

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ MECHANICZNY
INSTYTUT TECHNOLOGII MASZYN I AUTOMATYZACJI

Na prawach rękopisu

Jacek Reiner

MODELOWANIE OBIEKTOWE I UML
DLA
AUTOMATYZACJI WYTWARZANIA

PRACA DOKTORSKA

Słowa kluczowe:

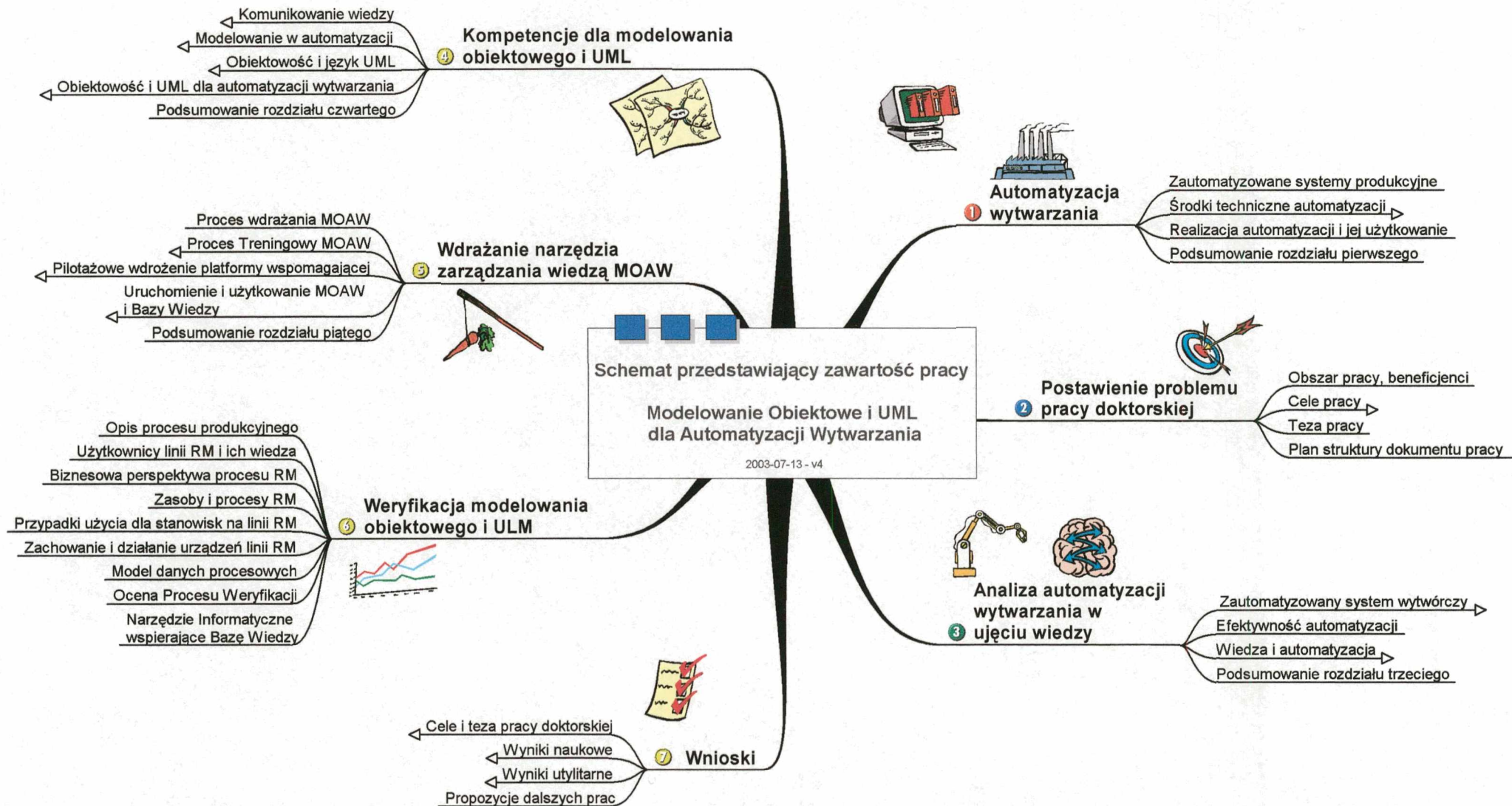
automatyzacja, praca grupowa, modelowanie obiektowe,
UML, zarządzanie wiedzą

Promotor:

