funkcionalitu více fyzických zařízení, či dokonce celé výrobní linky nebo i většího celku. Mezi RA a fyzickým zařízením neexistuje předepsaný způsob komunikace. Ta může být realizována celou řadou tzv. Fieldbus protokolů, protokolů průmyslového Ethernetu, OPC-UA, nebo může být výrobcem zvoleno čistě proprietární řešení. Komunikační vrstva je vždy spravována odděleně od zbytku aplikační logiky, což přispívá k aplikační a platformní nezávislosti celého řešení [132].

Product Agent

Product agent (PA) představuje inteligentní softwarovou nástavbu nad vyráběným produktem. Takový inteligentní výrobek by měl disponovat zejména prostředky pro proaktivní alokaci výrobních požadavků pomocí komunikačního rozhraní s orientací na resource agenty. Dále by měl být schopen monitorovat svůj stav, stejně tak jako stav výrobních zařízení, případně jiných PA. Na základě těchto informací by měl být schopen plně řídit a optimálně plánovat výrobu produktu. PA dále musí obsahovat procedury pro odstranění své fyzické části a uvolnění výrobního procesu v případě, že některá z výrobních operací vede ke vzniku neopravitelného zmetku. Optikou správy životního cyklu výrobku hraje inteligence výrobku důležitou roli při sběru dat pro zpětnou vazbu k vývoji další generace výrobků či k proaktivní údržbě. Agent inteligentního výrobku tedy nemusí ukončit svůj životní cyklus ihned po výrobě. Agent (I4.0 komponenta)

bývá umístěn v podnikové infrastruktuře (lokální či cloudové), a tam také může dále setrvávat pro další využití v průběhu životního cyklu výrobku, či celé generace výrobků.

Mnoho inteligentních výrobků je při výrobě montováno z dílčích částí, které mohou být opatřeny vlastní inteligencí (agentem). Jednou z vlastností PA je tedy také míra integrace inteligence. Hierarchicky nejvyšší entita v procesu výroby bude vystupovat buď jako inteligentní položka (kdy již není možné rozlišit její dílčí komponenty), nebo jako inteligentní kontejner (kdy nedochází k agregaci inteligence) [132].

Další typy agentů

Kromě zmíněných dvou typů agentů je možné v topologii CPPS uvažovat například o *zásobovacím agentu*, který uspokojuje materiální požadavky zdrojů, dále je v [132] specifikován např. *Logistic Management Agent*, který obsahuje logistické parametry (např. přepravní kapacitu či dopravní časy). Rozhraní mezi agenty a člověkem může být implementováno rovněž jako samostatný agent inteligentní objednávky. Takový agent by spravoval všechny nabízené služby systému a sledoval rozpracovaností zadaných objednávek. Rovněž by vytvářel a inicializoval Product Agenty.

6.7 Shrnutí kapitoly

Zatímco v případě centralizovaně organizovaných architektur je plán výroby vypočten některou z vhodných metod v centrálním členu, v architekturách bez centrálního členu (heterarchických) je při plánování výroby nutné využít odlišné principy. Jednotlivá zařízení organizovaná v heterarchické architektuře jsou modelována jako agenti s různými rolami. Existuje několik typů tzv. referenčních architektur, které popisují různé typy agentů. Dvě hlavní agentní entity - Resource agent a Produkt agent jsou v různých formách zastoupeny ve všech těchto architekturách.

Prostřednictvím distribuovaných algoritmů mohou jednotlivé stroje (Resource agenty) a výrobky (Produkt agenty) přijímat vlastní rozhodnutí o řízení výroby týkající se přidělování zdrojů a koordinace pomocí automatizované formy vyjednávání, a tím sestavit plán výroby. Těmto algoritmům bude věnována následující kapitola.

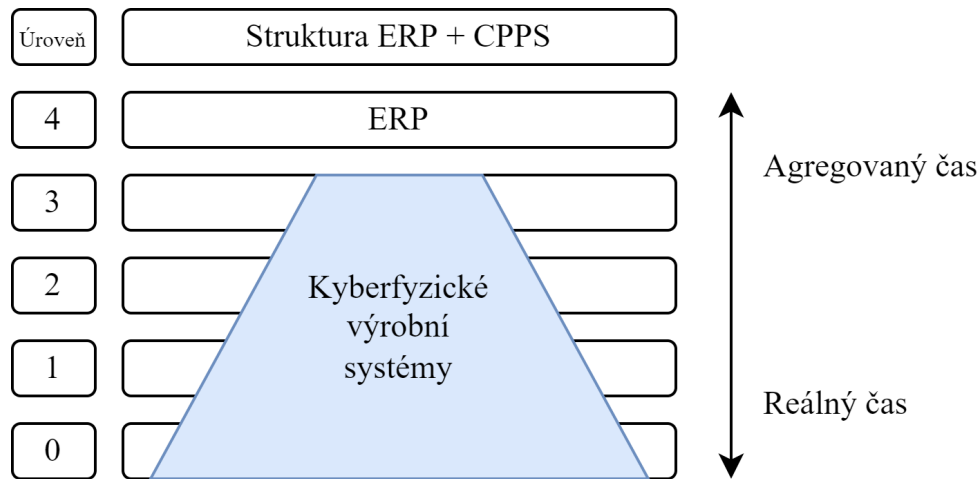
Kapitola 7

Algoritmy pro distribuovanou organizaci výroby

Při analýze a návrhu distribuovaných rozhodovacích algoritmů s akcentem na principy Průmyslu 4.0 je vhodné vycházet z normy ISA 95. Tato norma poskytuje všeobecně uznávanou reprezentaci architektury napříč výrobním podnikem a hierarchicky organizuje jednotlivé úrovně rozhodování (viz kapitola 3.1.1 a obrázek 1.4), sjednocuje terminologii a objektové struktury pro výměnu informací mezi řízením výroby a podnikem.

Cílem modelu rozvrhování v multiagentních CPPS systémech je zabezpečení splnění plánů všech produktových agentů v systému, a zároveň zabezpečení co nejlepší využitelnosti resource agentů. Rozhodování v CPPS mění tradiční způsob přijímání rozhodnutí v oblasti průmyslového řízení. Kapacity, kterými disponuje CPPS architektura, umožňují integraci prvních čtyř úrovní hierarchického modelu do jediné distribuované architektury (viz změna na obrázku 7.1). Model CPPS poskytuje přirozený způsob integrace dispečerského řízení výrobních procesů, dohledových a dalších funkcí spolu se samotným řízením technologického zařízení. Právě z této integrace plynou přímé výhody, zejména již zmíněná zvýšená flexibilita a odolnost vůči neočekávaným událostem [24].

Systémy ERP v novém modelu rozhodování plní funkci propojení s lidskými rozhodovacími záměry na úrovni strategického či střednědobého plánování. ERP systém agreguje a transformuje tyto záměry pro nižší vrstvu CPPS, která zcela autonomně zastřešuje zbytek výrobního procesu a poskytuje možnost simultánní optimalizace rozvrhování výroby při vzniku nečekaných událostí [38]. Vzájemná vertikální koordinace jednotlivých komponent ERP systémů v různých doménách (např. logistické, servisní, dopravní) s CPPS architekturou je však nezbytná k dosažení plné propojenosti v rámci digitální továrny. Jedním z hlavních cílů budoucích výzkumných aktivit zůstává právě řešení problémů integrace vzájemně ne vždy plně kompatibilních entit a předefinování tradičních metod a algoritmů tak, aby naplňovaly distribuovaný rámec [93].



Obrázek 7.1: Struktura úrovní prvků automatizace při použití CPPS architektury, dle [137]

7.1 Model rozvrhování úloh v CPPS

V této kapitole je popsán přístup, který umožňuje decentralizované, dynamické a autonomní plánování. Přístup také umožní hodnotit vhodnost výběru výrobních zdrojů při zpracování úloh. Tyto úlohy je možné modelovat jako *alokační problémy*, při kterých se zásoby a zdroje přiřazují určitému počtu zakázek tak, aby byla dodržena existující omezení. Na rozdíl od klasického statického plánování je množina úloh modifikována příchodem nových událostí - přidáváním, změnou či odstraňováním objednávek. Problém se tak stává dynamickým a jeho řešení zahrnuje nutnost činit rozhodování v reálném čase. Skupina CPPS agentů inteligentních produktů a zdrojů na základě průběžně se měnících požadavků vyjednává cestu fyzického produktu výrobou. Takto koncipovaná decentralizovaná výroba by měla mít možnost nastavit také stupeň priority pro jednotlivé objednávky. *Produkt agent* s vyšší prioritou by měl být upřednostněn v rezervačních frontách *resource agentů* s cílem nejrychlejšího průchodu výrobou. Vhodnými prostředky pro zmíněné interakce jsou strategie kooperace a negociace, jejichž prostřednictvím získává produktový agent informace o dostupnosti, kapacitě a stavu výrobních zařízení[137].

7.1.1 Robustnost rozvrhování

Pro zaručení dostatečné robustnosti plánování v multiagentních systémech je důležité rozpoznat typ anomálie a iniciovat proces pro vypořádání se s případným selháním. Důležité jsou rovněž mechanismy adaptivní obnovy systému do požadovaného stavu [137]:

- **Škálování** - definuje schopnost zvládnout přidávání a odebrání agentů bez nepřiměřené ztráty efektivity systému, či nárůstu složitosti plánování úloh. Pro

dosažení škálovatelnosti je nutné využívat adresovanou komunikaci, případně dělit skupiny agentů do separátních komunikačních domén nebo využívat tzv. *mediátorské agenty*, kteří se specializují na splnění určitého úkolu v agentním systému¹. Z pohledu rozvrhování úloh je v tomto případě měřítkem výkonnosti počet zpracovaných úloh a počet přenesených zpráv, které hodnotí flexibilitu interakční strategie [75].

- **Reakce na změny prostředí** - definuje schopnost zotavení systému v případě, že se změní požadavky na systém ze strany zákazníků (například náhlá změna počtu objednávek). Z tohoto pohledu může být měřítkem počet realizovaných úloh a vícenáklady vzniklé za přizpůsobení systému.
- **Reakce na poruchu agenta** - definuje schopnost systému detekovat selhávajícího agenta a následně delegovat naplánované úlohy na jiné agenty. V systémech s interakčním protokolem typu CNP je tato kompenzace selhání přímo podporovaná (viz kapitola 7.1.2). Řešení může spočívat např. v tzv. stínových agentech, kteří poskytují redundanci v systému. Míru výkonnosti v tomto případě můžeme vyjádřit počtem selhaných úloh [94].

7.1.2 Interakce mezi agenty

Interakce mezi agenty jsou do velké míry ovlivněny typem organizace, ve které agenti operují. Spektrum forem organizace sahá od autonomní formy koordinace vzájemně samostatných agentů, až po organizaci, ve které se částečně či úplně vzdávají své samostatnosti.

Skupiny agentů mohou být agregovány do větších holonických struktur - korporací či aliancí se striktně definovanými dlouhodobými vztahy jednotlivých členů. V rámci těchto skupin se mění forma interakce mezi agenty a snižuje se úroveň jejich samostatnosti.

V tomto textu je rozebírána výlučně forma autonomních agentů, kteří si bez další koordinace pouze vzájemně poskytují služby. Jedná se o formu, která vychází z topologie CPPS popsané dříve. Toto uspořádání agentům poskytuje možnost výměny *splnění úkolu za získání protihodnoty* - určitého druhu užitku. Agenti mají vztah v rovině poskytovatel - zákazník. Zároveň nejsou součástí žádné společné organizace. Jejich interakce je krátkodobá a omezená výlučně na čas potřebný pro vykonání úlohy.

Protokoly založené na spolupráci

Agenti jsou samostatné entity, avšak ve struktuře CPPS mohou spolupracovat. Jedním ze způsobů takové spolupráce je tvorba koalic, do kterých se mohou slučovat agenti

¹Časté je využití mediátorských agentů pro efektivní přeplánování úloh v případě vzniku nečekané události.

s podobnými plány a cíli, kteří disponují společnými parametry a operují ve stejném prostředí [10].

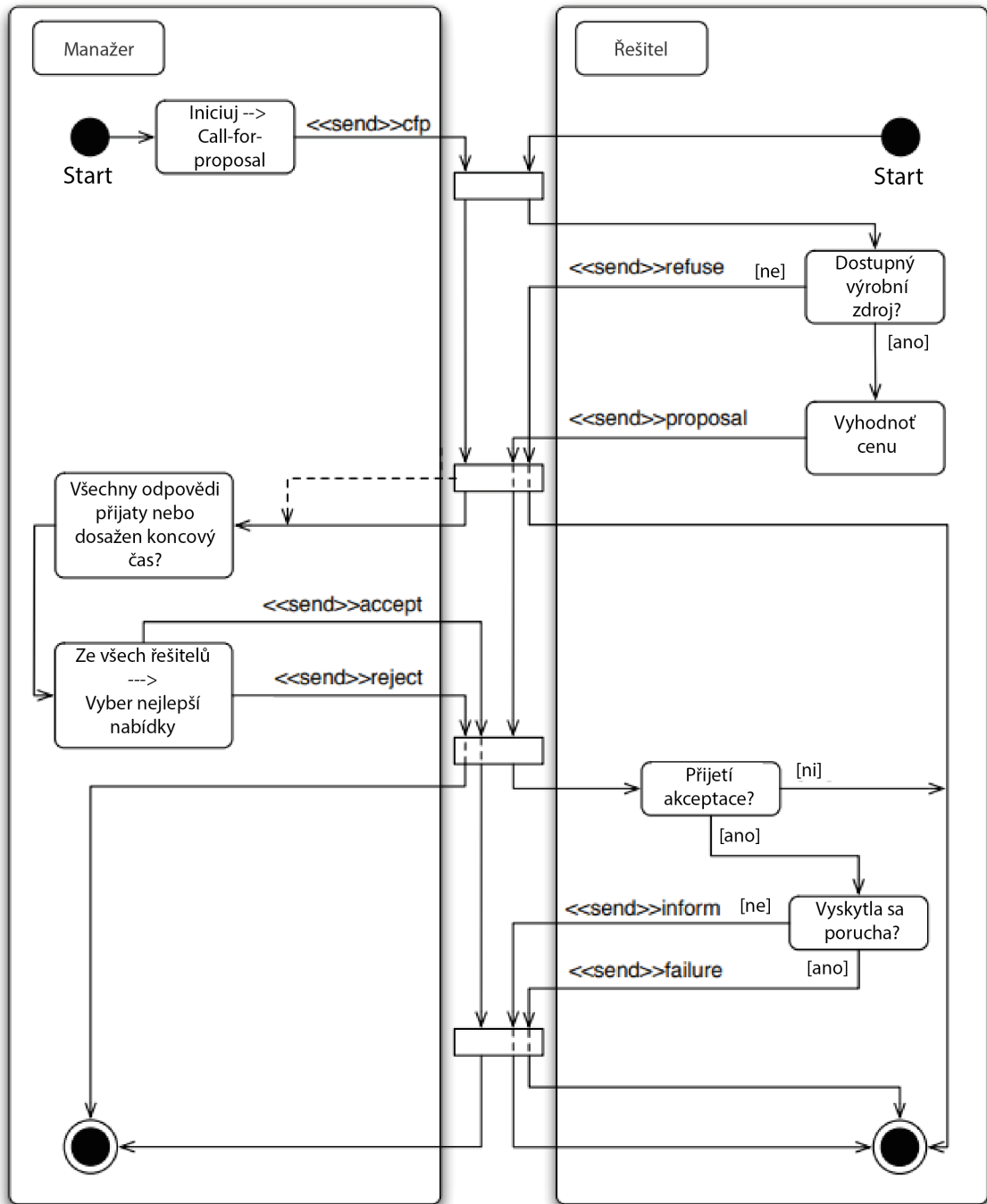
Při spolupráci agentů se uplatňují dva mechanismy. Agenti s různými schopnostmi mohou spolupracovat, aby spojili své síly (kvalitativní motivace), nebo mohou agenti se stejnými schopnostmi spolupracovat za účelem zvýšení kapacity systému (kvantitativní motivace). Spolupráce uvnitř koalice agentů vyžaduje technické komunikační prostředky. Jedním ze široce rozšířených je **Contract Net Protocol** (CNP). V mírně upravené formě je použit k simulacím CPPS v rámci této práce. Níže je demonstrována jeho funkčnost a jsou diskutovány problémy související s jeho používáním.

Síť agentů pro distribuované řešení problémů se obecně skládá ze dvou typů agentů - *řešitelských* a *manažerských*. Manažerští agenti navrhnou a zasílají předem definovanou formou zprávy specifikující požadovanou úlohu. Podle možností komunikační infrastruktury je toto odesílání realizováno buď zprávou typu *Broadcast*, nebo se zpráva posílá pouze vybrané skupině kompetentních agentů. Řešitelský agent vyhodnotí přijaté požadavky a s přihlédnutím ke svým schopnostem posílá *nabídku*. Řešitelský agent může zastávat také aktivní funkci a inzerovat své služby. Manažerský agent si takto vytváří otevřený seznam nabídek, ze kterých vybírá ty nejvhodnější [57].

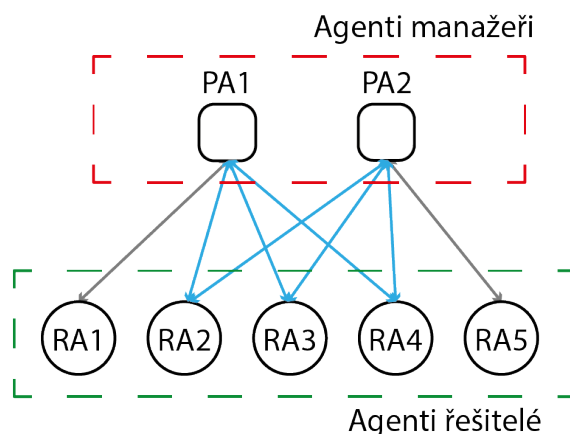
Efektivita protokolu CNP je velmi závislá na množství komunikovaných dat a s rostoucím objemem výrazně klesá. To je způsobeno zejména faktem, že pro odeslání tzv. *call-for-proposal* zprávy (viz obrázek 7.2) se často používá vysílání typu Broadcast. Takové omezení lze řešit využitím agenta *moderátora* pro organizaci komunikace, který zprostředkovává agentu *manažerovi* spojení pouze s kompetentními agenty *řešiteli*.

Protokol CNP dále naráží na své limity zejména při paralelním vykonávání více koordinačních spojení. Základní forma CNP postačuje, pokud v daném čase provádí alokaci agentů *řešitelů* pouze jediný agent *manažer*. Situace se však dramaticky změní, pokud předpokládáme, že existuje několik agentů *řešitelů* (*produktových agentů*) alokujících výrobní operace. Protokol CNP implicitně nepodporuje koordinaci souběžně vykonávaných alokací (na obr. 7.3 jsou modře označená spojení problematická). Protokol rovněž žádným způsobem neurčuje, kolik nabídek by měl řešitelský agent vystavit. Bude-li se řídit pouze racionální úvahou, měl by své zdroje nabízet co největšímu počtu *poptávek*. Takové souběžné vedení transakcí s více iniciátory však přináší stav zvaný *problém nedočkavého uchazeče* (eager bidder problem). Pokud totiž *resource agent* rezervuje poptávané zdroje příliš brzy (např. již během odesílání své nabídky), může se stát, že produktový agent nakonec tuto nabídku nepřijme a *resource agent* zůstane nevyužitý. Pokud se však chce tomuto problému vyhnout a zdroje přiděluje později, např. po získání akceptace nabídky, může nakonec získat více akceptovaných nabídek, než mu umožňují jeho fyzická omezení (typicky objednávku na zpracování dvou či více úloh ve stejném časovém okně) [94].

Uvažujeme-li o prvním uvedeném příkladu, tedy že *resource agent* alokuje své zdroje ihned v čase odeslání nabídky, hovoříme o tzv. aukci s jedním během (*resource agent* mohou v jedné instanci vyjednávání učinit pouze jednu nabídku). Uvedené chování



Obrázek 7.2: UML diagram základního CNP protokolu [94]



Obrázek 7.3: CNP protokol nepodporuje synchronizaci pro souběžnou komunikaci

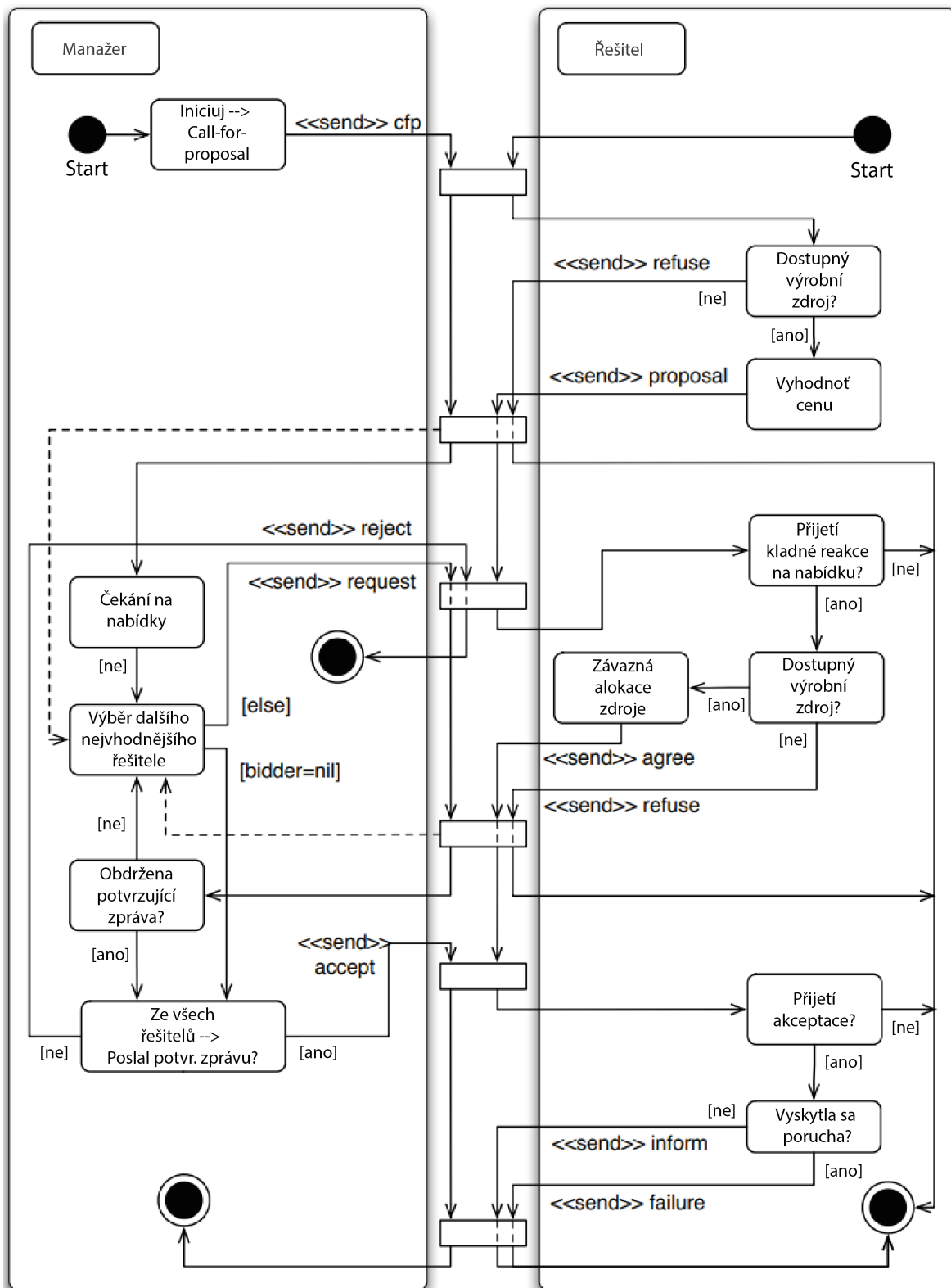
vede k neúplnému přidělení zdrojů, neboť budou existovat iniciátoři (produkt agent), kteří na svou žádost nedostanou žádnou nabídku, zatímco jiní iniciátoři jich dostanou hned několik. Pokud však má produkt agent k přidělení pouze jednu úlohu, nabídky všech resource agentů, které produkt agent přijal, jsou kromě jedné zbytečné. Je zřejmé, že iterativním opakováním procesu alokace, kdy iniciátoři opakovaně poptávají řešení svých úloh, agenti nakonec dosáhnou správného přiřazení všech úloh. Tento způsob řešení problému je však velmi neefektivní a je obtížné jej analyticky hodnotit.

Rozšířený CNP protokol s potvrzováním

Efektivnější přístup k řešení popisovaného problému je založen na přepracování základní podoby protokolu tak, aby co nejvíce odkládal čas závazku. V původním CNP protokolu účastník přijímá svůj závazek již ve fázi odeslání nabídky, zatímco v rozšířeném se závazek přijímá až tehdy, pokud iniciátor (produkt agent) explicitně objedná u řešitelského agenta realizaci úlohy. UML diagram funkčnosti tohoto rozšířeného protokolu CNP je uveden na obrázku 7.4.

Stejně jako v případě jednoduchého CNP protokolu, také zde můžeme hovořit o aukci s jedním během, neboť každý řešitel může vystavit pouze jednu nabídku. Začátek tohoto kooperativního procesu vychází z iniciativy produkt agenta, který v prvním kroku sbírá nabídky všech řešitelských agentů. Do dalšího kroku se přesune buď po posbírání určitého počtu nabídek², nebo po uplynutí definovaného času. Omezení délky časového okna pro sběr nabídek řeší nebezpečí uvíznutí agenta při selhání doručení zprávy, či v případě nedoručení ani jedné nabídky). Následně produkt agent nabídky seřadí sestupně podle svých kritérií výhodnosti a osloví řešitele s nejvýhodnější nabídkou. Oslovený řešitelský agent v tomto bodě ví, že úlohu dostane přidělenou a že stačí pouze souhlasit s jejím vykonáním. Tato skutečnost mu umožňuje bezpečnou alokaci zdrojů a zaslání potvrzení pro iniciátora. V případě, že vykonání úlohy oslovený resource agent odmítne, či pokud uplynul termín pro odpověď, iniciátor osloví s

²Všech, pokud disponuje informacemi o počtu řešitelů, případně omezeného počtu dle vlastní specifikace.



Obrázek 7.4: UML diagram rozšířeného CNP protokolu s potvrzováním [94]

požadavkem o alokaci zdrojů dalšího řešitele v seznamu. Iterace se zastaví, jakmile jeden z účastníků odpoví souhlasem. Všem ostatním agentům je následně odeslána zpráva s odmítnutím návrhu. Komunikační akt je ukončen opět ze strany iniciátora, který posílá finální zprávu s akceptací (v tuto chvíli však nemá důvod nabídku odmítnat, neboť ji odeslal nejvhodnější z dostupných agentů, který je schopen danou úlohu vykonat).

Popsaný rozšířený CNP protokol v ideálním případě vyžaduje, aby každý iniciátor poslal takový počet zpráv, kolik je řešitelských agentů zvýšený o dvě zprávy - žádost o potvrzení a odpověď na ni. V nejhorším případě musí iniciátor kontaktovat všechny řešitele, aby postupně zjistil, že žádný z nich nechce, či nedokáže úlohu vykonat. To vede k navýšení možné komunikační režie a následně také času, zejména v případě čekání na časový limit při selhání některého ze zúčastněných agentů.

Nevýhodou tohoto přístupu je fakt, že iniciátor potřebuje určitou režii na třídění seznamu nabídek řešitelů. Ve výsledku však rozšířený CNP protokol poskytuje robustnější přístup, než jednoduchý CNP [94].

Statistický přístup

Základní myšlenkou statistického přístupu je rozhodnutí riskovat porušení rezervačních závazků, pokud je pravděpodobnost této události z dlouhodobého hlediska nižší, než určitá prahová hodnota. Tento přístup je funguje stejně, jako rezervační systémy leteckých společností, které umožňují na daný let rezervovat více míst, než je kapacita letadla [94].

V kontextu multiagentního systému organizace výroby to znamená určit pro daný práh rizika takový počet nabídek, které může řešitelský agent X odeslat (tedy kolika aukcí se může zúčastnit) nad rámec množství zdrojů, které má k dispozici. Definujme N_{PA} jako počet iniciátorů (produkt agentů), N_{RA} počet řešitelů (resource agentů) a B_X počet nabídek, který resource agent X může poslat ($B_X \leq N_{RA}$). Pro zjednodušení předpokládáme, že všichni agenti postupují podle stejné strategie, a posílají stejný počet nabídek ($B_X = B_A$). Jako τ označíme zadaný práh rizika.

Pravděpodobnost p_c , která určuje, že řešitelský agent si vybere konkrétního iniciátora, jemuž vystaví nabídku, je možné stanovit jako:

$$p_c = \frac{1}{N_{PA}} \cdot B_A \quad (7.1)$$

Pravděpodobnost, že bude konkrétnímu řešitelskému agentovi X potvrzena nabídka v případě, že ji pošle náhodně vybranému iniciátorovi (výpočet uvažuje pravděpodobnost, že nikdo, kromě agenta X si daného iniciátora nevybral je stanovena vztahem 7.2).

$$(1 - p_c)^{N_{RA}-1} \quad (7.2)$$

Pravděpodobnost p_A pro agenta X (viz vztah 7.3), která udává, že bude přijata jediná jeho nabídka, je součinem počtu všech možných permutací a pravděpodobnosti výběru konkrétního iniciátora, umocněná počtem již vykonaných kol. Součin dále pokračuje pravděpodobností akceptace nabídky. Poslední člen součinu vyjadřuje pravděpodobnost výběru konkrétního řešitele iniciátorem.

$$p_A = \sum_{i=0}^{N_{RA}-1} \binom{N_{RA}-1}{i} \cdot p_c^i \cdot (1-p_c)^{N_{RA}-1-i} \cdot \frac{1}{i+1} \quad (7.3)$$

Po získání pravděpodobnosti přijetí nabídky jediným iniciátorem p_A je možné do počítat pravděpodobnost, resp. riziko, že bude přijata více než jedna nabídka při účasti na více aukcích (vztah 7.4). Vystavení více nabídek je z matematického pohledu řetězec pokusů, přičemž všechny mají stejnou pravděpodobnost přijetí. Proto je možné tuto pravděpodobnost vypočítat jako Bernoulliho řetěz:

$$P_X = \sum_{i=2}^{B_X} \binom{N_X}{i} \cdot p_A^i \cdot (1-p_A)^{N_X-i} \quad (7.4)$$

Ve vztahu 7.4 uvažujeme minimálně dva zúčastněné agenty (jinak výpočet pravděpodobnosti nedává smysl). V tomto kroku je možné stanovit maximální počet vystavených nabídek agenta X tak, aby nepřekročil práh rizika. Rozdělení rizika je funkcí proměnných B_X a B_A , přičemž musí platit, že $P_X < \tau$. Analytický pohled na rozdělení rizika odráží skutečnost, že řešitelský agent X zvyšuje riziko porušení rezervačních závazků vůči iniciátorovi tím více, čím více předkládá nabídek. Riziko pro agenta X naopak klesá, pokud ostatní agenti předkládají více nabídek, a tím snižují pravděpodobnost, že bude vybrán právě X . Pokud je B_A pevná, riziko monotónně roste s rostoucím B_X a pokud je B_X pevné, riziko monotónně klesá s rostoucím B_A . Viděno optikou teorie her je určení počtu nabídek B_X pro řešitelské agenty tzv. *Nashovou rovnováhou* [94].

Uvedený přístup ke koordinaci a delegování úloh v multiagentních systémech je přijatelný pouze v oblastech, které umožňují neúplnost a nekompletnost přidělování úloh s uvedenou pravděpodobností prahu rizika.

Kombinace statistického přístupu a CNP s potvrzováním

Důležitou vlastností statistického přístupu je, že se dá kombinovat jak s jednoduchým, tak i rozšířeným CNP protokolem. Kombinace statistického přístupu vstupujícího do fáze odesílání nabídek řešitelských agentů a deterministicky určitého přidělování zdrojů přináší zlepšení efektivity a snížení počtu přenášených zpráv mezi jednotlivými účastníky. Důležité je však podotknout, že taková kombinace nezaručuje úplné přiřazení, neboť statistickým omezením počtu vystavených nabídek může nastat stav, že někteří iniciátoři nedostanou vůbec žádnou nabídku a nemají možnost přidělit své úlohy. Tento

nedostatek se, podobně jako v případě základního CNP protokolu, dá vyřešit zavedením iterativního opakování nabídkové fáze.

Uvedené nedostatky CNP protokolů a postupný rozvoj alternativních přístupů pro souběžné přidělování úloh připravil základ pro protokoly založené na teorii aukcí. Tržní mechanismy (aukce) zobecňují CNP a tvoří důležitou součást výzkumu multiagentních systémů.

Negociační protokoly - tržní mechanismy

Kromě popsaného kooperativního chování existuje v multiagentních systémech také jiný přístup k řešení konfliktních situací. Tím je *aukce*, tedy soutěž o zdroje. Na rozdíl od předchozích kooperačních protokolů jsou v procesu aukce její účastníci v konkurenčním vztahu. Tržní mechanismy založené na ekonomických principech poskytují pokročilejší možnosti, které při využití analytických metod přidělování úloh nejsou zastoupeny. Pomyslný trh určuje cenu výrobního zdroje v čase a prostoru. Na tuto cenu reagují produktoví agenti nabízenou cenou, kterou jsou ochotni za tento výrobní zdroj (jeho čas) zaplatit. Aukční procesy probíhají iterativně, přičemž řešení konverguje ke konečné dohodě (tou je stav, ve kterém agenti nemění své požadavky na zdroje vzhledem k aktuálním cenám). Průběh negociace určují tyto charakteristiky [55]:

- míra kooperace (od plné autonomie agentů až po velké koalice spolupracujících agentů);
- regulace (agenti mohou během procesu vyjednávání podléhat regulačním opatřením);
- množství a typ agentů (proaktivní vs. reaktivní vystupování v procesu vyjednávání);
- výběr modelu vyjednávání - z pohledu času potřebného k dosažení dohody, který je důležitým faktorem z hlediska nákladů na samotnou komunikaci a blokování výpočetního výkonu.

V tomto textu nejsou tržní mechanismy dále rozvíjeny.

7.1.3 Operativní plánování

Rozdíl mezi doménou offline centralizovaného rozvrhování popsanou v rámci kapitoly 5 spočívá v tom, že při dynamickém on-line plánování je výrobní plán aktualizován v reakci na měnící se podmínky a vznik nepředvídaných událostí narušujících výrobu. V případě, že výrobní zařízení vypadne z provozu, ať z důvodu poruchy, či jiné příčiny, musí se operace, které mu byly původně přidělené, přesměrovat na jiné stroje, čímž se kompletně změní nejen plán, ale také podmínky pro plánování dalších

Metody rozvrhování výroby				
Statické		Dynamické		
Deterministické (všechny parametry výroby apriorně známy)	Stochastické (malá míra nejistoty v některých parametrech)	Žádná míra variability (cyklická výroba)	Variabilní čas uvolnění do výroby (flow shop)	Variabilita v celém procesu (job shop)

Tabulka 7.1: Rozdělení metod rozvrhování výroby [62]

operací. Oproti statickému - deterministickému plánování s sebou dynamické plánování přináší nová kritéria rozvrhů - stabilitu, robustnost a nekonzistenci (nervozitu rozvrhu). Tato kritéria slouží k určení výkonnosti algoritmů dynamického plánování. Vzhledem k tomu, že řešení uvedeného problému není triviální, existuje mnoho strategií, politik a metod založených na široké škále experimentálních a praktických přístupů ke změnám harmonogramu [62].

Metody rozvrhování výroby

Deterministické, statické metody (kapitola 5.5) je možno považovat za speciální případ rozvrhování, kdy existuje konečná množina úloh bez nejistot plynoucích z reálné výroby. Podobně je tomu u stochastických statických metod, kde rovněž existuje konečná, dopředu definovaná množina úloh (kapitola 5.6), avšak jsou použitelné také v případech, kdy některé parametry úloh nejsou pevně dány (např. časy zpracování úloh mohou mít statisticky přípustný rozptyl, skutečné časy startu a dokončení úloh pak nebudou odpovídat těm očekávaným). Vykonávání plánu si následně vyžádá minimálně zapojení nějakého pravidla či politiky pro kompenzaci chyby v plánu. Tou může být např. taková možnost úpravy rozvrhu během jeho vykonávání, aby se reagovalo na skutečnou situaci. Manévrovací prostor pro kompenzaci nepředvídaných událostí mohou přinést uměle přidané časy nečinností do původního rozvrhu [4]. Proti tradičním statickým metodám stojí metody dynamické, kdy každá úloha vyžaduje naplánování bezprostředně před zpracováním. Z pohledu variability času uvolnění do výroby a samotného toku zakázky výrobou lze rozlišit tři případy - od cyklicky se opakující výroby stále stejných kusů, až po reentrantní tok bez časové stability či předvídatelné opakovatelnosti [137].

Strategie přeplánování výroby

V dynamickém prostředí, ve kterém nejsou dopředu známy časy příchodů a typy úloh, lze rozlišit dvě strategie řízení výroby - dynamické a prediktivně-reaktivní. Pro dynamické strategie plánování je typické, že nedochází k vytváření výrobních plánů. Namísto toho jsou příchozí úlohy přímo přidělovány výrobním zdrojům výlučně na základě informací dostupných v okamžiku zpracování. Taková schémata využívají dispečerská přiřazovací pravidla, či jiné heuristiky pro určení priorit čekajících úloh (viz kapitola 5.6.1). Výhodou je, že výpočetní náročnost dispečerských pravidel je nízká a je možné je rozšířit také o implementaci pravidel JIT, případně KANBAN, či o jiné

mechanizmy založené na principu tahu (viz kapitola 5.3) [31].

V případě výrobních zdrojů, ve kterých se neuplatňuje časová penalizace (doba přenastavení stroje) před zpracováním zakázky, upřednostňují navrhovaná dispečerská pravidla úlohu podle rozdílu časů nejpozdějšího dokončení a očekávaného času zbývajícího do skutečného dokončení. V případě zdrojů s potřebným časem přenastavení se navrhovaná pravidla zaměřují na dokončení všech čekajících úloh jednoho typu před počátkem zpracování úloh jiného typu. Všechna zkoumaná pravidla udržují stroj v provozu v případě, že na zpracování ještě čekají nějaké úlohy (tzn. stroj nemůže ignorovat čekající úlohy).

Prediktivně-reaktivní plánování se obecně skládá ze dvou kroků. Nejprve se vytváří počáteční plán výroby. Ten se v kontrolní fázi porovnává s průběhem skutečné výroby. Pokud jejich rozdíl překročí stanovenou hranici, následuje druhý krok, ve kterém dochází k aktualizaci plánu v reakci na poruchu, či jinou událost s cílem minimalizovat její vliv na efektivitu systému. Prediktivně-reaktivní doména plánování zahrnuje tři typy strategií změny plánování: *periodické*, *řízené událostmi* a *hybridní*. Změnu plánu mohou iniciovat například následující události:

- porucha výrobních strojů,
- příchod urgentní (prioritní) objednávky,
- zrušení objednávky,
- změna času dokončení úlohy,
- nereálný předpoklad času trvání výroby,
- problémy s kvalitou či nedostupnost materiálu.

Rozsáhlé výrobní systémy, ve kterých vzniká v rychlém sledu mnoho událostí, na základě kterých je nutné upravit rozvrh, se mohou dostat do stavu permanentní úpravy rozvrhu. Vzhledem k tomu, že úprava rozvrhu je výpočetně náročná, jsou v tomto stavu kladeny na výkon systému nadměrné požadavky.

Pro eliminaci nepřiměřeně častých změn rozvrhu (a následně jeho snížené stability), ale zároveň pro zachování včasné reakce na závažné změny ve výrobě, vznikly hybridní metody. Ty představují spojení strategií událostí a periodického schématu přeplánování. Jsou také označovány za metody s *klouzavým časovým horizontem*.

Hybridní přístup s klouzavým horizontem redukuje nepraktický požadavek globální optimalizace na strategii lokálního optimálního rozvrhování (platného jen pro nejbližší vykonávané časové okno). Snížením komplexnosti problému je možno jej řešit i v reálném čase, a proto jsou tyto metody vhodné pro dynamické plánování. Hlavním principem plánování v klouzavém horizontu je výběr několika úloh čekajících na zpracování tak, aby se naplnilo nejbližší časové okno. Okno úloh je vytvářeno pomocí

Strategie přeplánování výroby			
Dynamické (nevytváří rozvrh)	Prediktivně-reaktivní (vytváří odhad a aktualizují) Politika iniciace přeplánování		
Přiřazovací pravidla JIT paradigme	Periodická	Vyvolaná událostí	Hybridní

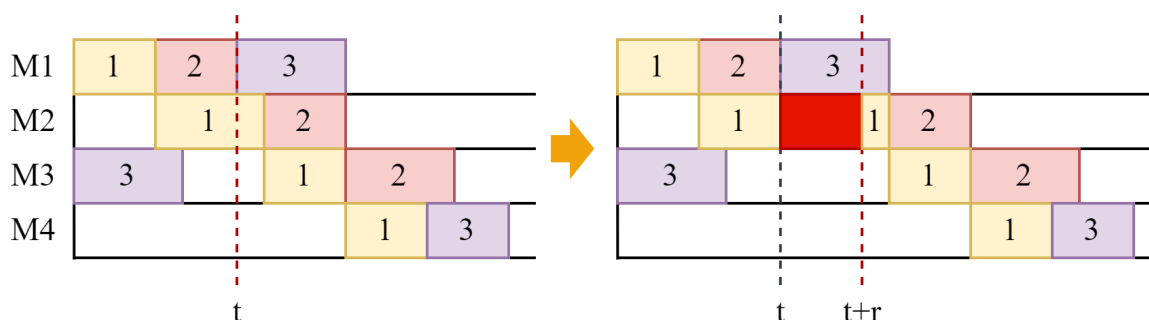
Tabulka 7.2: Strategie přeplánování výroby [62]

výběrového pravidla. V úvahu se berou časy příchodu úlohy a rozdíl mezi plánovaným časem ukončení a termínem dokončení úlohy. Přidáním váhovacích koeficientů pro tyto parametry je možné určit prioritu úlohy. Klouzání okna úloh je realizováno odstraněním dokončených úloh a přidáním nových. Postup se následně opakuje, dokud se nedokončí všechny operace ve všech úlohách. Čím více úloh se nachází v pomyslném okně, tím blíže je výsledek globálnímu optimu, avšak výpočetní čas je delší [124]. Tabulka 7.2 shrnuje strategie přeplánování výroby v kontextu operativního plánování [4, 137].

Metody přeplánování výroby

Metody pro tvorbu výrobního rozvrhu zahrnují širokou oblast algoritmů. Práce [106] prezentuje zástupce algoritmů pro generování robustních rozvrhů založených na genetickém algoritmu, který minimalizuje očekávané zpoždění a očekávanou délku trvání. Rozlišujeme následující metody: *změna posunem doprava*, *částečná změna rozvrhu* a *úplná změna rozvrhu*.

Změna harmonogramu s posunem doprava odkládá každou zpozdující se operaci o čas potřebný k tomu, aby byl rozvrh realizovatelný a splňoval daná precedenční omezení. V diagramu na obr. 7.5 je uveden příklad selhání stroje M2 v čase t během zpracování úlohy 1 (M2), přičemž čas opravy vyžaduje r časových jednotek. Čas opětovného spuštění úlohy 1 (M2) se posune z t na $t+r$, dokončení se příslušně zpozdí. Zaostávající úlohy na strojích M2, M3 a M4 se rovněž zpozdí o r časových jednotek.



Obrázek 7.5: Posunu doprava při přeplánování rozvrhu (dle [62])

Částečné přeplánování přeplánuje jen operace, které jsou přímo či nepřímo ovlivněny poruchou. Tato metoda zachovává původní harmonogram v největší možné míře. To se pozitivně projevuje na stabilitě rozvrhu. Také proto většina vyvinutých

Metody přeplánování výroby			
Tvorba rozvrhu	Úprava rozvrhu		
Implicitní rozvrhy, robustní rozvrhovací metody	”Right-shift” přeplánování	Částečné přeplánování	Kompletní změna plánu

Tabulka 7.3: Metody přeplánování výrobního rozvrhu [62]

heuristik uplatňuje při přeplánování pouze změnu rozvrhu ovlivněných operací. V [4] je navržená strategie přeplánování využívající myšlenku *match-up*. Ta směřuje ke generování takového změněného plánu, který v jistém okamžiku dosáhne zpět stejného stavu, jako má originální rozvrh. Na ten může následně navázat a vrátit se na původně plánovanou trajektorii výroby. Při poruše stroje se určí bod shody pro každý stroj a přeplánuje se jen ta část rozvrhu, který pokrývá časový interval mezi poruchou a časem *match-up*. Změna plánování s posunem doprava je speciálním případem této obecnější metody.

Úplná změna (regenerace rozvrhu) přeplánuje celou množinu operací, které nebyly zpracovány před časem výskytu poruchy, včetně těch, které nebyly ovlivněny narušením. Hlavní nevýhodou tohoto přístupu je nadměrná výpočetní náročnost. Shrnutí metod přeplánování výroby je uvedeno v tabulce 7.3.

7.2 Návrh algoritmu organizace výroby CPPS

Rozhodování o přidělování úloh v multiagentních systémech představuje komplexní koordinační proces, který vyžaduje inovativní přístupy. Algoritmus popsany v této kapitole, vychází z vlastností dynamické strategie rozvrhování. Z povahy dopředu neznámého, stochastického příchodu úloh nevzniká implicitně rozvrh. Namísto toho dynamická strategie rozvrhování spočívá v cyklickém přeplánování postupně vznikajícího rozvrhu na úrovni jednotlivých *Resource agentů*. Úplně se tak eliminuje centralizace, a v důsledku toho je rozvrhování vlastní výroby plně v kompetenci inteligentního produktu (*Produkt agenta*). Cílem algoritmu je umožnit produktovi najít vhodnou sekvenci resource agentů, čímž alokuje výrobní zdroje pro daný typ úlohy. Je nutné podotknout, že architektura CPPS má předpoklad implementovat pouze dispečerské rozvrhování výroby. Strategické a střednědobé plány výrobního podniku jsou stále plně v kompetenci systémů ERP a specializovaných modulů systémů MES. V kontextu kapitoly 7.1.3 je možné tento algoritmus označit také jako *heuristiku ukládání*. Principem této heuristiky je výběr takového místa (časového slotu) v rezervačních frontách resource agentů, které minimalizuje nákladovou funkci vyjadřující cenu výroby produktu. Formalismus v definici modelů agentů je čerpán z [14].

7.2.1 Model výrobního zdroje (Resource agent)

Úkolem každého Resource agenta je vytvořit a udržovat model schopností a rezervační frontu stroje. Tento model se může s každou interakcí agenta s okolními agenty změnit. Formální popis resource agenta v navržené architektuře CPPS je n -tice:

$$RA = (X, E, T_r, Prp_p, Prp_{np}, x_0, R_T)$$

$X = x_0, \dots, x_n$ je množina všech fyzických stavů, ve kterých se může RA nacházet.

$E = e_0, \dots, e_n$ je množina všech logistických a výrobních operací, které mohou nastat.

$T_r : X \times E \Rightarrow X$ je přechodová funkce RA.

$Prp_p : X \Rightarrow P_p$ přiřazuje fyzickým stavům RA vlastnost měnit kompozici výrobku, tj. například fyzikální či chemickou změnu, montáž části, atd.

$Prp_{np} : X \Rightarrow P_{np}$ přiřazuje fyzickým stavům RA vlastnost, která neumožňuje měnit kompozici výrobku, tj. přesun v sekvenci výrobních zdrojů, natočení výrobku, atd.

x_0 je počáteční stav výrobního zdroje.

$R_t : E \Rightarrow Sc_e$ vyjadřuje rezervační frontu RA, ve které jsou jednotlivým událostem přiřazeny plány vykonání. Množina $Sc_e = (t_0, t_1), (t_2, t_3), \dots, (t_{m-1}, t_m)$ udává ohraničené časové sloty, ve kterých může resource agent obsluhovat událost e_n . Pro nezápornost plynutí času platí $t_0 \leq t_1 \leq t_2 \leq \dots \leq t_m$.

V [14] je uveden postup, kdy jednotliví resource agenti spolupracují vždy s resource agenty ve svém sousedství tak, aby společně vytvořili kompletní nabídku pro produktového agenta. Taková týmová práce je založena na speciální znalosti resource agentů - totiž znalosti svého fyzického okolí. Formálně je vždy definována množina stavů ($X_S \in X$), které resource agent sdílí s ostatními resource agenty (fyzicky je takový stav reprezentován místem, ve kterém si jej jednotliví resource agenti předají).

Ve tomto případě je však veškerá iniciativa přesunuta na produktové agenty, kteří musí výrobní úlohu vhodně rozložit na dílčí úlohy (ty optikou standardu ISA 95 odpovídají *segmentům produktů*), které mají vazbu na modely schopností konkrétních resource agentů (ve standardu zavedeny jako *segmenty procesů*). Uvedená skutečnost zaručuje, že resource agenti budou vždy schopni poskytnout kompletní nabídku pro uskutečnění výrobní úlohy, přičemž limitující bude jen kapacita časových slotů.

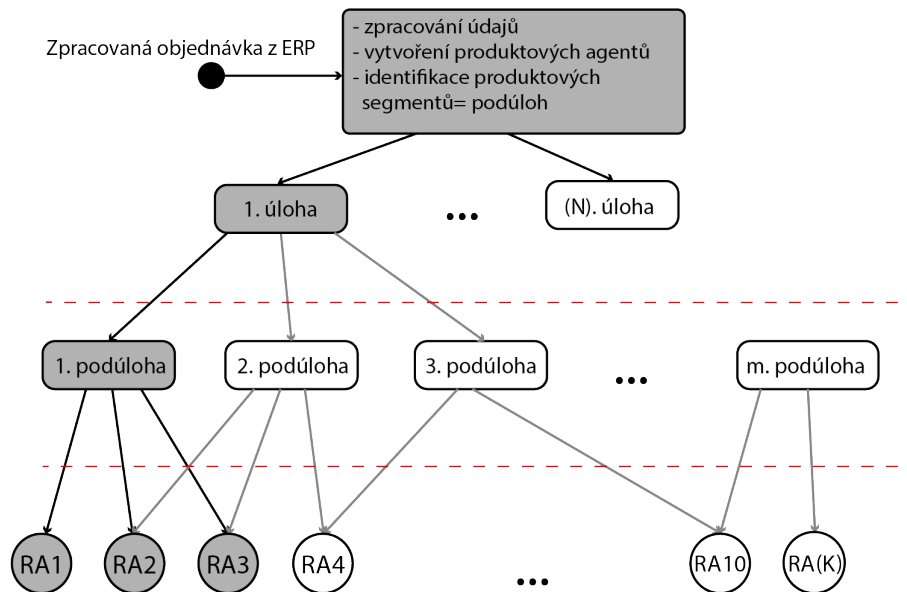
Model inteligentního výrobku (Product agent)

Model produktového agenta charakterizují tyto atributy:

- Požadavky na výrobní zdroje (resource agenty) - dle ISA 95 sem patří *fyzické vybavení, personální kapacity a materiálové vstupy do výroby*.
- Technologický plán výroby, tj. postup procesních operací nutných pro zabezpečení výroby fyzického produktu. Plán vzniká tak, že každá objednávka přicházející do systému se rozdělí na N úloh ($P = P_1, P_2, P_3, \dots, P_n$). Každá úloha je zastoupena

jedním produkt agentem a dále se dělí na M podúloh $Pd = p_{d1}, p_{d2}, \dots, p_{dM}$. Takto lze podle složitosti a potřeby výrobního plánu dělit úlohy i víceúrovňově. Sjednocením všech stupňů podúloh příslušejících jedné technologické operaci vzniká *koordináční skupina*, která koresponduje s koordinační skupinou resource agenta. Pokud inteligentní výrobek vyžaduje vykonání posloupnosti M podúloh P_{nmk} ³ a každá podúloha má známý čas vykonání T_{nmk} , potom životnost produkt agenta T_i je možno vyjádřit jako $T_i \geq \sum_{MK} T_{nmk}$.

- Historie všech vykonaných operací spolu s datovými reporty pro účely zpětného trasování a zabezpečení požadované kvality.
- Výrobní cena produktu, která reflektuje aktuální náklady požadovaných technologických operací. Tyto náklady ovlivňuje zejména personální náročnost výroby a aktuální ceny vstupujících energií do technologie výroby.
- Stavové veličiny charakterizující inteligentní výrobek. Sem patří životnost produkt agenta T_i , nejpozdější čas dokončení produktu d_n a požadovaná kvalita produktu.
- Unikátní identifikátor výrobku v rámci systému.



Obrázek 7.6: Struktura hierarchického dělení technologických operací

7.2.2 Plánování operací

Plánování probíhá prostřednictvím protokolu založeného na spolupráci (kapitola 7.1.2). Jsou zde definovány dva nejdůležitější typy zpráv pro koordinaci mezi *produkt agentem* a *resource agentem*: *poptávka* (*bid request*) a *nabídka* (*offer*).

³ m označuje pořadí podúlohy, k označuje číslo výrobního zdroje, na kterém se daná úloha vykonává

- Poptávka po službě je v CNP protokolu zastupována metodou *call for proposal*, která slouží k odeslání žádosti o výrobní zdroje všem resource agentům v systému. V případě, že resource agenti jsou integrováni do koordinačních skupin dle vzájemné příslušnosti (viz kapitola 7.3), mohla by se komunikace zjednodušit pouze na oslovení resource agentů v příslušné koordinační skupině. Následně v rámci koordinační skupiny by např. na základě ukazatelů OEE (či jiných) proběhlo přerozdělení konkrétnímu resource agentovi. Odpovědí na poptávku je *nabídka* ze strany resource agenta.
- Nabídka služby je v CNP protokolu zastoupena metodou *proposal*. Jedná se o zprávu, ve které resource agent předává informace o poskytovaných službách, spolu s časovými možnostmi v podobě rezervačních front.

Struktura žádosti o výrobní zdroje (Bid Request) obsahuje parametry:

$$B_{req} = (PA, P_{id}, d_n)$$

První parametr PA představuje identifikaci produkt agenta, který zprávu B_{req} odesílá. Aktuální míru rozpracovanosti úlohy reprezentuje parametr P_{id} . Ten obsahuje množinu doposud nerealizovaných technologických operací - podúloh, které jsou poptávány od resource agentů. Poslední parametr d_n obsahuje nejpozdější čas, do kterého je nutné poptávané operace rozvrhnout do rezervačního plánu resource agenta. Resource agent tím zjistí, jak určit rozpětí časového úseku své rezervační fronty, který je ještě relevantní pro žádajícího produkt agenta a toto rozpětí mu nabídnout. Konkrétní časové sloty si pak vybírá produkt agent na základě vlastní heuristiky výběru.

Jako odpověď na žádost přichází ze strany resource agenta nabídka, jejíž struktura je tato:

$$Bid = (Str_e, Str_x, Prp_p, Prp_{np}, R_T)$$

Parametry *sekvence možných událostí* - $Str_e = e_0, e_1, \dots, e_n$, sekvence možných stavů fyzického systému - $Str_x = x_0, x_1, \dots, x_n$ spolu s Prp_p a Prp_{np} definují schopnosti resource agenta v kontextu požadovaných služeb, které může tento agent alokovat. Parametr R_T je v této fázi velmi důležitý pro výběr nejvhodnějšího volného časového slotu výrobního zdroje.

Produkt agent po přijetí nabídky iniciuje vytvoření rezervace, tj. rozšíření existující množiny $Sc_e = (t_0, t_1), (t_2, t_3), \dots, (t_{m-1}, t_m)$ časových úseků v R_T o vybraný časový slot. Parametry Str_e a Str_x odesílané nabídky se získají porovnáním schopností resource agenta s těmi, které jsou poptávány ze strany produkt agenta.

$$X_m = \{x \in X | Prp_p(x) \in P_{id} \cup Prp_{np}(x) \in P_{id}\} \cap \{x_{i+1} = Tr(x_i, e_i)\}$$

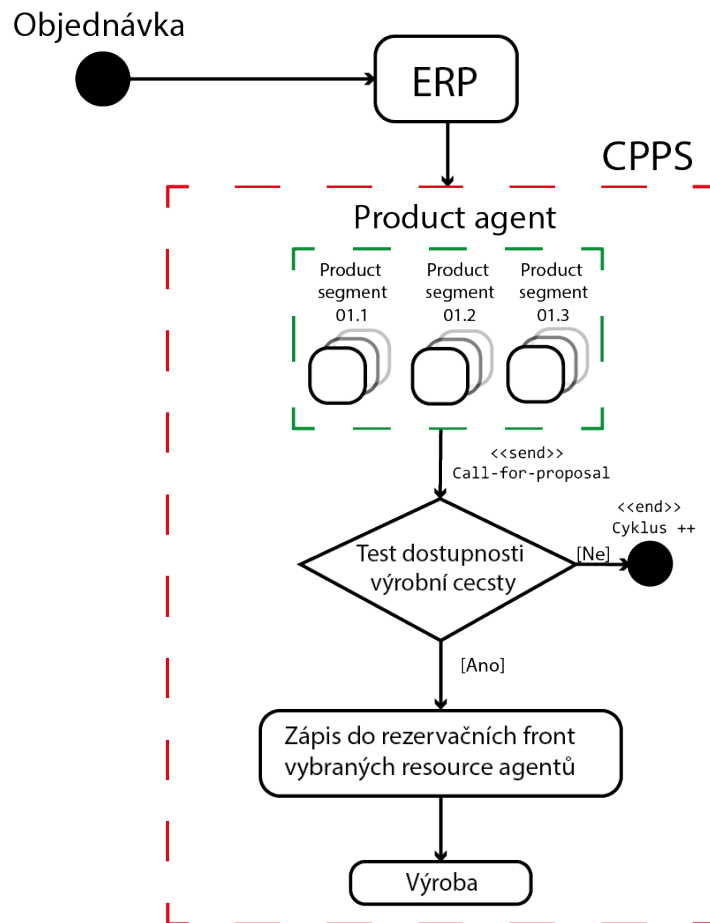
$$\text{kde } 0 \leq i \leq n - 1$$

Ze zápisu výše plyne, že resource agent může odpovídat na poptávku služeb pouze v případě, kdy najde takovou sekvenci stavů X_m , která mu umožní dosáhnout požadovaného koncového stavu vyráběného produktu. Po odeslání poptávkové žádosti čeká pro-

dukt agent na nabídky odeslané resource agenty. Pokud je do určitého času nedostane v požadovaném počtu, či z nich nedokáže vytvořit model výrobního prostředí, může modifikovat parametry žádosti a odeslat poptávky znovu. Nejčastěji se pokusí prodloužit čas platnosti (nejpozdější čas vyrobení), případně požádat o asistenci operátora.

V diagramu na obrázku 7.7 jsou zobrazeny základní kroky přidělování výrobních zdrojů jednotlivým produktovým agentům. Po přijetí objednávky systém ERP tuto objednávku zpracuje, určí její výrobní postup, prioritu, a rozvrhne její vyrobení v rámci dlouhodobého strategického plánování podniku. Po tomto kroku je objednávka přesunuta do výrobní části, kterou zastupuje multiagentní systém CPPS. Následně se vytvoří příslušný počet produkt agentů podle modelu (viz výše). Produktoví agenti mezi sebou nekomunikují. Jejich jediná interakce je přímo směrem ke konkrétním resource agentům, či ke koordinačním skupinám resource agentů.

V dalším kroku produkt agenti sesbírají nabídky od kompetentních resource agentů, tj. těch, kteří kladně odpověděli na jejich požadavky a disponují požadovanými schopnostmi (vybavením, materiálem a lidskými kapacitami). Produkt agent se na základě svého technologického plánu pokusí sestavit výrobní cestu (cesty). V případě úspěchu se přistoupí k zápisu do rozvrhů všech dotčených výrobních zdrojů.



Obrázek 7.7: Vývojový diagram základních částí algoritmu

7.2.3 Heuristika vkládání úloh do rozvrhů RA

Zájmem každého produkt agenta je minimalizace nákladů na výrobu produktu, jehož v multiagentním systému zastupuje. Dynamická strategie rozvrhování v reálném čase a bez implicitního rozvrhu umožňuje jen velmi omezené možnosti optimalizace. Produktivní agenti přicházejí do systému stochasticky, přičemž dopředu nejsou známy žádné parametry potřebné pro tvorbu optimálního rozvrhu. Strategie vkládání úloh do rozvrhů výrobních zdrojů, která je popsána níže, vychází z principů JIT organizace výroby. Podle této teorie jsou v nákladech na plánování zahrnuty výrobní náklady, náklady na vzniklé dočasné zásoby a náklady za penalizaci zpoždění. Úlohy by měly být správně přidělovány jednotlivým výrobním zdrojům tak, aby byl minimalizován čas jejich rozpracovanosti, a zároveň byly vyrobené co nejdříve svému nejpozdějšímu času vyrobení (JIT).

Popsaná optimalizace je implementována nad globálními náklady na celou výrobní cestu produkt agenta. Lokální optimalizace rozvrhu na úrovni konkrétních resource agentů není v této heuristice zahrnuta. Z důvodu flexibilní povahy uvažovaného typu výroby Job Shop, velké variability výrobků, ale zejména precedenčních omezení operací by tato změna rozvrhu na jednom z resource agentů ve výrobní cestě produktu způsobila změnu rozvrhů všech následujících operací. Matematický popis optimalizace výběru časového slotu je:

S je množina zdrojů $\{1, 2, \dots, S\}$.

M je množina podúloh (produktových segmentů) $\{1, 2, \dots, M\}$.

$T^{(S)}$ jsou dostupné pracovní hodiny stroje s v rozsahu horizontu plánování.

$C^{(S)}$ jsou časově proměnlivé náklady za jednotku časového slotu stroje s .

$P^{(S)}$ je časová náročnost výroby na stroji s (počet časových slotů).

d_m je nejpozdější čas výroby podúlohy m .

δ_T^m je váhovací koeficient zpoždění podúlohy m (přípustný pouze pro poslední technologickou operaci p_m . V opačném případě by rozvrh porušil precedenční omezení). Pro podúlohy (p_1, \dots, p_{M-1}) se zadává dostatečně velká hodnota.

λ váhovací koeficient pro předčasnou realizaci úlohy.

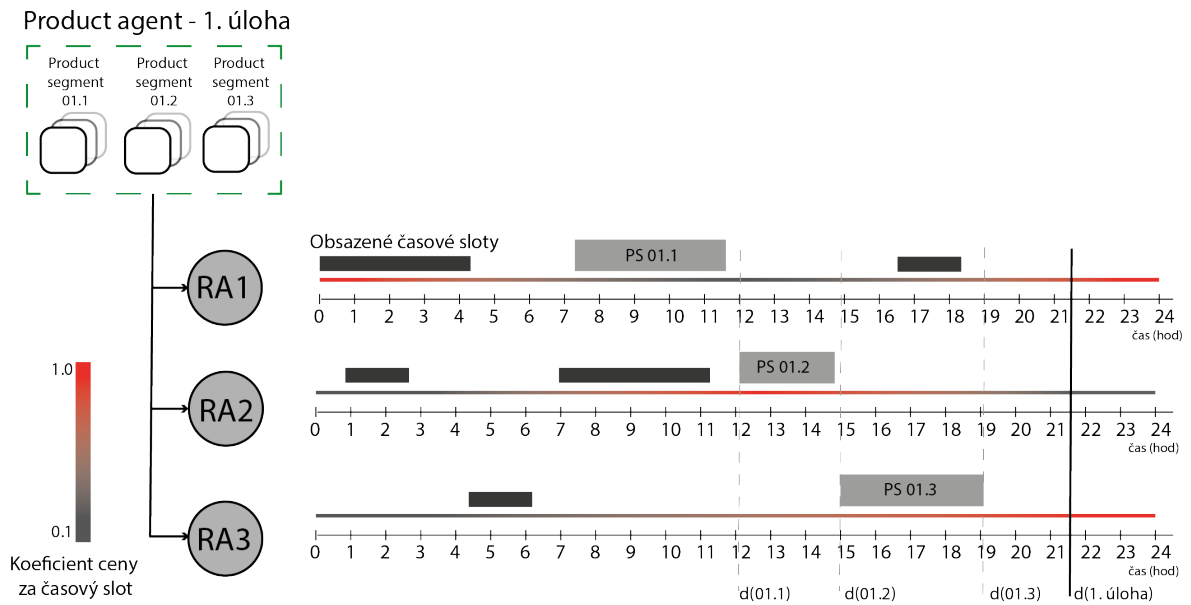
$T_m^{(s)}$ je rozvrhovaný čas ukončení zpracování podúlohy m pomocí stroje s .

Heuristiku vkládání je pak možné určit výpočtem výrazu (7.5). První člen vyčísľuje náklady na výrobu objednávky, druhý člen vyjadřuje penalizaci vzniklou zpožděním všech podúloh objednávky a třetí vyjadřuje penalizaci vzniklou předčasným vyrobením podúloh.

$$\min \left(\sum_{s=1}^S C^{(S)} \cdot P^{(S)} + \sum_{m=1}^M \delta_T^m (T_m^{(S)} - d_m) + \sum_{m=1}^M \lambda \cdot (d_m - T_m^{(S)}) \right) \quad (7.5)$$

$$T_m^{(S)} > d_m, \lambda = 0$$

$$T_m^{(S)} < d_m, \delta_T^m = 0$$



Obrázek 7.8: Ukázka heuristiky vkládání výrobních operací do rozvrhů RA [137]

V příkladu na obrázku 7.8 je znázorněn rozvrhovací proces, za který je odpovědný produkt agent konkrétní úlohy z pomyslné objednávky. Produkt agent v odpovědích *Bid* resource agentů obdržel informace z rezervačních front těchto agentů. Cílem produkt agenta je nalézt kompromis mezi co nejmenší penalizací za předčasné či pozdní vyrobení a cenou časových slotů konkrétního výrobního zdroje. Tato cena je určena koeficientem, který je graficky znázorněn v rezervačních frontách na obrázku. Cena konkrétního časového slotu se může odvíjet od mnoha faktorů. Ve výrobním podniku to může být např. cena elektrické energie, cena za lidskou pracovní sílu (rozdílná v noci a ve dne), cena dopravy, materiálu, apod.⁴ Koeficient váhy pro velikost penalizace za brzké či pozdní vyrobení produktu nemusí být konstantní. Naopak je žádoucí, aby koeficient přiměřeně narůstal s odchylkou od optimálního času výroby, a tím zvyšoval penalizaci za velké časové nejistoty ve výrobě. V závislosti na implementaci může koeficient růst lineárně, případně po vhodné křivce.

Algoritmicky je možné heuristiku dynamického rozvrhování popsat takto: Začíná se naplánováním poslední z podúloh *PS 01.3* do časových slotů s minimální hodnotou

⁴V kontextu spotřeby elektrické energie může toto plánování sloužit jako nástroj pro regulaci energetických veličin, například dodržování zaslíbeného čtvrt hodinového maxima dodaného výkonu. Cena časového slotu výrobního zdroje by tak kopírovala odhadovanou potřebu snížení/zvýšení odběru elektrické energie, čímž by jej kompenzovala.

nákladové funkce. Začátek výroby této podúlohy tak tvoří nejpozdější čas realizace předposlední úlohy $d_{(01,2)}$ ve smyslu dodržení precedenčních omezení úloh. Dále se opět vypočtou časové sloty s nejnižší hodnotou nákladů rozvrhu a podúloh *PS 01.2* a vloží se do rezervační fronty druhého resource agenta. Nyní je zřejmé, že navržená heuristika vkládání úloh minimalizuje také čas rozpracovanosti celé úlohy - produkt agenta, neboť penalizuje každé vzdálení se od ideálního průběhu výroby bez prostojů. Rekurzivně proběhne výběr časových slotů pro všechny další podúlohy (produktové segmenty). Následuje rezervace časových slotů s využitím protokolu CNP s potvrzování.

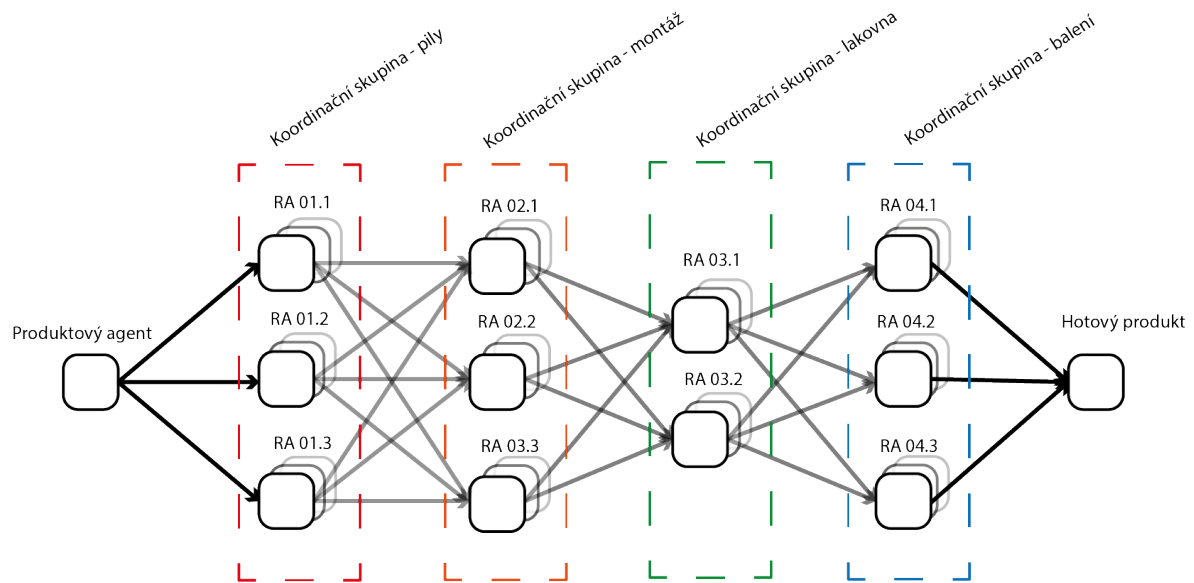
7.2.4 Možná rozšíření heuristiky vkládání

Popsaná heuristika může být rozšířena o prediktivní rozměr v podobě jednoduchého plánování v klouzavém časovém horizontu. Přicházející úlohy v podobě produkt agentů by se shromažďovaly v předem známém časovém okně. Po jeho uplynutí by na základě priorit, či např. na základě jednoduchého dispečerského pravidla došlo k jejich seřazení. Podle pořadí by produkt agenti přistupovali k alokaci zdrojů. Tím by bylo dosaženo jisté formy zvýhodnění pro ty produkt agenty, jejichž výroba je v určitém smyslu důležitější. Ti by tedy měli v úvodu na výběr z většího počtu časových slotů resource agentů. Při návrhu a implementaci této úpravy heuristiky by bylo nutno dbát na to, aby nedošlo ke vzniku jediného centrálního koordinačního členu (agenta), a tím nebylo porušeno paradigma čistě decentralizovaného přístupu.

7.3 Simulace MAS řízení výroby

Empirické hodnocení navrženého přístupu dynamického plánování v CPPS architektuře vyžaduje definování konkrétních instancí výrobního prostředí a úloh. Navržený simulační model slouží pro zobrazení chování reálného výrobního systému - zachycuje všechny důležité dynamické vlastnosti včetně náhodných jevů či poruchovosti výrobních zařízení. Simulace je cílena zejména na reprodukci vykonávání výrobní sekvence s použitím navržených rozvrhovacích metod.

Pro prezentaci teoretických výstupů této kapitoly, zejména navrženého algoritmu, byla vytvořena simulace CPPS agentního systému reprezentujícího montážní linku nábytku. V rámci simulace se neuvažuje konsolidovaný stálý odběr vyrobených kusů, namísto toho převažuje nepředvídatelný tok objednávek bez možnosti jakékoli predikce. Je proto uplatněna strategie make-to-order, přičemž není možnost držet skladové zásoby. Z tohoto důvodu je při rozvrhování výroby kladen důraz na metodiku JIT. Tažná strategie výroby umožňuje realizovat široké portfólio možných produktů s individuálními vlastnostmi, jejichž výroba vyžaduje flexibilitu volby výrobních tras vždy podle konkrétní specifikace. Každý produkt tedy může být realizován pomocí množiny individuálních technologických operací vázaných precedenčními omezeními. Tyto skutečnosti vedou na rozvrhovací problém typu Job-Shop, jehož suboptimální



Obrázek 7.9: Rozvržení výrobních zdrojů v simulovaném podniku

řešení může být získáno pomocí navržených agentně orientovaných algoritmů (Předpoklady využitelnosti MAS jsou uvedeny v kapitole 6.4).

Simulovaný model obsahuje fixní strukturu výrobní linky, organizaci a technologické procesy výroby, které jsou dopředu jasně definovány. Jeho uspořádání je znázorněno na obr. 7.9, který přibližuje rozvržení jednotlivých resource agentů do koordinačních skupin podle příslušnosti k určitému typu nabízené služby. Agenti v těchto skupinách se od sebe odlišují specifikacemi svých vlastností ve vztahu k výrobě. Definice těchto vlastností má základ ve standardu ISA 95, který definuje objektové modely výrobního procesu. Ze standardu byly převzaty objektové modely materiálu (*material*), vybavení (*equipment*) a lidských zdrojů (*personel*). Každý z těchto modelů dále obsahuje třídu (*class*) a definici (*definition*) vlastností. Všichni resource agenti jsou v simulaci zastoupeni svými procesními segmenty, které jsou definovány jako spojení uvedených tří objektových modelů. Počet a typ různých **procesních** segmentů agenta určuje, jaké technologické operace, s jakými parametry, v jaké kvalitě (apod.) dokáže resource agent vykonat. Z obrázku je zřejmá úplná vzájemná propojenost resource agentů - nutný předpoklad flexibilní výrobní trasy. V realitě je taková trasa tvořena nejčastěji dopravníky či jinými manipulátory v případě plně automatických provozů, případně může roli hrát také ruční transport materiálu. V simulaci není přesun výrobků mezi výrobními zdroji uvažován. Jeho doplnění by díky škálovatelnosti CPPS architektury znamenalo jednoduché přidání resource agentů manipulátorů, případně jejich koordinační skupiny.

Surový materiál pro výrobu dřevěného nábytku je k dispozici ve standardních rozměrech a kvalitě. Pro zjednodušení jsou tyto parametry vyhovující pro všechny myslitelné varianty produktů. Ve výrobním podniku se např. nacházejí různá pracoviště pro dělení materiálu (pily) s různými vlastnostmi. Během simulace se nepočítá se skladovými kapacitami, proto je surový materiál umisťován na jednotlivá pracoviště pily

```

yield return new ResourceAgent(
new List<ProcessSegment>()
{
    new ProcessSegment(
        personel: "pavel[truhlář]",
        equipment: "okružní_pila_1[pila]",
        material: "dub[dřevo],buk[dřevo],modřín[dřevo]"),

    new ProcessSegment(
        personel: "pavel[truhlář]",
        equipment: "okružní_pila_2[pila]",
        material: "borovice[dřevo]"),

    new ProcessSegment(
        personel: "pavel[truhlář]",
        equipment: "přímočará_pila_1[pila]",
        material: "smrk[dřevo]")
}, "truhlárna1");

```

Obrázek 7.10: Výpis z kódu konstruktora resource agenta - procesní segmenty

podle jejich specifikace. Následně se v trase produktu výrobou nachází montážní dílna s různými zaměřenými na konkrétní technologickou operaci. Další pracoviště jsou definovány jako lakovna s možnou specializací na konkrétní typy lakování a různé barvy. Poslední z nabízených výrobních zdrojů jsou pracoviště balení. Svými kapacitami odpovídají rozsahu velikostí vyráběných produktů. Výpis na obr. 7.10 demonstruje definici objektového modelu resource agenta. Podobně jsou specifikovány také další typy resource agentů.

Po konfiguraci výrobní části simulace je nutné definovat také samotné vyráběné produkty. Každý z produktů je reprezentován svým produktovým agentem, který v multiagentním systému zastupuje jeho zájmy (tedy zejména hlavní cíl - být vyroben). Agent inteligentního výrobku je, stejně jako resource agent, rozdělený na segmenty *produktu*. Tyto segmenty definují požadavky jednotlivých technologických operací. S výhodou byl tento model využit v algoritmu pro hledání a přiřazování úloh pro vykonání takovým zdrojům, které splňují požadované vlastnosti. Produkt agent definuje další parametry potřebné pro rozvrhování, zejména se jedná o dobu trvání výroby (je-li známa) a nejpozdější termín realizace (viz výpis na obr. 7.11). Simulovaný podnik vyrábí nábytek, proto byly zvoleny výrobky *židle*, *police* a *skříňka*. Produktoví agenti zastupující tyto výrobky během inicializace obdrží konfiguraci parametrů jednotlivých technologických operací a následně jsou přidáváni do agentního světa.

7.3.1 Validace simulace řídicí logiky

Připomeňme, že simulovaný příklad, popsáný v rámci minulé kapitoly, přestože zachycuje mnoho důležitých vlastností představeného konceptu organizace výroby ve smyslu MAS, je navržen s velkou mírou abstrakce a zanedbává parametry plynoucí z reálné výroby.

Pro sledování průběhu simulace bylo vytvořeno jednoduché grafické okno (viz ob-

```

public static ProductAgent CreateChairAgent()
{
    return new DistributedLib.Agents.SimpleAgents.ProductAgent(
        new List<ProductionSegment>()
        {
            new ProductionSegment(1,
                personel: "[truhlář]",
                equipment: "[pila]",
                material: "dub[dřevo],buk[dřevo]",
                new Dictionary<string, dynamic>() { { "production_time", 16 } })),
            new ProductionSegment(2,
                personel: "[montáž]",
                equipment: "[stul]",
                material: "",
                new Dictionary<string, dynamic>() { { "production_time", 15 } })),
            new ProductionSegment(3,
                personel: "[lakovna]",
                equipment: "[stříkací_pistole]",
                material: "fermež[barva]",
                new Dictionary<string, dynamic>() { { "production_time", 12 }, { "color", "red" } })),
        }, Clock.Now);
}

```

Obrázek 7.11: Výpis z kódu instanciacie produkt agenta - segmenty produktu

rázek A1). V jeho levé části je zobrazen přehled vložených produkt agentů spolu se sledovanými parametry (aktuálním stavem, rozpracovaností produktu a případně název resource agenta, na kterém probíhá výroba spolu s časem ukončení probíhající operace). Ve spodní části okna pak vidíme přehled modelu výrobní linky, tedy seznam resource agentů. Pro každý je opět uvedeno, který z produktů aktuálně obsluhuje, stejně tak jako statistické údaje pro měření produktivity výrobního zdroje. Informace o průběhu simulace se nacházejí v pravé části okna. Vzhledem k tomu, že je celý systém založen na generování diskrétních událostí, je možné v průběhu simulace zrychlovat, či zpomalovat tok času. Simulátor tak může snadno sloužit pro ladění a optimalizaci organizačních schémat výroby i v delších časových měřítcích.

Stav simulátoru na počátku výroby je znázorněn na obrázku A.1. Logy komunikace zde uvádějí první zprávy o akceptacích nabídek resource agentů. Následně každý z produkt agentů ve své režii s využitím heuristiky popsané v kapitole 7.2.3 plánuje a rezervuje vlastní výrobu. Konkrétně je v okně simulace zobrazena počínající výroba prvního ze tří produkt agentů (agenta s nejdřívějším časem zpracování zakázky). Z dalšího obrázku ze závěru simulace A.2 je zřejmé, že první produkt je úspěšně vyroben (v logu komunikace je možné vidět, že algoritmus správně naplánoval výrobu jednotlivých produktových segmentů v čase tak, aby minimalizoval nákladovou funkci rozvrhu, minimalizoval dobu rozpracovanosti a zároveň dodržel principy výroby JIT). V uvedeném simulačním příkladu byl koeficient ceny za časové sloty konstantní. Časy dodání jsou pro druhý a třetí produkt 220 s, resp. 320 s. V okně simulátoru z obr. A.2 je celá simulovaná úloha dokončena. Všechny produkty jsou označené, jako úspěšně vyrobené. Tento jednoduchý scénář simulace výroby poskytuje ověření správnosti navržené topologie, organizace a rozvrhování v prostředí CPPS.

Simulátor samozřejmě předpokládá komplexnější ověřování algoritmů organizace distribuované výroby. Umožňuje simulovat zpoždění nad některým z rozvrhovaných

produktových segmentů se zvoleným statistickým rozložením. Rovněž podporuje vyhodnocování OEE parametrů a poskytuje možnost řídit využití redundantních výrobních zdrojů (v kompetenci koordinačních skupin).

7.4 Shrnutí kapitoly

V rámci této kapitoly byly popsány existující algoritmy pro distribuovanou organizaci výroby založené na agentním přístupu a jejich možné variace. Dále byl adaptován algoritmus CNP, který byl rozšířen o vlastní heuristiku vkládání úloh. Tento algoritmus byl následně simulován ve vytvořeném prostředí.

Při implementaci simulované výroby byly empiricky porovnávány dvě metody vkládání nových úloh do rezervačních front výrobních zdrojů.

- Vložení nové úlohy vždy na poslední pozici rezervační fronty (na konec aktuální výrobní sekvence) resource agenta
- Vložení nové úlohy na nejvýhodnější pozici s využitím popsané heuristiky vkládání (tedy výpočet ceny vložení nové úlohy do všech možných pozic rezervační fronty a vložení do časového slotu, který vede k minimalizaci nákladové funkce).

Jednoduchá metoda vkládání úlohy vždy na konec rezervační fronty neposkytuje žádnou možnost ovlivnit čas fyzické realizace produktu, tedy nebere vůbec v úvahu nejpozdější čas výroby. Důsledkem jsou zvýšené náklady na skladování při příliš brzké realizaci a rovněž pravděpodobnost, že termín realizace nebude vůbec dodržen. Úspěšnost celého systému rozvrhování je tak velmi citlivá na pořadí příchodu jednotlivých produktů do systému alokace zdrojů. Navíc pokud některý z produktů vyžaduje vykonání technologické operace, která je precedenčně závislá v procesu výroby, vytvářejí se v rezervačních frontách výrobních zdrojů přirozeně nevyužitá časová okna.

Je tedy možné říci, že uvedená metoda je postačující pro typ výroby *Flow shop*, který je omezen na sekvenční tok bez vzájemné propojenosti výrobních tras. Pro typ *Job shop* je navržená architektura multiagentní organizace výroby nevhodná.

Kapitola 8

Testbed Průmyslu 4.0

V současné době, kdy jsme svědky postupné přeměny průmyslové výroby směřující k naplnění filozofie Průmyslu 4.0, vzniká na vysokých školách potřeba uvést do výuky velké množství nových, z velké většiny interdisciplinárních témat.

Na základě dosavadních zkušeností je možné říci, že taková témata nejsou pro studenty oboru automatizace právě snadno uchopitelná. To je způsobeno zejména faktem, že studenti jsou vedeni převážně teoretickým směrem a v rámci časové dotace výuky jsou schopni nastudovat a osvojit si pouze základní principy v jednotlivých oborech. Přechod k problematice tak komplexní, jako jsou výrobní procesy a vše co s nimi souvisí, však představuje velký skok, který je bez předchozích zkušeností v omezeném čase realizovatelný jen těžko. Dalším z problémů je již zmíněná interdisciplinarita, kdy student automatizace, zabývající se dosud výhradně navrhováním regulátorů, aplikací snímačů, měřením jejich charakteristik, případně programováním základních řídicích algoritmů, se má stát zároveň strojním inženýrem se znalostí konstruování a informatikem – expertem na IT systémy a jejich vzájemnou interoperabilitu. A to vše v kontextu komplexního výrobního prostředí [6].

Bylo by snadné před tímto faktem zavřít oči, tvrdit, že teoretická příprava studentům postačuje, a zachovat status quo. Do jisté míry lze říci, že po absolventy nakonec praxe naučí vše, co budou potřebovat. Ve Skupině průmyslové automatizace na Ústavu automatizace a měřicí techniky, kde působíme, však převládá názor, že by studenti měli být připraveni na praxi, která naprostou většinu z nich čeká, a měli by již v průběhu studia získat k teoretickým znalostem nejen strohé teoretické informace, jako např. výčet výrobních technologií a používaných IT nástrojů, ale také praktické dovednosti a poznatky. Právě to vedlo k základní myšlence návrhu a realizace zkušebního pracoviště – testbedu – které umožní uchopit, demonstrovat a ověřovat principy Průmyslu 4.0 nejen v teoretické rovině, ale také prakticky. Mimo klasická témata vycházející z řešení majoritních problémů charakteristických pro nižší stupně automa-

tizační pyramidu¹, vychází využití testbedu ze záměru přiblížit studentům představu chytré továrny, ve které se uplatňují některá nová paradigmatu:

1. Základními prvky chytré továrny jsou kyber-fyzické produkční systémy (CPPS).
2. Každá buňka i celý testbed nejprve vzniká v maximální možné míře jako digitální model sloužící k ověření funkčnosti ještě před samotnou výrobou.
3. *One-piece flow* - možnost customizace výroby až ke konkrétnímu kusu produktu. Každý produkt může být tedy vyráběn podle vlastního podrobného předpisu bez nutnosti rekonfigurace výrobní technologie.
4. Možnost navrhovat algoritmy pro řízení skutečné výroby fyzických produktů. Tuto výrobu je možné, s ohledem na konstrukci zkušebního pracoviště chápat jako diskrétní výrobu jednotlivých kusů. Pracoviště však také umožní demonstrovat principy dávkové výroby podle standardu ISA 88.
5. Možnost navrhnout vlastní nebo integrovat existující systém pro řízení výroby (MES) a systém pro plánování podnikových zdrojů (ERP) opět nad skutečnou výrobou fyzických produktů. S ohledem na charakter výsledných produktů i cenu vstupních surovin je možné realizovat i relativně dlouhé výrobní sekvence, ze kterých lze získávat, zpracovávat a vyhodnocovat relevantní data.
6. Výrobní zařízení je modulární, všechny entity jsou horizontálně propojeny bez nutnosti řízení nadřazeným systémem a svou činnost umí společně koordinovat.
7. Využití rozšířené reality – testbed je doplněn systémem rozšířené reality. Při provozu lze prostřednictvím kamery v mobilním zařízení a příslušné aplikace získat přehled o aktuálním stavu výroby v jednotlivých buňkách.

Protože je testbed řešen jako *modulární decentralizovaný systém*, v němž jsou jednotlivé buňky navrhovány a konstruovány zcela samostatně, vzniklo v průběhu vývoje mnoho dílčích projektů, vhodných jako zadání pro závěrečné práce studentů bakalářského i magisterského studia. Během několika let realizace projektu jich bylo v souvislosti s testbedem takto zadáno a úspěšně obhájeno několik desítek.

Základním cílem těchto prací byl *testbed Průmyslu 4.0* a předmětem řešení pak *návrh a konstrukce* jeho části. Tato kategorizace vždy zásadně ovlivňovala pohled na celou práci. Ohlédneme-li se do nedávné historie, zjistíme, že zadání prací velmi podobná těm současným, byla na ústavech a katedrách automatizace elektrotechnických fakult zcela standardní. Chtěl bych podtrhnout to slovo *podobná*. Pokud se termíny *návrh a konstrukce* vyskytly u standardních zadání, byl cílem pouze návrh a realizace elektrického a softwarového uspořádání. Činnosti v takových projektech se odehrávaly v rámci všeobecně známé automatizační pyramidy (obrázek 3.3 uprostřed).

¹Jedná se zejména o programování řídicích a obslužných algoritmů, zpracování dat ze snímačů a řízení akčních členů, řešení různých posloupností zpracování algoritmů či návrhů spojitých regulátorů, to vše pro řízení skutečných - fyzických systémů

V rámci projektu realizace testbedu však zadání studentských závěrečných prací obsahovala jako cílové činnosti *návrh a konstrukce*. Myslí se tím návrh a konstrukce mechanického uspořádání vyvíjeného zařízení ve smyslu anglického slova *design*. Student – konstruktér tedy není postaven před hotové fyzické zařízení, které by měl vybavit optimální moderní automatizací, nezačne svoji práci v polovině návrhu, ale uplatní svůj tvůrčí um od začátku projektu, který je jeho vizí, a to jak v oblasti mechanické konstrukce a operační technologie, tak při integraci s informačními technologiemi.

Při návrhu se k testbedu přistupovalo účelově jako k chytré továrně. Testbed byl proto navržen jako modulární kyber-fyzický systém, přičemž každý modul (výrobní či skladovací buňka) je koncipován jako samostatný CPPS. Funkčně jsou tyto systémy propojeny tokem informací a materiálu. Nepředpokládá se, že by studenti při konstrukci dílčích buněk v rámci jedné práce vytvořili kompletní kyber-fyzický systém CPPS s celou pro něj definovanou strukturou a všemi jeho vlastnostmi, včetně digitálního dvojčete. Nicméně cesta, která byla pro řešení projektu zvolena, k vytvoření systémů CPPS nakonec vede.

Stejná úvaha platí i pro vlastní počítačový návrh. Nejsme zatím na úrovni vytváření digitálního výrobku (továrny), kdy by návrh probíhal paralelně s fyzickou konstrukcí a formou zpětné vazby od senzorů by byl počítačový návrh korigován. Jednotlivé buňky se vytvoří ve formě virtuálního výrobku, virtuálně se otestují a pak se vyrobí. Vzhledem ke dlouholeté spolupráci pracoviště se společností Siemens byla jako PLM platforma pro práci na testbedu zvolena platforma Teamcenter. V této platformě se nachází i kompletní portfolio řešení digitální továrny Tecnomatix, které obsahuje mimo jiné inženýrské nástroje. Mezi nimi je i nástroj NX, ve kterém byly vytvářeny jako virtuální komponenty všechny díly jednotlivých buněk. S využitím doplňku nástroje NX - *Mechanics Concept Designerem* byly provedeny rovněž mechanické simulace pohybujících se částí v buňkách. Celý inženýrský postup návrhu a realizace byl tedy zjednodušen a postupy uvedené v kapitole 2.7 této práce tak byly využívány pouze v omezené míře.

Naprostou nezbytností při realizaci jednotlivých projektů je využívání technologie aditivní výroby (viz kapitola 1.4.2), která je nedocenitelná při prototypování a malosériové výrobě. Bez možností jejího využití by konstrukce testbedu v našich podmínkách vůbec nebyla možná.

8.1 Konstrukce testbedu

Základem platformy kompaktního testbedu je pracovní plocha, na které jsou vytvořeny sloty s univerzálním rozhraním sloužící pro umístění jednotlivých výrobních buněk. Pod pracovní plochou je místo pro podpůrné systémy - rozvaděče elektroinstalace, kontrolér robotu a síťová infrastruktura. Celek - platforma společně s buňkami pak tvoří jednoduchou výrobní linku, ve které jsou postupným procesem vyráběny produkty. Transport kontejnerů s produkty mezi jednotlivými pracovišti - buňkami a ven z továrny realizuje SCARA manipulátor, který je pevnou součástí platformy.

Všechny buňky jsou vybaveny totožným rozhraním:

- místem pro vkládání a výběr kontejneru s výrobkem - na toto místo kontejnery pokládá a z něj odebírá SCARA manipulátor;
- připojením zdroje elektrické energie a smyček pro funkční bezpečnost;
- připojením komunikačního rozhraní Ethernet;
- světelným indikátorem pro zobrazení stavu buňky;
- dotykovým LCD panelem pro indikaci detailních informací stavu buňky, stavu aktuální výroby a případné potřeby servisu.

Výrobní buňky, které se umísťují na platformu testbedu, mají unifikované rozměry (330×330×500 mm). Tyto rozměry představují při realizaci studentských prací zásadní omezení a (nejen) kvůli němu je nutné důsledně využívat modelovací a simulační nástroje. Díky jednotným rozměrům mohou být buňky na platformě umístěny v libovolných kombinacích. V současné době disponuje testbed těmito realizovanými a funkčními buňkami:

- **Sklad kontejnerů**, z něhož je na počátku výroby distribuován prázdný kontejner. Kontejnery jsou v buňce stohovány ve čtyřech pozicích, přičemž buňka disponuje schopnostmi změřit a určit zbývající počet kontejnerů [25].
- **Zásobník vstupních surovin**, sestávající z pozic pro různé typy surovin s technologií přečerpávání obsahu a tříosým manipulátorem, který dopraví kontejner ke správnému zásobníku a zpět do výdejního místa [86, 43].
- **Zásobník vstupních surovin**, skládající se ze čtyř tanků, systému pro přečerpávání materiálu a systému pro měření teploty a hladiny v těchto tancích. Oba zásobníky pracují na odlišných principech [18].
- **Homogenizátor**, který umožňuje kontejner s vyráběným produktem hermeticky uzavřít a protřepat tak, aby došlo k dokonalému promísení jednotlivých složek směsi [51, 45].
- **Sytič**, který v dávkách připravuje sycenou kapalinu, kterou je následně možné přimíchat do produktu. Tato buňka mimo vlastní technologii sycení obsahuje zásobu kapaliny a sytícího plynu [32].
- **Drtič ledu** s dávkovačem pro rychlé a přesné dávkování ledu do kontejneru [44].
- **Dopravníkový pás**, který slouží pro distribuci hotových výrobků z výrobní zóny k zákazníkům [108].
- **Robotický manipulátor** pro transport kontejnerů mezi jednotlivými buňkami a na dopravníkový pás [22].

Na nejnižší (procesní) úrovni je každá z autonomních buněk řízena prostřednictvím programu v PLC Simatic S7-1200. Tyto programy zajišťují logické řízení každé buňky, tedy zpracování signálů z připojených snímačů, ovládání akčních členů, implementace zpětnovazebních regulačních smyček, ochran, alarmů a chybových stavů. Dále je v PLC vytvořena struktura pasivního AAS, který zpřístupňuje technologické informace a výrobní metody.

8.2 Softwarové vybavení testbedu

Jako platforma PLM bude používán software Teamcenter společnosti Siemens. Ten je v projektu částečně použit již nyní, resp. jeho nástrojové portfolio Tecnomatix, které je používáno pro stavbu testbedu a jeho buněk od počátečního návrhu, přes výrobu dílů, virtuální prototypování, simulaci, ověřování a zprovoznění.

Integrace některého z existujících (komerčních či volně dostupných) PLM či MES systémů do platformy testbedu je záležitostí relativně novou. V raných etapách vývoje a provozu testbedu byla implementována vlastní platforma pro správu výrobních entit dle ISA-95. V nedávné době došlo k adaptaci platformy EspoCRM, která je primárně orientována spíše na CRM část ERP systémů, avšak nabízí možnosti dodatečné implementace dalších modulů. Na platformě byly spuštěny standardní moduly, jako *správa uživatelů systému*, *správa objednávek*, atd. Dále byly vytvořeny další moduly pro správu *materiálu*, *personálu*, *vybavení* a *procesních* a *produktových segmentů* dle standardu ISA-95. Pro tyto moduly byla rovněž vytvořena webová uživatelská aplikace, s jejíž pomocí je v současnosti možné provádět správu celého objednávkového a výrobního procesu [71, 49].

8.3 Typ výroby v chytré továrně

Pokud je testbed s názvem *Barman* modelem *chytré továrny*, jaký typ výroby v ní bude probíhat? Každý kontejner - skleničku, ve které se postupně vyrábí nápoj podle zadané receptury, lze chápat jako tank, ve kterém se vyrábí jedna dávka. Je to typicky dávková výroba s řízením typu *Batch*. Na druhé straně se sklenička při výrobě nápoje pohybuje v krocích po definované trajektorii a posléze po dopravníku, a v každém kroku je s ní provedena nějaká operace. To je typické pro diskrétní výrobu. Zde nicméně chybí základní prvek definice diskrétní výroby, a sice že při výrobě se mění fyzikálně měřitelné rozměry výrobku. I tak můžeme říci, že výroba nápoje je v tomto případě kombinací dávkové a diskrétní výroby. S otázkou k typu výroby naší chytré továrny souvisí také základní obchodní i produkční řízení testbedu. Zákazník musí nějakým způsobem sdělit svůj požadavek, a ten se musí přeměnit ve výrobní operace (workflow), které se nakonec v zařízení (testbedu) vyrobí. Budou tedy existovat tři klíčová rozhraní: *Zákazník - obchod*, *obchod - řízení výroby* a *řízení výroby - výroba*.

Výroba produktu začíná v katalogu produktů - zákaznické webové aplikaci. Tato aplikace (rozhraní *zákazník - obchod*) je navázána na obchodní databázi testbedu, ze které čerpá data o dostupných produktech. Jejich seznamy pak kategorizovaně zobrazuje, přičemž zákazník z nich může vybírat a objednávat (viz obrázek A.7). Objednávky jsou aplikací vkládány zpět do databáze. Potvrzení objednávky obsluhou může, ale nemusí být vyžadováno. Pokud vyžadováno není, jsou objednávky okamžitě zařazeny do výroby.

Celý systém je rovněž připraven na druhý způsob zadávání objednávek - tedy na jejich tvorbu jako nových receptur podle přání zákazníka. Ten by v tomto případě vyplnil objednávku prostřednictvím webové aplikace, navolil jednotlivé parametry výroby - materiály a vlastnosti finálního produktu.

Funkce rozhraní *obchod - řízení výroby* obecně zabezpečuje propojení modulů ERP a MES systémů, které v souladu se standardem ISA 95 komunikují prostřednictvím výměny standardizovaných (ISA 95) zpráv. Plánovací modul MES systému pak poskytuje prostředky a uživatelské rozhraní pro vytvoření operativního plánu výroby. Na základě tohoto plánu probíhá uvolňování zakázek do výroby.

Vzhledem k tomu, že MES, který testbed využívá, vznikl adaptací CRM řešení doplněného o implementaci některých klíčových modulů, je na platformě testbedu rozhraní *řízení výroby - výroba* zabezpečeno implementací vlastního softwarového modulu (tzv. *exekečního modulu*) složeného ze dvou částí. První, *plánovací část*, přebírá z obchodní části vytvořené objednávky, na základě kterých sestavuje postup výroby konkrétního produktu. Tento postup je tvořen skupinou *segmentů produktu* dle ISA 95, z nichž každý je složen ze specifikace materiálu a vybavení². Vytvořený výrobní postup každé objednávky je vložen do seznamu objednávek pro výrobu, kde je následně využit ve druhé - *výrobní části* exekečního modulu.

8.4 Centralizovaně řízená výroba

Klasická hierarchie řízení, popsaná v kapitole 3.1.1 je na spodních úrovních tvořena řídicím modulem MES systému a skupinou PLC pro řízení technologických procesů. Rozhraní mezi řízením výroby a výrobou je tedy reprezentováno komunikací výkonného modulu MES systému s PLC. Způsob komunikace mezi těmito dvěma úrovněmi není standardizován, jak je tomu o úroveň výše, ale obecně jsou využívány následující čtyři typy dat:

- informace o produktu - jedná se o soubor informací definujících vlastnosti vyráběného produktu, nezbytný pro realizaci výroby v konkrétním výrobním zařízení;
- povely pro výrobu - jedná se o příkazy, na základě kterých výrobní zařízení spouští, či přerušuje výrobu;

²Specifikace ISA 95 popisuje objekt *personál*, který však v rámci automatické výroby nevyužíváme

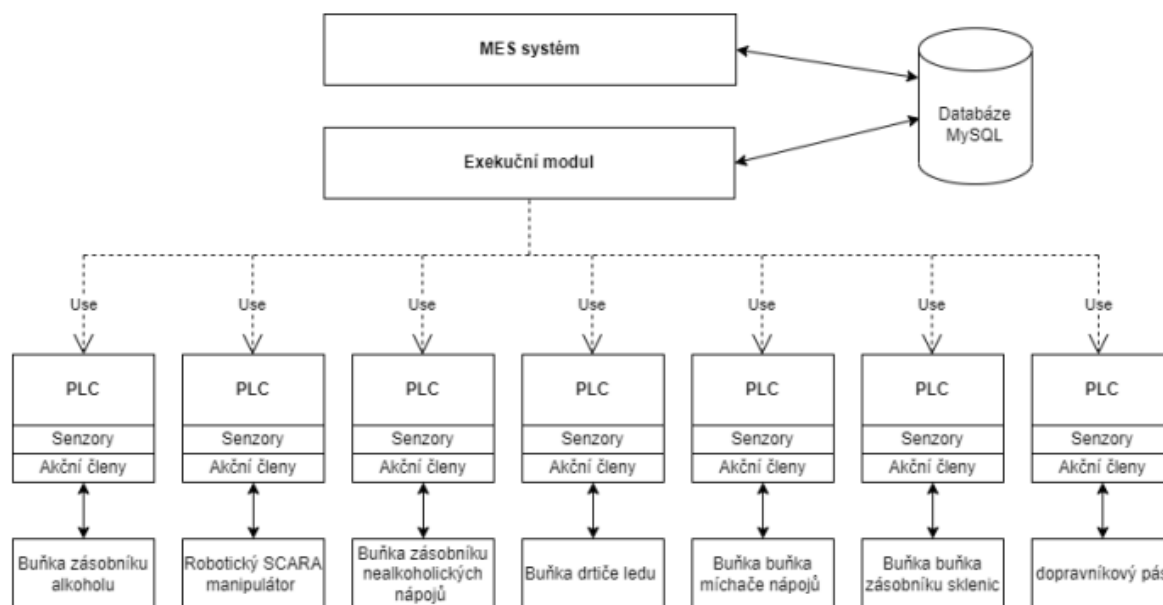
- stav výroby - představuje balík informací přenášených periodicky z výrobního zařízení, který obsahuje informace o stavu a rozpracovanosti aktuálního výrobního cyklu;
- informace o výrobě - jsou soubor informací přenášených jednorázově z výrobního zařízení po ukončení výroby obsahující podrobnosti o vyrobeném produktu (např. měřitelné informace o jeho kvalitě).

Při tomto způsobu řízení výroby testbedu komunikuje exekuční modul MES systému s PLC prostřednictvím proprietárního S7 protokolu, jehož specifikace je však volně dostupná a existuje mnoho implementací pro nejrůznější programovací jazyky. Výrobní část exekučního modulu uvolňuje do výroby prioritně vždy nejstarší zakázku (pořadí zakázek je tedy určeno pravidlem FCFS, viz kapitola 5.6.1). Uvolnění do výroby probíhá tak, že zakázka se rozdělí na jednotlivé segmenty a hledá se volný *procesní segment* (výrobní buňka), jehož výrobní schopnosti odpovídají požadavkům na výrobu prvního segmentu produktu. Pokud je buňka s odpovídajícími schopnostmi nalezena, exekuční modul provede její rezervaci a zajišťuje případný transport kontejneru. Jakmile je kontejner transportován do buňky, řídicí modul spustí výrobu. Po jejím dokončení se celý proces opakuje s těmito specifiky:

- První *výrobní* buňkou pro každý produkt je buňka skladu prázdných kontejnerů. Prvnímu kroku výroby tedy nepředchází transport kontejneru, neboť ten se právě z této buňky vyskladňuje.
- Výroba produktů je zřetězena, testbed tedy vyrábí více produktů najednou, pokud je to možné. Na implementační úrovni vytváří řídicí část exekučního modulu pro každý produkt abstrakci - *digitální stín* (viz kapitola 2.4.2), který udržuje kompletní výrobní informace o produktu a zrcadlí stav skutečného kontejneru. Exekuční modul pak může na základě těchto abstrakcí jednoduše přidělovat jednotlivé podúlohy buňkám. Algoritmus řízení testbedu využívá toho, že výrobu lze v naprosté většině případů popsat modelem Flow Shop (přestože obecně se jedná o model *Open shop*), a tedy nedochází k potížím s dostupností jednotlivých výrobních buněk a *uvíznutí* kontejnerů³.
- Hotový produkt je umístěn na dopravníkový pás. Část tohoto pásu leží mimo pracovní prostor robota, a může být přístupná obsluze, či rovnou zákazníkům. V systému je vytvořeno digitální dvojče dopravníkového pásu spolu s vizualizací, jejímž prostřednictvím je možné produkty jednoznačně identifikovat a přiřadit zákazníkům.

Přestože myšlenky centralizovaně řízené výroby sahají až hluboko do minulého století, má demonstrace základních principů, postupů a standardů používaných při centrálně řízené výrobě své nezastupitelné místo při výuce studentů.

³Pro eliminaci uvíznutí v případě Open shop modelu by testbed musel být doplněn odkládací plochou, která umožní vyměnit mezi sebou kontejnery ve dvou, či více buňkách za pomoci jednoho robotického ramene



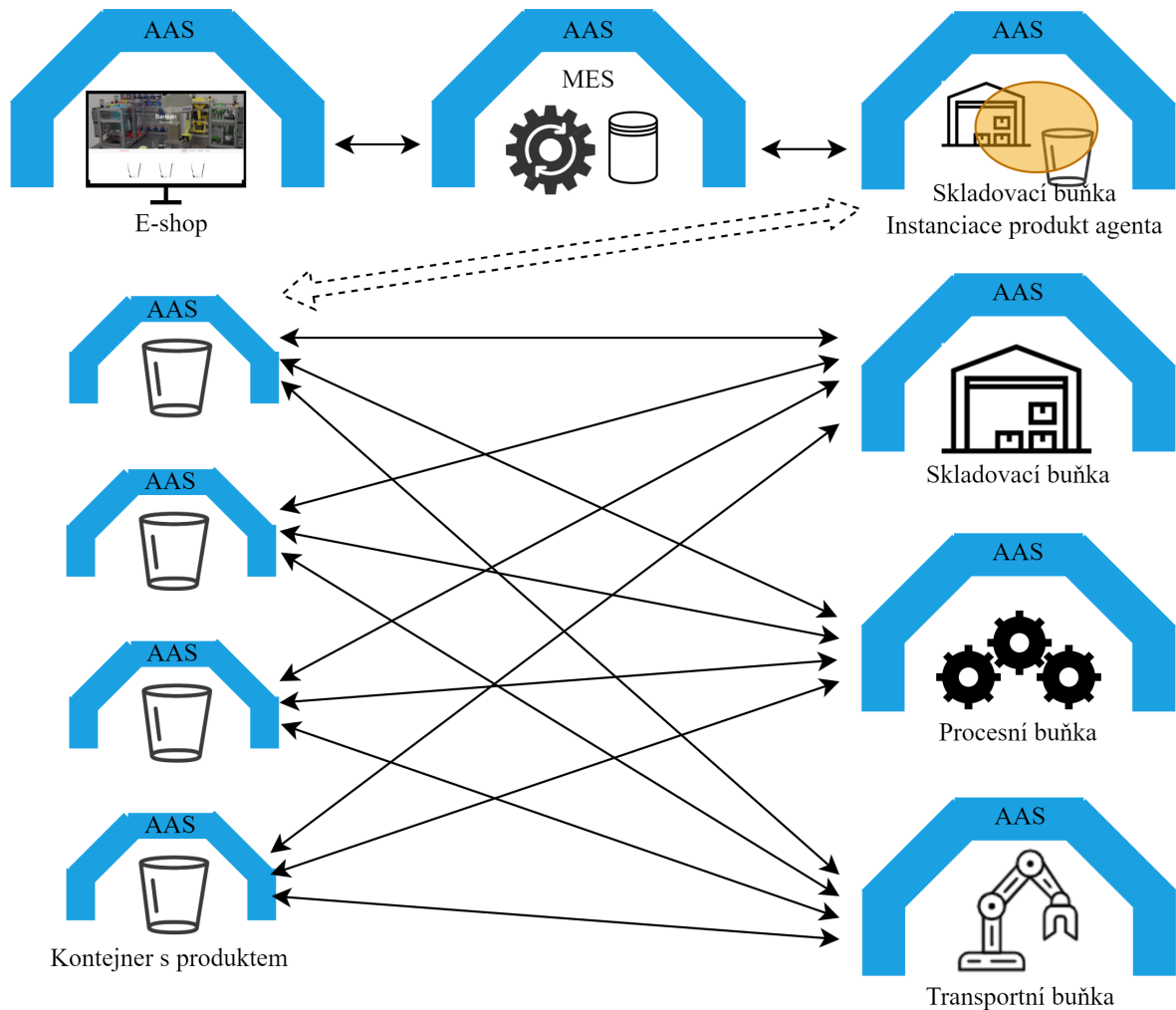
Obrázek 8.1: Hierarchie centralizovaně řízeného systému [71]

8.4.1 Distribuovaně realizovaná výroba

Jednotlivé výrobní buňky je možné považovat za autonomní kyber-fyzické výrobní systémy (viz kapitola 3.2.2), resp. za I4.0 komponenty (kapitola 4.2). Buňky jsou schopny vzájemné výměny informací prostřednictvím pasivních částí Asset Administration Shell rozhraní (kapitola 4.2.4, [129]). V rámci aktivní části AAS (kapitola 4.2.5) jednotlivé buňky vystupují jako poskytovatelé služeb (service provider), neboli jako *resource agenti* (kapitoly 4.2.5, 6.6, 7.2.1).

Vzhledem k tomu, že PLC, které řídí výrobu v buňkách nepodporují komunikaci prostřednictvím OPC-UA protokolu 4.6.2 nutnou pro výměnu dat AAS, je autonomní CPPS v buňce koncipován takto: centrálním členem každé buňky je miniaturní embedded systém s integrovanou čtečkou NFC (dále jen jako *čtečka*). V této čtečce je s pomocí knihovny *open62541* vytvořen AAS model a je zde implementován OPC-UA server pro horizontální komunikaci s dalšími entitami. Čtečka komunikuje s technologickým PLC prostřednictvím protokolu *Modbus TCP*, případně proprietárního, ale otevřeného protokolu *S7*. Tímto způsobem jsou implementovány CPPS ve všech výrobních i skladovacích buňkách, stejně jako CPPS, manipulátoru a pásového dopravníku.

Pro Průmysl 4.0 je klíčové to, že autonomní entity v rámci výrobního systému nejsou jen výrobní, či transportní stroje, dopravní pásy a jejich části, ale i výrobky, dávky vstupního materiálu či jednotlivé součásti. Při výrobě jsou jednotlivé vyráběné produkty zastupovány svými produkt agenty (kapitoly 4.2.5, 6.6, 7.2.1). V jejich případě je praktická implementace pro testbed komplikovanější, neboť pro běh software těchto agentů není explicitně vyčleněn žádný hardware.



Obrázek 8.2: Návrh AAS demonstrátoru

Vzhledem k povaze vyráběných produktů není možné opatřit samotné produkty jakýmkoli identifikátory. Je tedy nutno identifikovat alespoň kontejner, který slouží pro transport každého produktu. Každý kontejner má proto ve spodní části nalepen NFC tag, který obsahuje nejen jeho identifikátor, ale v datové části čipu je nahrán celý výrobní předpis pro produkt, včetně informace, v jakém stavu rozpracovanosti se produkt nachází. Tyto NFC tagy jsou zpracovány čtečkou vždy po umístění kontejneru do výrobní buňky (fyzicky je čtečka v buňce realizována jako podstavec, na který robot umístí a ze kterého vybírá kontejner [58, 48]). Po identifikaci kontejneru se v OS čtečky spouští vlákno produkt agenta, který je vybaven všemi stavovými informacemi vyčtenými z NFC. Na základě těchto informací má produkt agent schopnosti vyjednávat výrobní krok s:

- resource agentem výrobní buňky, do které byl právě vložen (oba agenti jsou v tuto chvíli spuštěni ve společném HW a OS, avšak vyjednávací proces to nijak neovlivňuje),
- resource agentem výrobní buňky, která bude realizovat příští výrobní krok nebo

- transportní entitou, která kontejner přepraví do následující buňky.

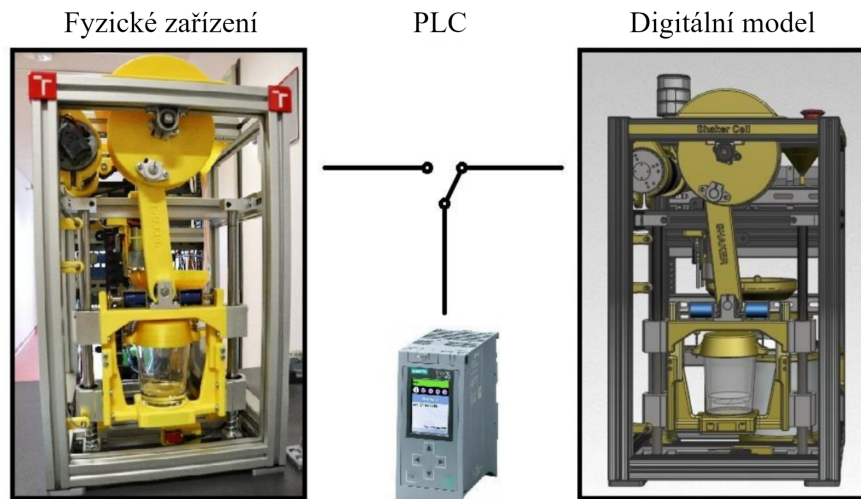
Vyjednávání probíhá prostřednictvím přenosu zpráv I4.0 jazyka (kapitola 4.5), vyjednávání proces je převzat ze specifikace Contract Net protokolu bez modifikací (kapitola 7.1.2). Tím, že je v NFC tagu uložen celý výrobní postup (nejen identifikátor), může proces výroby jednoho kroku, ale i celého produktu probíhat bez jakékoli vazby na nadřazený MES systém s jednou výjimkou: *někdo* musí předpis výroby do NFC tagu prázdného kontejneru umístit. Tuto operaci zastane výrobní část exekučního modulu, která má nyní podstatně jednodušší situaci ve srovnání s centralizovaným řízením - musí pouze stanovit, která z čekajících zakázek bude uvolněna do výroby, přičemž opět využívá pravidlo FCFS. Při uvolňování do výroby předá exekuční modul kompletní balík informací o budoucím produktu buňce skladu kontejnerů, která provede jednak vyskladnění *prázdného* (fyzicky i datově) kontejneru na místo, ve kterém je přítomna čtečka, jednak nahrání dat do NFC čipu kontejneru (instanciace produkt agenta je znázorněna na obrázku 8.2). Po dokončení instanciace dojde ke zpětnému vyčtení dat z NFC a spuštění produkt agenta nově vznikajícího výrobku. Ten postupuje ve vyjednávacím procesu v souladu s výše uvedeným CNP protokolem.

8.4.2 Virtuální zprovoznění

Modely výrobních buněk byly vytvořeny s pomocí nástroje SIEMENS NX a pro definici jejich kinematiky byl využit zabudovaný modul Mechatronics Concept Designer. Tento modul umožňuje propojení se skutečným, či simulovaným řídicím systémem prostřednictvím protokolu OPC UA. Obrázek 8.3 nabízí srovnání fyzické a virtuální buňky. Prvotní ověření funkce modelů jednotlivých buněk testbedu probíhalo metodou *Software in the Loop*. Digitální model byl řízen prostřednictvím virtuálního řídicího systému PLCSIM Advanced [45].

V prvním kroku proběhly testy spojení a mapování jednotlivých vstupů a výstupů, které ověřily funkčnost komunikace a správnost mapování virtuálních snímačů a akčních členů. Další testy byly zaměřeny na ověření simulovaných fyzikálních vlastností celého systému. Nástroj PLCSIM Advanced bohužel nedokáže virtualizovat konkrétní řadu PLC použitých pro řízení fyzických buněk, proto bylo nutné přejít k simulaci *Hardware in the Loop*. Dalším problémem byla nemožnost přímo provozovat OPC UA server na fyzickém PLC S7-1200. S tímto problémem se však počítalo od začátku vývoje konceptu buněk a OPC-UA server (realizace aktivní i pasivní části AAS) byl implementován jako samostatný software pomocí knihovny Open62541 a spuštěn ve čtečce [65]. Komunikace čtečky s technologickým PLC je, jak již bylo uvedeno, realizována S7 protokolem, pomocí kterého lze jednoduše synchronizovat konkrétní funkční bloky PLC, které jsou na straně čtečky (AAS) mapovány jako submodely. Detailní informace týkající se virtuálního zprovoznění buněk jsou dostupné v [45].

Základní myšlenky virtuálního zprovoznění popsané v kapitole byly naplněny a díky jeho využití se podařilo ověřit digitální návrhy buněk ještě před jejich realizací, a



Obrázek 8.3: Schéma konceptu testování



Obrázek 8.4: Marker pro detekci pracovní plochy testbedu [83]

zároveň bylo možné odstranit veškeré chyby v řídicích programech. Propojené virtuální modely byly navíc úspěšně využity v rámci praktické výuky.

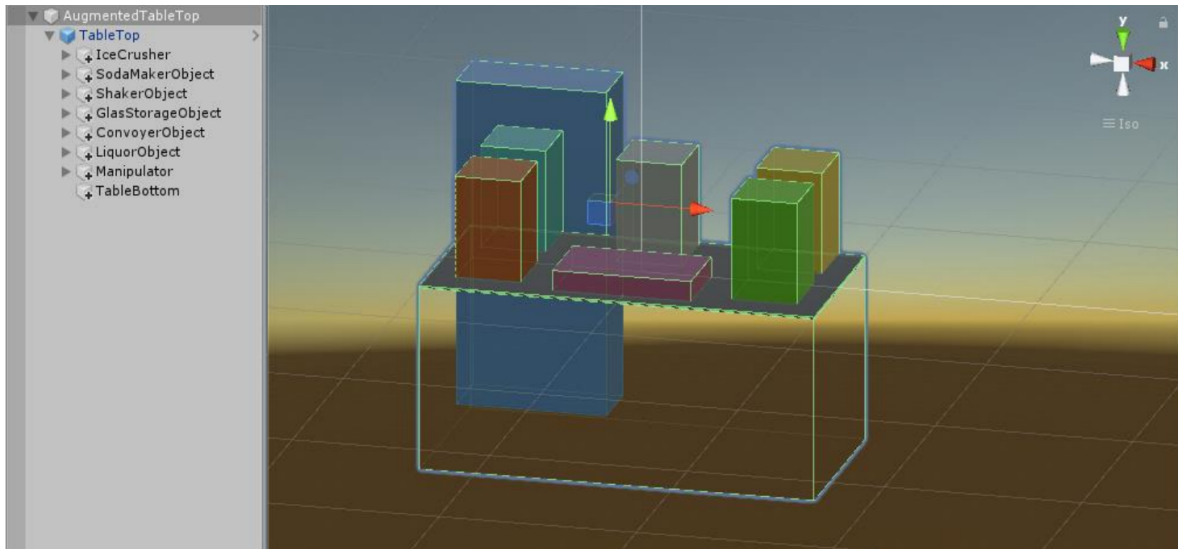
8.4.3 Rozšířená realita

Integrace rozšířené reality do testbedu je založena na principu *marker-based AR* spolupracujícího s informačním systémem testbedu. Jako terminál pro rozšířenou realitu lze využít jakékoli zařízení s operačním systémem AndroidOS, na němž lze spustit aplikaci využívající framework ARCore [83].

AR aplikace je založena na principu marker-based s využitím frameworku ARCore. K integraci AR musí být objekt (testbed) označen pomocí markeru, prostřednictvím kterého se algoritmus aplikace orientuje v prostoru reálné scény a získá informace potřebné pro správnou funkci rozšířené reality (pozice, rotace, měřítko a vzdálenost plochy vůči snímacímu zařízení). Jako marker, jehož pozice na pracovní ploše je přizpůsobitelná, byl vytvořen QR kód (viz obrázek 8.4).

Pro možnost správného vykreslení informací rozšířené reality byl vytvořen jedno-

duchý virtuální 3D model se všemi buňkami, který byl následně využit v AR aplikaci. Po lokalizaci testbedu v prostoru je tento model virtuálně mapován do souřadnic, na kterých fyzický testbed leží. Model se skládá z jednoduchých geometrických objektů (viz obrázek 8.5), které zastupují jednotlivé fyzické buňky. Vzhledem k tomu, že pozice jednotlivých buněk na platformě je možné libovolně měnit, také aplikace rozšířené reality při svém startu ze serveru získává datový balíček, jehož součástí je aktuální konfigurace továrny. Jednotlivé objekty ve virtuálním modelu slouží k vymezení prostoru, který zaujímá jejich fyzický protějšek. Pokud uživatel skrze aplikaci *klikne* na objekt reálného testbedu, je možné tuto akci detekovat prostřednictvím entit modelu.



Obrázek 8.5: Virtuální model testbedu v prostředí Unity [83]

Po kliknutí na fyzickou buňku ve scéně je vyvolána uživatelská akce - zobrazení informační tabule. Ta je tvořena dvěma sekcemi - panelem s názvem buňky s texty, prezentujícími data o stavu a tlačítka k ovládání. První část - prezentační - obsahuje data o stavu buňky. Vzhledem k požadavku na univerzálnost jsou všechny prezentované informace stahovány ze serveru jako formátovaný text. Stejně tak definice tlačítek a oznámení o jejich stisknutí je přímo komunikováno s REST API serverem. Zobrazená data jsou tedy kompletně řízena serverem, aby ani v případě provedení podstatných změn v technologii nebyl nutný zásah do uživatelské aplikace. Bližší informace o systému rozšířené reality jsou dostupné z [83].

8.5 Shrnutí kapitoly

Testbed v současné podobě představuje nástroj, jehož prostřednictvím jsou ověřovány a prezentovány základní principy výroby, a to jak standardně centralizovaně řízené, tak i moderně pojatého distribuovaného konceptu. Vzhledem k tomu, že se jedná o fyzický systém, je možné jeho funkčnost cílové skupině studentů snadno prezentovat a v průběhu let se prokázalo, že míra jejich pochopení problematiky je při využití testbedu vyšší, než bez něj. Informace uvedené v této kapitole jsou pouze přehledové.

Detailně je problematika týkající se testbedu popsána v závěrečných pracích, na které se tato kapitola odkazuje.

Kapitola 9

Závěr

Cílem práce bylo poskytnout v ucelené formě informace týkající se témat souvisejících se čtvrtou průmyslovou revolucí se zaměřením na problematiku *Digitálních dvojčat*. Motivací pro její vytvoření byla absence materiálu zahrnujícího všechna potřebná témata, který by bylo možné využít jako zdroj informací při výuce předmětu Průmysl 4.0 na Ústavu automatizace a měřicí techniky Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně.

Tato práce je dělena na tři oblasti, přičemž první je možné shrnout pojmem *Digitálně podporovaná výroba*. Ústřední problematikou je teorie životního cyklu produktů ve výrobním podniku a možnosti softwarových systémů pro jeho řízení. Data z těchto systémů tvoří základ kompletní digitální reprezentace produktu v podniku a v dnešní době slouží jako podklad k vytváření různých typů modelů a k následným simulacím. Produktem (jen jiného výrobce) je také výrobní zařízení, a proto je problematice simulace a virtuálnímu zprovoznění výrobních strojů, linek, ale také celých provozů věnována část práce. Pokud je digitální model vytvořen v rané fázi životního cyklu produktu a zůstává dále neměnný, možnosti jeho využívání se snižují. V případě, kdy je model v průběhu celého životního cyklu aktualizován s využitím skutečně měřených dat a jeho výstupy jsou využity pro změnu parametrů produktu, stává se digitálním dvojčetem a pracuje současně s fyzickým dvojčetem v rámci kyber-fyzického systému. V práci jsou popsány možnosti využití softwarových nástrojů z portfolia firmy Siemens pro návrh digitálních modelů a dvojčat v jednotlivých kategoriích produktů. Zatímco modelování vlastností výrobků je vhodné založit na konstrukčních modelech NX, pro výrobní stroje je vhodné využít spolupráci jeho doplňku Mechatronics Concept Designer a programu SIMIT. Doménou platformy Tecnomatix je simulace robotických výrobních linek, respektive kompletních výrobních procesů.

Druhá oblast této práce popisuje transformaci od tradičních metod a konceptů k Průmyslu 4.0, která probíhá od roku 2011. Systémy pro správu životního cyklu výrobku, modelovací nástroje, simulace a virtuální zprovoznění tvořily jeden z důležitých spouštěčů této transformace. Dalšími spouštěči pak byly kyber-fyzické systémy, internet věcí, dat a služeb a mnoho dalších technologických inovací v oblastech sběru, přenosu

a zpracování dat. V této části práce jsou také popsány tři hlavní směry průmyslové integrace (totiž vertikální integrace, horizontální integrace a integrace inženýrských procesů) ve stavu před a po transformaci k Průmyslu 4.0. Kyber-fyzické systémy, představené v první části, jsou dále rozvíjeny ve formě I4.0 komponent - praktické implementace, která sjednocuje heterogenní prostředí průmyslových zařízení různých výrobců tím, že předepisuje existenci tzv. Asset Administration Shell - rozhraní pro výměnu dat založené na submodelech. K tomuto rozhraní komponenty (např. stroje) se mohou ostatní entity připojit např. komunikačním protokolem OPC-UA, jehož popis má své místo v této části. Pasivní AAS je možné doplnit aktivním, které přináší možnost autonomního chování a rozhodování každé komponenty. Aktivním AAS může být vybaven např. výrobní stroj, který může vystupovat v roli *Service Provider*, tedy poskytovatel určité (výrobní) služby. V průmyslu 4.0 může být samotný vyráběný produkt také účastníkem komunikace - vyjednávání. V takovém případě vystupuje v roli *Service Requester* a (výrobní) služby poptává. Zbývá už jen určit, jakým způsobem bude vyjednávání mezi oběma stranami probíhat - jaká budou zařízení a produkty dodržovat pravidla, a jakým komunikačním protokolem budou *mluvit*. Aktivní AAS je předmětem výzkumu na našem pracovišti. Jako jedni z prvních jsme začali teoretické myšlenky testovat a dále rozvíjet v praxi s využitím testbedu a to mimo jiné i v rámci česko-německého projektu RACAS¹ ve spolupráci s průmyslovým partnerem *Compas automatizace, spol. s r.o.*

Vyjednávání procesu výroby mezi skupinou autonomních zařízení je jednou z možností, jak sestavit plán výrobních operací, ne však jedinou, a proto je třetí oblast práce zaměřena na popis problematiky plánování výroby. V centralizovaných architekturách s ústředním řídicím členem jsou typicky nasazovány centralizované metody plánování, které se dělí na deterministické a stochastické. Propojení jednotlivých poskytovatelů a konzumentů služeb jako I4.0 komponent však není vertikálně hierarchické a postrádá řídicí člen. Místo toho se uplatňuje heterarchická architektura a horizontální propojení. Algoritmy pro tyto architektury bez vertikální hierarchie jsou obecně založeny na bázi agentních systémů, v nichž jednotliví agenti v systému autonomně interagují s ostatními agenty a sledují splnění svých cílů. Tato práce cílí na algoritmy vzájemné kooperace, proto je zde detailně popsán Contract Net Protocol a jeho varianty, které jsou základem pro návrh distribuovaného vyjednávání I4.0 komponent definovaných také ve specifikaci *VDI2193 - Sprache für I4.0-Komponenten*. V rámci implementace protokolu v testovacím systému bylo navrženo rozšíření protokolu, které přináší heuristiku vkládání zakázek do rezervačních front jednotlivých agentů strojů. Tato heuristika pomáhá naplňovat Just-in-Time princip. Testováním a simulacemi efektivity algoritmu se ukázalo, že bez dalších úprav není vhodný pro modely výroby typu Job Shop, ale lze jej bez obtíží využít na rozvrhování úloh typu Flow Shop.

Hlavní koncepty všech tří oblastí, do kterých míří tato práce, sjednocuje testbed Průmyslu 4.0, který byl vytvořen jako projekt spolupráce se studenty v rámci závěrečných prací. Tento testbed, nacházející se v laboratořích na Ústavu automa-

¹Digital Representation of Assets with Configurable AAS for CPP-Systems

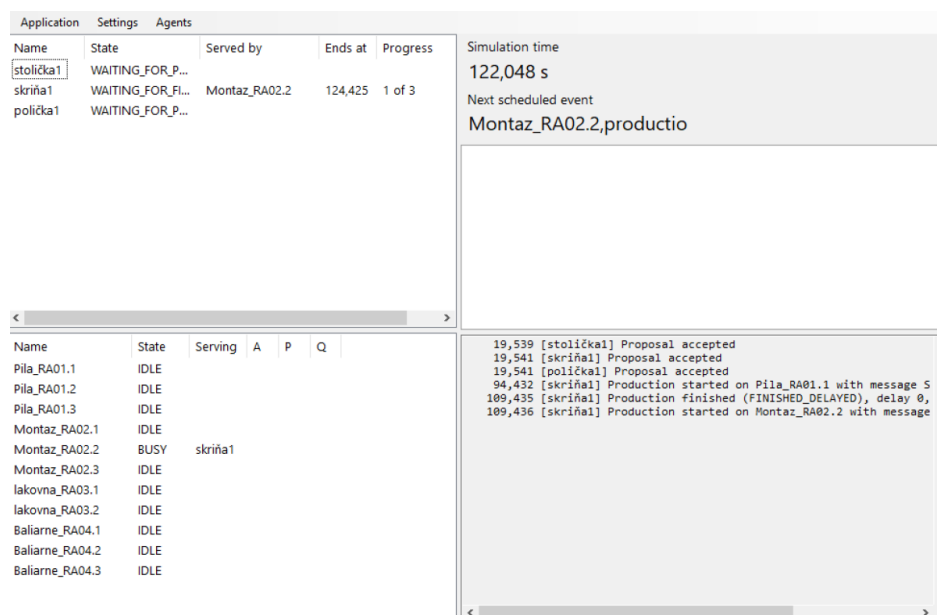
tizace a měřicí techniky má své nezastupitelné místo při výuce témat souvisejících s Průmyslem 4.0. Vzhledem k tomu, že tato témata jsou často interdisciplinární s přesahem do strojního inženýrství a informatiky, je pro studenty velice užitečné mít možnost demonstrovat hlavní problémy a jejich možná řešení na fyzickém zařízení. Kromě možnosti realizovat praktickou výrobu (centralizovaně řízenou i distribuovaně vykonávanou) na základě různých řídicích algoritmů testbed umožňuje testování i dalších zmiňovaných konceptů, např. virtuálního zprovoznění, rozšířené reality. Celá oblast se dynamicky rozvíjí, testbed je průběžně inovován a do budoucna se jedná o cennou platformu podporující přenos výsledků teoretického výzkumu do praxe.

Věřím, že tato práce, stejně jako testbed, usnadní studentům nejen přístup k informacím ale, podá jim základní přehled v problematice Průmyslu 4.0 a digitálních dvojčat a přispěje k dalšímu rozvoji tohoto tématu nejen v rámci vyučovaného předmětu, ale i celkově na Ústavu automatizace a měřicí techniky.

Příloha A

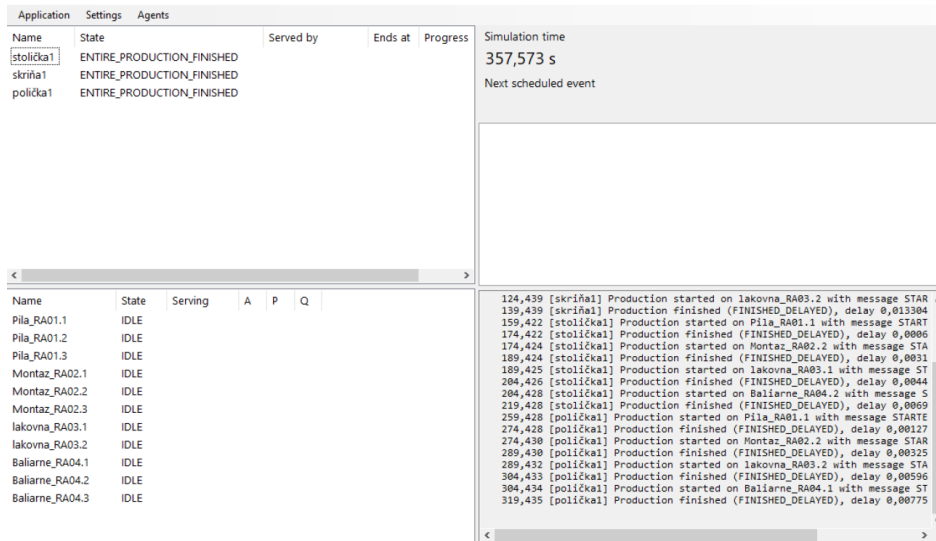
Popis implementace simulačního prostředí

Pro implementaci simulátoru CPPS agentního prostředí byl zvolen .NET framework a jazyk C#. Platforma byla vybrána pro svou otevřenost, nezávislost na licenčních omezeních programů třetích stran a do vysoké míry platformní nezávislost. Záměrem bylo implementačně se přiblížit aktuálním řešením a zjednodušit možnou integraci algoritmu např. do funkčních bloků a knihoven systémů průmyslových zařízení. Na obrázcích A.1 a A.2 je prezentováno okno simulátoru po startu a po dokončení modelového případu plánování (viz kapitola 7.3.1).

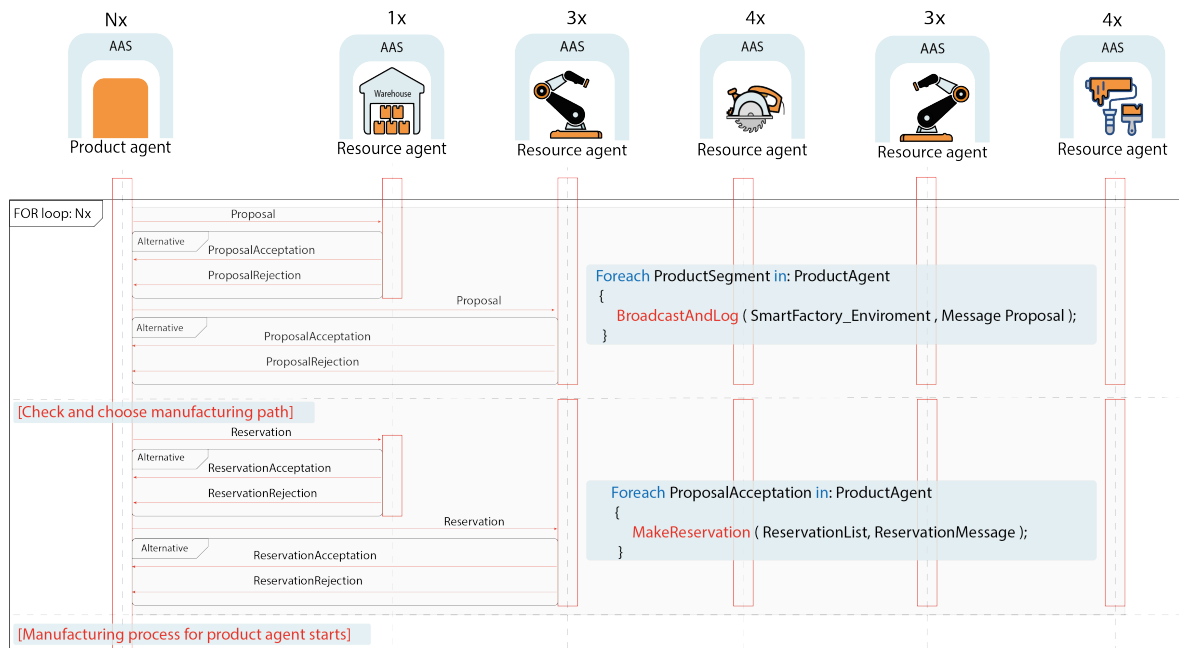


Obrázek A.1: Grafické okno simulátoru - počátek simulované výroby

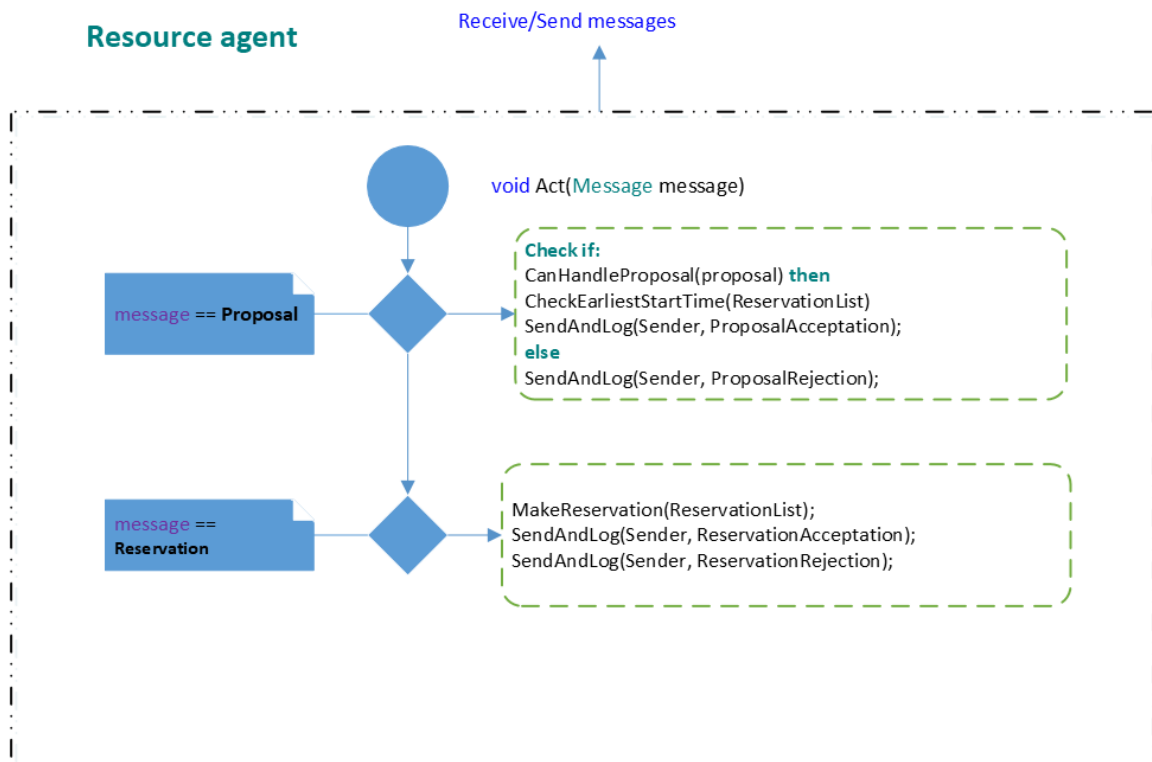
Na těchto stranách je stručně uvedena struktura implementace software simulátoru v podobě UML sekvenčních diagramů a diagramů tříd. Simulátor využívá CNP protokol s potvrzováním, resource agenti mohou odpovídat na výzvy produkt agenta (posílat nabídky) a teprve po potvrzení z jeho strany mohou alokovat časové sloty. Na obrázcích A.3, A.4, A.5 a A.6 je detailněji znázorněna struktura interakcí mezi agenty pomocí CNP protokolu. Vysvětlení k jednotlivým diagramům je uvedeno v textu práce.



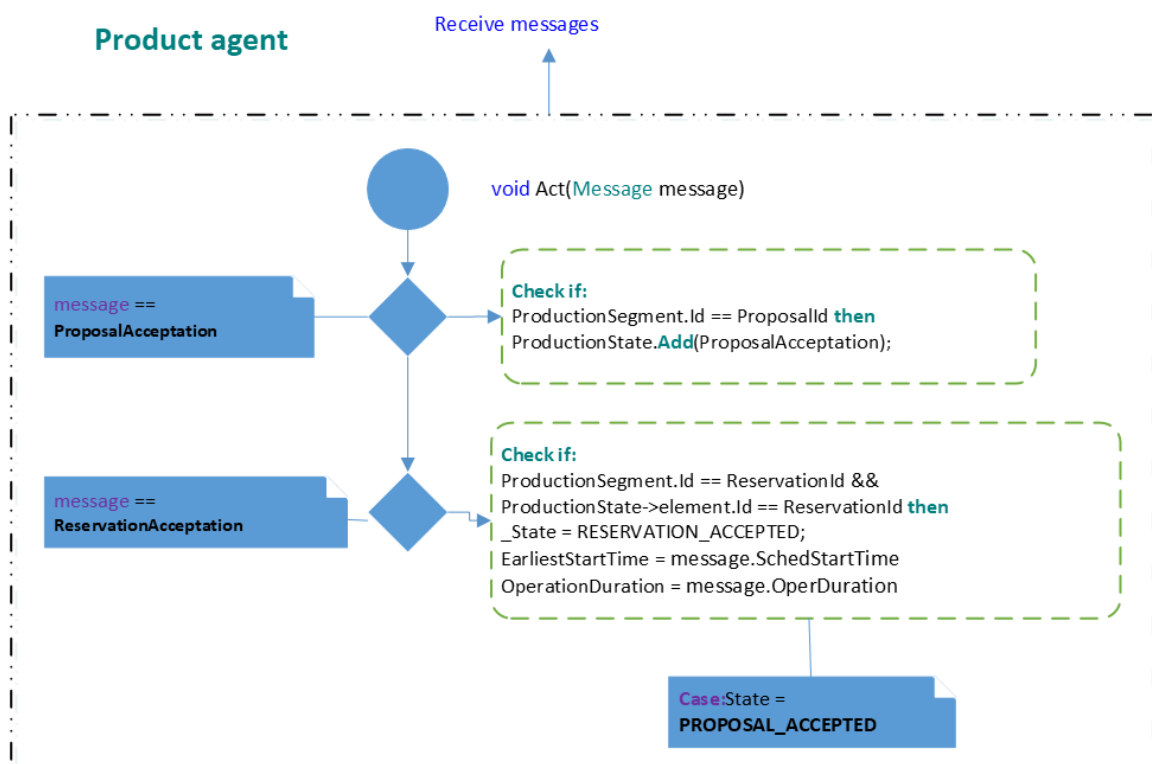
Obrázek A.2: Grafické okno simulátoru - konec simulované výroby



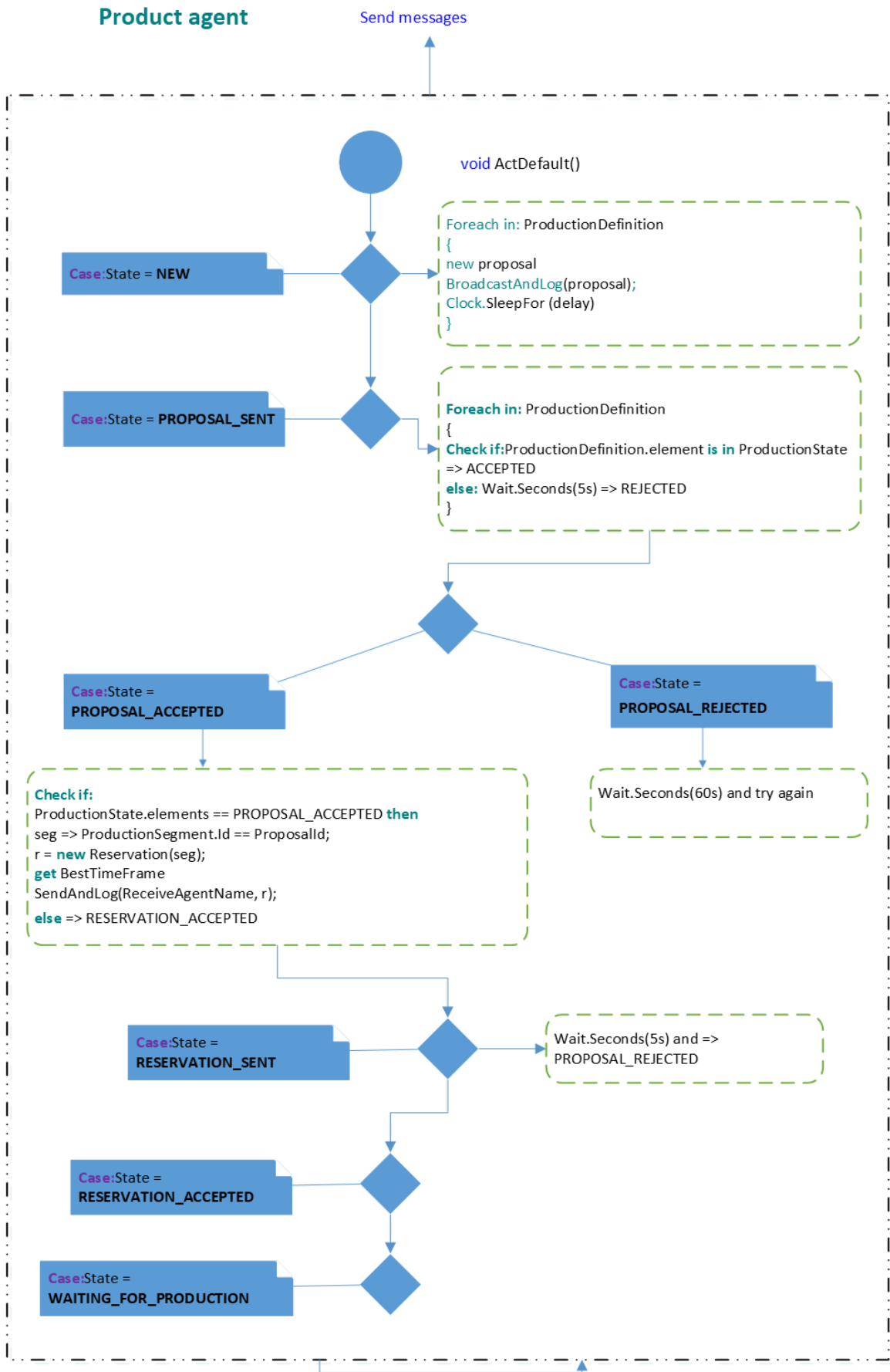
Obrázek A.3: Průběh komunikace mezi agenty (CNP)



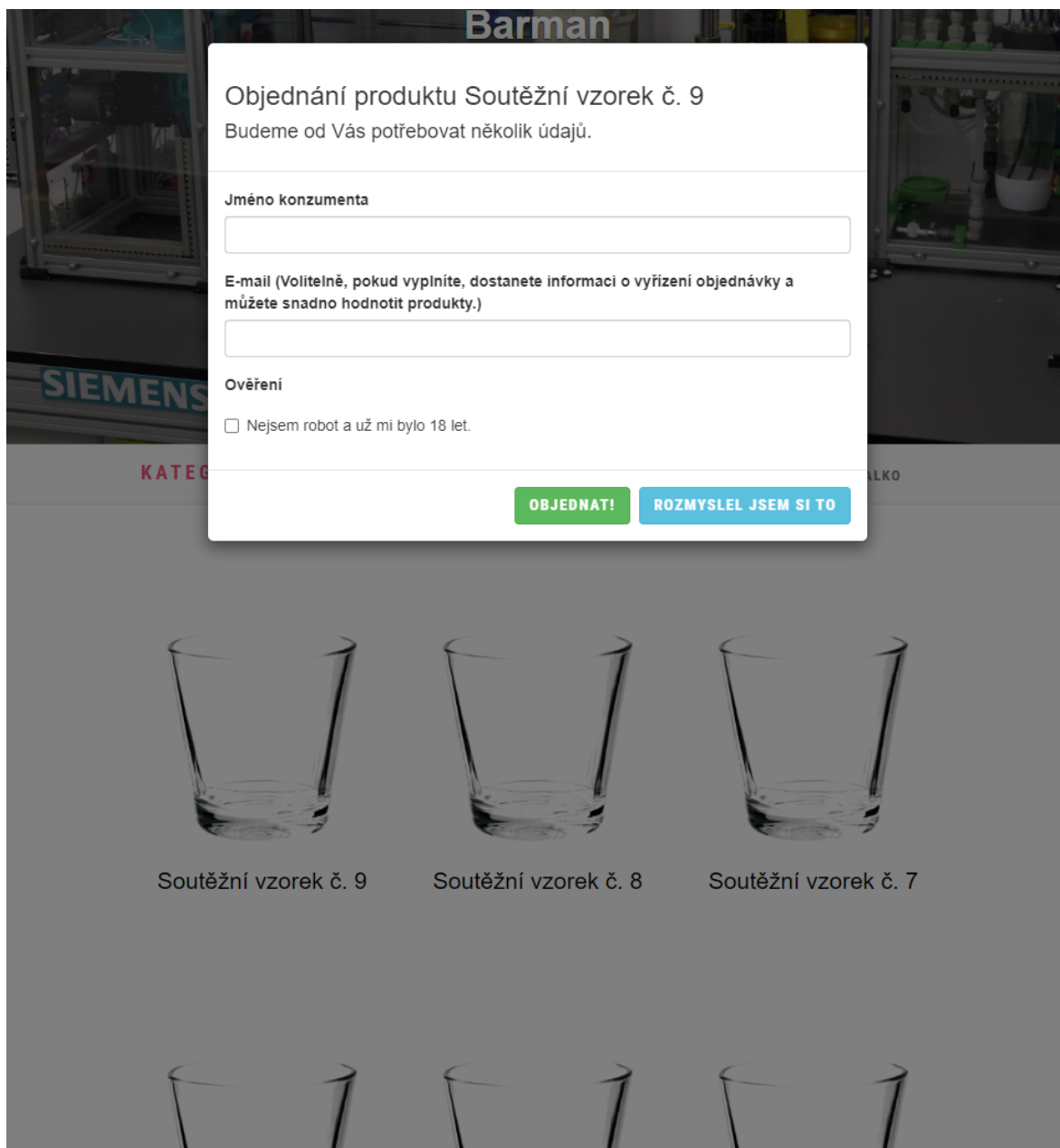
Obrázek A.4: Průběh rezervace na straně RA (příjem zpráv)



Obrázek A.5: Rezervace na straně PA (příjem zpráv)



Obrázek A.6: Rezervace na straně PA (odesílání zpráv)



Obrázek A.7: Testbed - rozhraní zákazník/obchod

Literatura

- [1] Platform Industrie 4.0. The structure of the administration shell: trilateral perspective from france, italy and germany. Technical report, Platform Industrie 4.0, 2018. online, <https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/hm-2018-trilaterale-coop.html>, cit. 6.7.2022.
- [2] Platform Industrie 4.0. What is the asset administrationshell from a technical perspective?, 2021. online, https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/2021_What-is-the-AAS.html, cit. 1.3.2023.
- [3] Acatech. *Cyber-Physical Systems*. Springer Berlin, Heidelberg, 1 edition, 2011.
- [4] Elif Gorgulu; Selim Akturk. Match-up scheduling under a machine breakdown. *European Journal of Operational Research*, 1999.
- [5] Klára Anyalaiová. Životní cyklus výrobku a jeho vliv na cenovoustrategii podniku. Master's thesis, Západočeská univerzita v Plzni, 2017.
- [6] Václav Kaczmarczyk; Ondřej Baštán; Zdeněk Bradáč; Jakub Arm. An industry 4.0 testbed (self-acting barman): Principles and design. *IFAC-PapersOnLine (ELSEVIER)*, 2018.
- [7] Philip Kotler; Gary Armstrong. *Marketing*. Grada, 2003.
- [8] Tom Austin-Morgan. Maserati has fused cutting-edge digitalisation methods with italian passion to meet customer demand. *Eureka*, 2017. online, <https://www.eurekamagazine.co.uk/content/interview/maserati-has-fused-cutting-edge-digitalisation-methods-with-italian-passion-to-meet-customer-demand/>, cit. 15.8.2022.
- [9] Jan Bajcar. Reflexe potřeb průmyslu 4.0 v managementu měst České republiky. Master's thesis, ČVUT v Praze, 2020.
- [10] Tung dang Baltazar Frankovič, Ivana Budinská. Agents coalition in coordination process. *IFAC Proceedings Volumes*, 2002.

- [11] Giacomo Barbieri, Alberto Bertuzzi, Andrea Capriotti, Lorenzo Ragazzini, David Gutierrez, Elisa Negri, and Luca Fumagalli. A virtual commissioning based methodology to integrate digital twins into manufacturing systems. *Production Engineering*, 03 2021.
- [12] Aidan Fuller; Zhong Fan; Charles Day; Chris Barlow. Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, 8:108952–108971, 2020.
- [13] Leo Barteveyan. Industrie 4.0 - kompaktwissen. online, 2016. online, <https://www.dlg.org/de/lebensmittel/themen/publikationen/expertenwissen-lebensmitteltechnologie/industrie-40>, cit. 10.9.2022.
- [14] Ilya Kovalenko; Daria Ryashentseva; Birgit Vogel-Heuser; Dawn Tilbury; Kira Barton. Dynamic resource task negotiation to enable product agent exploration in multi-agent manufacturing systems. *IEEE ROBOTICS AND AUTOMATION LETTERS*, 4(3), July 2019.
- [15] Martin Baumruk. Digitální továrna a ergonomické analýzy v automobilovém průmyslu. *MM Průmyslové spektrum*, 2012. online, <https://www.mmspektrum.com/clanek/digitalni-tovarna-a-ergonomicke-analyzy-v-automobilovem-prumyslu>, cit. 11.10.2022.
- [16] Tomáš Beneš, Václav Kaczmarczyk, Tomáš Sýkora, Jakub Arm, Petr Dvorský, Michal Husák, Petr Marčoň, and Zdeněk Bradáč. Asset administration shell - manufacturing processes energy optimization. *IFAC-PapersOnLine*, 55(4):334–339, 2022. 17th IFAC Conference on Programmable Devices and Embedded Systems PDES 2022 — Sarajevo, Bosnia and Herzegovina, 17-19 May 2022.
- [17] Michael Hoffmeister; Andreas Orzelski; Marco Mendes; Jonas Grote; Denis Göllner; Sebastian Bischof. Aasx package explorer: Github repository, 2023. online, <https://github.com/admin-shell-io/aasx-package-explorer>.
- [18] Jan Boch. Autonomní sklad kapalin s chladicím systémem. Master's thesis, Vysoké učení technické v Brně, 2022.
- [19] Jan Boucník. Chytrá města kolem nás. *Technický týdeník*, November 2017. online, https://www.technickytydenik.cz/rubriky/archiv/chytra-mesta-kolem-nas_42568.html, cit. 3.12.2022.
- [20] Edward H. Bowman. The schedule-sequencing problem. *Operations Research*, 1959.
- [21] Peter Brucker. *Scheduling algorithms*. Springer Berlin; New York, 2004.
- [22] David Buchal. Virtuální model stroje tecnomatix process simulate. Master's thesis, Vysoké učení technické v Brně, 2018.
- [23] Jaroslav Burian. *Geoinformatika v prostorovém plánování*. Univerzita Palackého v Olomouci, 2014.

- [24] Pavel Burian. Multiagentní systémy příčinou vyšší konkurenceschopnosti výrobních a logistických podnikových procesů. *AT&P journal*, 2006.
- [25] Lucie Byrtusová. Výrobní buňka pro testbed průmyslu 4.0. Master's thesis, Vysoké učení technické v Brně, 2019.
- [26] Andrea Cejnarová. Smart city index - hodnocení chytrých strategií měst. Technický týdeník, July 2017. online, https://www.technickytydenik.cz/rubriky/archiv/smart-city-index-hodnoceni-chytrych-strategii-mest_42512.html, cit. 6.7.2022.
- [27] Panasonic Automotive Systems Czech. O společnosti, 2023. online, <https://www.panasonic.com/cz/tovary-v-cr/o-spolecnosti.html>, cit. 3.5.2023.
- [28] Matthew DeBord. The history of car brands in the us. *Business Insider*, 2014.
- [29] Alexander Belayev; Christian Diedrich. Specification "demonstrator i4.0 language" v3.0. Technical report, Fakultat für Elektrotechnik und Informationstechnik, Universität Magdeburg, 2019.
- [30] Neil A. Duffie. Synthesis of heterarchical manufacturing systems. *Computers in Industry*, 14(1):167–174, 1990. Special Issue Josef Hartvany Memorial.
- [31] Steffen Heinrich; Holger Durr. An agent-based manufacturing management system for production and logistics within cross-company regional and national production networks. *International Journal of Advanced Robotic Systems*, 2005.
- [32] Petr Dvorský. Návrh, konstrukce a programové vybavení autonomní buňky so-
dovač pro testbed průmyslu 4.0. Master's thesis, Vysoké učení technické v Brně, 2019.
- [33] ECLASS. Eclass klasifikace, 2023. online, <https://eclass.eu/en/eclass-standard/search-content>, cit. 3.4.2023.
- [34] ECLASS. Eclass standard, 2023. online, <https://eclass.eu/>, cit. 2.4.2023.
- [35] German Electrical and Electronic Manufacturers' Association. Examples of the assetadministration shell forindustrie 4.0 components – basic part, April 2017.
- [36] German Electrical and Electronic Manufacturers' Association. Details of the assetadministration shell. Online, November 2019.
- [37] OPC Foundation. What is opc, 2023. online, <https://opcfoundation.org/about/what-is-opc/>, cit. 27.6.2023.
- [38] Daniel Rossit; Fernando Tohmé; Mariano Frutos. Production planning and scheduling in cyber-physical production systems. *International Journal of Computer Integrated Manufacturing*, 2023.

- [39] Alasdair Gilchrist. *INDUSTRY 4.0, The Industrial Internet of Things*. apress, 2016.
- [40] Michael Grieves. Origins of the digital twin concept. Working paper, August 2016.
- [41] Tpv Group. Vlastnosti pdm/plm teamcenter. online, <https://www.tpvgroup.cz/10-duvodu-pro-teamcenter/>, cit. 15.7.2022, May 2021.
- [42] HiveMQ. *MQTT & MQTT 5 Essentials*. 84030 Landshut Germany: Ergoldinger Str. 2A, 2022.
- [43] Jan Holba. Implementace skladovací entity v s7-1200. Master's thesis, Vysoké učení technické v Brně, 2023.
- [44] Lukáš Horák. Návrh, konstrukce a programové vybavení autonomní buňky drtič ledu pro testbed průmyslu 4.0. Master's thesis, Vysoké učení technické v Brně, 2019.
- [45] Michal Husák. Virtuální dvojče pro testbed průmyslu 4.0. Master's thesis, Vysoké učení technické v Brně, 2020.
- [46] Dagmar Jakubíková. *Strategický marketing*. Grada, 2008.
- [47] Hana Janišová. Zjednodušuje práci, zkvalitňuje práci, zrychluje práci. *MM Průmyslové spektrum*, 2018. online, <https://www.mmspektrum.com/clanek/zjednodusuje-praci-zkvalitnuje-praci-zrychluje-praci>, cit. 20.7.2023.
- [48] František Jirásek. Návrh a realizace univerzální nfc čtečky. Master's thesis, Vysoké učení technické v Brně, 2022.
- [49] Tomáš Martiník;Václav Kaczmarczyk. Manufacturing system for industry 4.0 demonstrator. In *Proceedings II of the 29th student EEICT 2023*, 2023.
- [50] Václav Kaczmarczyk, František Zezulka, Tomáš Beneš, Jakub Arm, Petr Marcoň, Jan Jirsa, and Lukáš Venkrbec. Revisiting the role of manufacturing execution systems in industry 4.0. *IFAC-PapersOnLine*, 55(4):151–157, 2022. 17th IFAC Conference on Programmable Devices and Embedded Systems PDES 2022 — Sarajevo, Bosnia and Herzegovina, 17-19 May 2022.
- [51] Radim Karniš. Návrh, konstrukce a programové vybavení autonomní buňky shaker pro testbed průmyslu 4.0. Master's thesis, Vysoké učení technické v Brně, 2019.
- [52] František Koblasa. *Uplatnění heuristických optimalizačních metod v oblasti rozvrhování strojírenské výroby středních a malých podniků*. PhD thesis, Technická univerzita v Liberci, 2013.

- [53] František Manlig; František Koblasa. Plánování a rozvrhování výroby. online, https://www.unipranet.zcu.cz/doc/15_Planovani%20a%20rozvrhovani%20vyroby%20Manlig%20&%20Koblasa%20TUL.pdf, cit. 3.2.2022, 2015.
- [54] E Kondil, Constantinos Pantelides, and Roger Sargent. A general algorithm for scheduling batch operations. 08 1988.
- [55] Sarit Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 1997.
- [56] Wen-Yang Ku and J. Christopher Beck. Mixed integer programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*, 73:165–173, 2016.
- [57] Aleš Kubík. *Inteligentní agenty*. Computer Press, 2004.
- [58] Richard Kubíček. Komunikační hardware pro i4.0 barman. Master’s thesis, Vysoké učení technické v Brně, 2020.
- [59] Jan Kuneš. Lhůtové rozvrhování strojírenské výroby v latecoere s.r.o. Master’s thesis, ČVUT v Praze, 2018.
- [60] Roman Kvasnička. *Interoperabilita znalostí*. PhD thesis, Česká zemědělská univerzita v Praze, 2013.
- [61] Lidovky.cz. Škoda zrychlí a sníží počet dodavatelů. online, 2008. online, https://www.lidovky.cz/byznys/skoda-zrychli-a-snizi-pocet-dodavatelu.A080604_123544_ln_ekonomika_bat, cit. 20.7.2023.
- [62] Vieira Ernani; Jeffrey Guilherme Herrmann; Edward Lin. Rescheduling manufacturing systems: A framework of strategies, policies and methods. *Journal of Scheduling*, 2003.
- [63] Jiří Loffelman. Informační systémy v plánování a řízení výroby. *SystemOnLine*, 2007.
- [64] Alfred Bauer; Richard Bowden; Jim Browne; James Duggan; Gerard J. Lyons. *Shop Floor Control Systems*. Springer Dordrecht, 1 edition, 1994.
- [65] Marek Magáth. Komunikační rozhraní pro testbed i4.0. Master’s thesis, Vysoké učení technické v Brně, 2020.
- [66] Martin Malý. Protokol mqtt: komunikační standard pro iot. online, 2016. online, <https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/>, cit. 6.7.2023.
- [67] Kudzai Manditereza. The key differences between opc ua and mqtt sparkplug. *HiveMQ*, 2022. online, <https://www.hivemq.com/article/iiot-protocols-opcua-vs-mqtt-sparkplug-digital-transformation/>, cit. 16.7.2023.

- [68] Alan S. Manne. On the job-shop scheduling problem. *operations Research*, 1960.
- [69] P. Marcon, J. Jirsa, L. Venkrbec, F. Zezulka, T. Benesl, V. Kaczmarczyk, and J. Arm. An experimental training production line to demonstrate the basics of industry 4.0. *IFAC-PapersOnLine*, 55(4):139–144, 2022. 17th IFAC Conference on Programmable Devices and Embedded Systems PDES 2022 — Sarajevo, Bosnia and Herzegovina, 17-19 May 2022.
- [70] Jiří Marek. *Konstrukce CNC obráběcích strojů IV*. MM publishing, 2018.
- [71] Tomáš Martiník. Systém pro řízení výroby demonstrátoru průmyslu 4.0. Master’s thesis, Vysoké učení technické v Brně, 2023.
- [72] Vladimír Mařík. *Průmysl 4.0 - Výzva pro českou republiku*. Management Press, 2016.
- [73] Marco Mendes, Francisco Restivo, Paulo Leitão, and Armando Colombo. Petri net based engineering and software methodology for service-oriented industrial automation. volume 314, pages 233–240, 02 2010.
- [74] Arnoštka Netrvalová. Úvod do problematiky multiagentních systémů. online, <https://www.kiv.zcu.cz/netrvalo/phd/MAS.pdf>, cit. 15.10.2022.
- [75] Clifford Neuman. Scale in distributed systems. *Readings in Distributed Computing Systems: IEEE Computer Society Press*, 1994.
- [76] Arlen Nipper. Iiot protocols: Comparing opc ua to mqtt. online, dostupné z: <https://www.manufacturingtomorrow.com/article/2020/07/iiot-protocols-comparing-opc-ua-to-mqtt/15534>, 2023.
- [77] Ladislav Nováček. Komunikace mezi digitálními dvojčaty. Master’s thesis, Vysoké učení technické v Brně, 2023.
- [78] The Association of German Engineers. Digital factory fundamentals – part 1 (richtlinien 4499), February 2008. online, <https://www.vdi.de/en/home/vdi-standards/details/vdi-4499-blatt-1-digital-factory-fundamentals>, cit. 15.11.2022.
- [79] Jan Paralič. *Rozvrhování a logistika*. PhD thesis, Technická univerzita v Košicích, 2010.
- [80] Ján Paralič. Riešenie úloh rozvrhovania logickým programovaním ohraničení. Master’s thesis, Technická univerzita v Košiciach, 1997.
- [81] Neil Patel. 7 key strategies that you must learn from apple’s marketing. *Neilppatel.com*, 2016. online, <https://neilpatel.com/wp-content/uploads/2016/03/176-7-Key-Strategies-That-You-Must-Learn-from-Apple%E2%80%99s-Marketing.pdf>, cit. 22.10.2022.
- [82] Michael L. Pinedo. *Scheduling, Theory, Algorithms, and Systems*. Springer New York, NY, 4 edition, February 2014.

- [83] Matěj Poláček. Integrace systému rozšířené reality do testbedu průmysl 4.0. Master's thesis, Vysoké učení technické v Brně, 2020.
- [84] Miroslav Prýmek. Multiagentní systémy v praxi - implementace distribuovaného dohledového systému. Master's thesis, Masarykova Univerzita, 2008.
- [85] Jan Pásek. Automatizace procesů 1, 2, skriptá. 2018.
- [86] Lukáš Rejchlík. Návrh, konstrukce a programové vybavení inteligentního skladu pro testbed průmyslu 4.0. Master's thesis, Vysoké učení technické v Brně, 2019.
- [87] Luis Ribeiro. Cyber-physical production systems' design challenges. In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pages 1189–1194, 2017.
- [88] Roland Rosen, Georg von Wichert, George Lo, and Kurt D. Bettenhausen. About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*, 48(3):567–572, 2015. 15th IFAC Symposium on Information Control Problems in Manufacturing.
- [89] Arndt Lüder; Ambra Calá; Jacek Zawisza; Ronald Rosendahl. Design pattern for agent based production system control — a survey. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pages 717–722, 2017.
- [90] Router. What is opc ua? a practical introduction. *OPC Router*, 2023. online, <https://www.opc-router.com/what-is-opc-ua/>, cit. 14.6.2023.
- [91] Petr Ručka. Spojitá/dávková výroba v prostředí industry 4.0. Master's thesis, Vysoké učení technické v Brně, 2017.
- [92] Daria Ryashentseva. *Agents and SCT based self* control architecture for production systems*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2016.
- [93] Massimo Paolucci; Robert Sacile. *Agent-Based Manufacturing and Control Systems: New Agile Manufacturing Solutions for Achieving Peak Performance*. CRC Press, 2004.
- [94] Michael Schillo. *Multiagent Robustness: Autonomy vs. Organisation*. PhD thesis, Naturwissenschaftlich-Technischen Fakultaten der Universität des Saarlandes, 2004.
- [95] Alexander Belyaev; Christian Block; Birgit Boss; Christian Diedrich; Philippe Juhel; Wilfried Hartmann; Oliver Hillermeier; Nikolaus Ondracek; Stephanie Pfeifer; Frank Scherenschlich; Josef Schmelter. Modelling the semantics of data of an asset administration shell with elements of eclass. Technical report, Platform Industrie 4.0, 2021. online, https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Whitepaper_Plattform-Eclass.html, cit. 10.10.2022.

- [96] Felix Gehlhoff; Luis Alberto (Cruz Salazar); Alexander Fay; Birgit Vogel-Heuser; Matthias Seitz. Automation platform independent multi-agent system for robust networks of production resources in industry 4.0. *IFAC-PapersOnLine*, 51(11):1261–1268, 2021. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.
- [97] Weiming Shen. Distributed manufacturing scheduling using intelligent agents. *IEEE Intelligent Systems* 17(1), 88-94. *IEEE Intelligent Systems*, 17:88 – 94, 02 2002.
- [98] Oleg Shilovitsky. Will tesla motors build their own plm system? *Beyond PLM*, 2013.
- [99] Siemens. Getting to market quickly. Online, 2017. <https://new.siemens.com/global/en/company/stories/industry/getting-to-market-quickly.html>.
- [100] Siemens. Fast, lean, and economical processes in process design. Online (<https://new.siemens.com/global/en/products/automation/industry-software/plant-engineering-software-comos/process-design.html>), 2022.
- [101] Siemens. Siemens plm software. <https://www.plm.automation.siemens.com/global/en/>, 2023.
- [102] Siemens. Simit simulationv11.0 sp1 - gettingstarted. online, <https://support.industry.siemens.com/cs/ww/en/view/109746485>, 2023.
- [103] Petr Sodomka. Pokročilé plánování a řízení výroby. *SystemOnLine*, 2011.
- [104] Siemens PLM Software. Digitální továrna ve Škoda auto. *Automa*, 2013.
- [105] Václav Staněk. Návrh digitálního dvojčete cnc obráběcího stroje. Master’s thesis, Vysoké učení technické v Brně, 2019.
- [106] Eui-Seok Byeon; Robert Storer. Decomposition heuristics for robust job-shop scheduling. *IEEE Transactions on Robotics and Automation*, 1998.
- [107] Jianhua Shi; Jiafu Wan; Hehua Yan; Hui Suo. A survey of cyber-physical systems. *Proc. of the Int. Conf. on Wireless Communications and Signal Processing*, 2011.
- [108] Tomáš Sýkora. Konstrukce a programové vybavení transportních entit pro testbed průmysl 4.0. Master’s thesis, Vysoké učení technické v Brně, 2019.
- [109] B. Giffler; G. Thompson. Algorithms for solving production scheduling problems. *European Journal of Operational Research*, 8:487–503, 1960.
- [110] Jan Tichý. Systémy mes/mom v prostředí industry 4.0. Master’s thesis, Vysoké učení technické v Brně, 2017.

- [111] Barbora Ulrychová. Simplexová metoda řešení jednoduchých optimalizačních úloh. Master's thesis, Západočeská univerzita v Plzni, 2015.
- [112] R.J.M. Vaessens, E.H.L. Aarts, and J.K. Lenstra. *Job shop scheduling by local search*. Computing science notes. Eindhoven University of Technology, 1994.
- [113] VDE. Reference architecture model industrie 4.0 (rami4.0). Technical report, VDI/VDE-Gesellschaft, 2015. online, <https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.html>, cit. 6.11.2022.
- [114] Juliane Fischer; Marcos Marga; Birgit Vogel-Heuser. Model-based development of a multi-agent system for controlling material flow systems. *at - Automatisierungstechnik*, 66:438–448, 05 2018.
- [115] Luis Alberto Cruz Salazar; Daria Ryashentseva; Arndt Lüder; Birgit Vogel-Heuser. Cyber-physical production systems architecture based on multi-agent's design pattern—comparison of selected approaches mapping four agent patterns. *The International Journal of Advanced Manufacturing Technology*, December 2019.
- [116] Antonín Vojáček. Průmyslová komunikace opc ua - 1.díl - popis protokolu. *Automatizace.HW.cz*, 2020. online, <https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1dil-popis-protokolu.html>, cit. 15.7.2023.
- [117] A.G. Volkswagen. Number of volkswagen ag employees worldwide from fy 2008 to fy 2022. online, <https://www.statista.com/statistics/272052/worldwide-number-of-volkswagen-group-employees/>, cit. 7.4.2023, 2023.
- [118] Dominik Váňa. Simulace modelu pomocí nástroje simit. Master's thesis, Vysoké učení technické v Brně, 2018.
- [119] Harvey M. Wagner. An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly*, 1959.
- [120] Lihui Wang; Xi Vincent Wang. *Cloud-Based Cyber-Physical Systems in Manufacturing*. Springer Cham, 1 edition, 2018.
- [121] Andreas Wannagat. *Entwicklung und Evaluation agentenorientierter Automatisierungssysteme zur Erhöhung der Flexibilität und Zuverlässigkeit von Produktionsanlagen*. PhD thesis, Technical University Munich, Germany, 03 2010.
- [122] Archur Laszczewski; Chris Whiteford. Cybersecurity and it/ot convergence: A pathway to digital transformation. online, 2021. online, <https://www.plantengineering.com/articles/cybersecurity-and-it-ot-convergence-a-pathway-to-digital-transformation/>, cit. 5.6.2022.
- [123] M.J. Wooldridge and N.R. Jennings. Software engineering with agents: pitfalls and pratfalls. *IEEE Internet Computing*, 3(3):20–27, 1999.

- [124] Jian Fang; Yugent Xi. A rolling horizon job shop rescheduling strategy in the dynamic environment. *International Journal on Advanced Manufacturing Technologies*, 1997.
- [125] Miroslav Zajíc. Návrh strategické implementace konceptu průmysl 4.0. Master's thesis, ČVUT, 2019.
- [126] F. Zezulka, J. Jirsa, L. Venkrbec, P. Marcon, T. Benesl, V. Kaczmarczyk, J. Arm, and Z. Bradac. The ideas of industry 4.0: Seven years after. *IFAC-PapersOnLine*, 55(4):145–150, 2022. 17th IFAC Conference on Programmable Devices and Embedded Systems PDES 2022 — Sarajevo, Bosnia and Herzegovina, 17-19 May 2022.
- [127] Vlastimil Braun;František Zezulka. Úvod do problematiky a základní modely industry 4.0. <https://www.systemonline.cz/rizeni-vyroby/uvod-do-problematiky-a-zakladni-modely-industry-4.0.htm>, January 2017.
- [128] Platform Industrie 4.0; ZVEI. Submodel templates of the asset administration shell, 2020. online, <https://www.zvei.org/>, cit. 15.11.2022.
- [129] Petr Černocký. Demonstrační systém výrobní linky řízené pomocí mes s využitím aas. Master's thesis, Vysoké učení technické v Brně, 2022.
- [130] Český statistický úřad. Informační společnost v číslech 2022. Technical report, Český statistický úřad, 2022. online, <https://www.czso.cz/csu/czso/informacni-spolecnost-v-cislech-2022>, cit. 15.4.2023.
- [131] Josef Čipera. Toc – nový nástroj řízení nejen výroby. *SystemOnLine*, 2006. online, <https://www.systemonline.cz/aps-scm/toc-novy-nastroj-rizeni-nejen-vyroby.htm>, cit. 1.3.2023.
- [132] Lukáš Ďurica. *Multiagentový logistický systém s implementáciou vo virtuálnej realitě*. PhD thesis, Žilinská univerzita v Žilíně, 2016.
- [133] GK Šanghaj. Výroba automobilů Škoda octavia v Číně. online, <https://www.mpo.cz/cz/zahranicni-obchod/mezinarodni-obchod-dle-teritorii/asie-australie-a-tichomori/vyroba-automobilu-skoda-octavia-v-cine-31734>, cit. 13.7.2023, 2007.
- [134] Škoda Auto. Škoda auto dodala v roce 2022 celosvětově 731 300 automobilů. online, <https://www.skoda-auto.cz/novinky/novinky-detail/2023-01-11-skoda-auto-dodala-v-roce-2022-celosvetove-731-300-automobilu>, cit. 7.2.2023, 2023.
- [135] Dagmar Šulová. *Metody plánování a řízení výroby v podnikových informačních systémech a jejich uplatnění při řízení výrobního procesu*. PhD thesis, Universita Tomáše Bati ve Zlíně, 2009.

- [136] Juraj Švančara. *Simulační analýza mnohovýrobního výrobního systému*. PhD thesis, Slovenská technická univerzita v Bratislavě, 2012.
- [137] Martin Žiačik. *Multiagentní systémy pro distribuovanou výrobu*. Master's thesis, Vysoké učení technické v Brně, 2023.