prof. zw. dr hab. inż. dr h.c. Jan KOCH

1	<i>Automatyzacja wytwarzania</i>	3
1.1	Zautomatyzowane systemy produkcyjne	3
1.2	Środki techniczne automatyzacji	5
1.2.1	Sterowniki PLC i PC	5
1.2.2	Sieci komunikacji przemysłowej	9
1.3	Realizacja automatyzacji i jej użytkowanie	12
1.4	Podsumowanie rozdziału pierwszego	17
2	<i>Postawienie problemu pracy doktorskiej</i>	19
2.1	Obszar pracy, beneficjenci	19
2.2	Cele pracy	19
2.3	Teza pracy	22
2.4	Plan struktury dokumentu pracy doktorskiej	24
3	<i>Analiza automatyzacji wytwarzania w ujęciu wiedzy</i>	25
3.1	Zautomatyzowany system wytwórczy	26
3.1.1	Systemy socjo-techniczne	26
3.1.2	Cykl życia zautomatyzowanego systemu wytwórczego	28
3.1.3	Automatyzacja wytwarzania	29
3.1.4	Eksploatacja i utrzymanie	35
3.2	Efektywność automatyzacji	38
3.3	Wiedza i automatyzacja	42
3.3.1	Produkty automatyzacji	44
3.3.2	Interakcja między aktorami	46
3.3.3	Komunikacja w systemie socjo-technicznym	48
3.3.4	Komunikacja podczas cyklu życia automatyzacji	52
3.4	Podsumowanie rozdziału trzeciego	54
4	<i>Kompetencje dla modelowania obiektowego i UML</i>	55
4.1	Komunikowanie wiedzy	55
4.1.1	Diagramatyka	55
4.1.2	Model i modelowanie	59
4.2	Modelowanie w automatyzacji	65
4.2.1	Programowanie graficzne w automatyzacji	65
4.2.2	Bloki Funkcyjne w automatyzacji wytwarzania	68
4.2.3	Paradygmat automatyzacji	71
4.3	Obiektowość i język UML	73
4.3.1	Paradygmat obiektowy	73
4.3.2	Standard UML	76
4.4	Obiektowość i UML dla automatyzacji wytwarzania	83
4.4.1	Modelowanie biznesowe procesów automatyzacji	84
4.4.2	Modelowanie logicznej struktury procesów	85
4.4.3	Modelowanie zakresu i funkcjonalności systemów	87
4.4.4	Modelowanie dynamiki systemu i jego obiektów	88
4.5	Podsumowanie rozdziału czwartego	88
5	<i>Wdrażanie narzędzia zarządzania wiedzą MOAW</i>	90
5.1	Proces wdrażania nowego narzędzia zarządzania wiedzą	90
5.2	Proces Treningowy MOAW	93
5.2.1	Motywacja uczestników	93
5.2.2	Praca grupowa – komunikacja i kompetencje	95

5.2.3	Przebieg treningu	97
5.3	Pilotażowe wdrożenie platformy wspomagającej	99
5.3.1	Platforma wspomagająca.....	99
5.3.2	Spodziewane rezultaty z wdrożenia Bazy Wiedzy.....	100
5.3.3	Narzędzia wspierające MOAW.....	103
5.4	Uruchomienie i użytkowanie MOAW i Bazy Wiedzy.....	104
5.4.1	Akceptacja czy jakość?	104
5.4.2	Skalowanie systemu	105
5.4.3	Wymagania i zagrożenia dla wdrożenia Bazy Wiedzy	106
5.5	Podsumowanie rozdziału piątego	107
6	<i>Weryfikacja modelowania obiektowego i ULM.....</i>	109
6.1	Opis procesu produkcyjnego	109
6.2	Użytkownicy linii RM i ich wiedza	111
6.3	Biznesowa perspektywa procesu RM.....	112
6.4	Zasoby i procesy RM.....	114
6.5	Przypadki użycia dla stanowisk na linii RM	117
6.6	Zachowanie i działanie urządzeń linii RM	119
6.7	Model danych procesowych	122
6.8	Narzędzie informatyczne wspierające Bazę Wiedzy.....	123
6.9	Ocena Procesu Weryfikacji.....	124
7	<i>Wnioski.....</i>	127
7.1	Cele i teza pracy doktorskiej.....	127
7.2	Wyniki naukowe	130
7.2.1	Automatyzacja wytwarzania	130
7.2.2	Narzędzie zarządzania wiedzą.....	130
7.2.3	Metodyka wdrażania narzędzia wiedzy MOAW.....	130
7.3	Wyniki użytkowe.....	130
7.3.1	Narzędzie zarządzania wiedzą dla poprawy automatyzacji wytwarzania	130
7.3.2	Sposób wdrożenia narzędzia zarządzania wiedzą	131
7.3.3	Weryfikacja - Projekt „RMdata+”	131
7.4	Propozycje dalszych prac.....	131
Literatura		133



Wstęp

Oczekiwania klienta i konkurencyjność przedsiębiorstw zmuszają do ciągłego i bardzo dynamicznego rozwoju produktów. Presja redukcji kosztów i utrzymywania wysokiej jakości, w odniesieniu do każdego z produktów indywidualnie, formułują trudne wymagania dla technologii ich wytwarzania.

Spełnienia powyższych wymagań poszukuje się w automatyzacji wytwarzania. Automatyzacja dysponuje coraz bardziej zaawansowanymi środkami technicznymi, stąd zrównanie udziału oprogramowania i mechaniki we współczesnych systemach produkcyjnych, na poziomie 40%. Technika określana jest jako coraz bardziej przyjazna dla człowieka (ang. *user friendly*).

Równocześnie oprogramowanie wymaga ciągłych poprawek i udoskonaleń dla zapewnienia niezawodności i bezpieczeństwa, a projekty programistyczne blisko w połowie kończą się niepowodzeniem. Jako nadrzędną przyczynę powyższych problemów identyfikuje się kompleksowość (ang. *complexity*). Kompleksowość w przeciwieństwie do złożoności nie daje się zredukować poprzez dekompozycję problemu, lecz wymaga całościowego rozwiązywania wybranej perspektywy problemu. Kompleksowość ujawnia się na *styku człowiek – technika*. Relacja ta dotyczy użytkowników obsługi, jak również użytkowników opracowywania i utrzymania zautomatyzowanych systemów wytwórczych, tj. systemów socjotechnicznych. Problemy niespójności lub braków wiedzy o automatyzacji, ujawniają się szczególnie, gdy wymagane jest zaangażowania zespołu interdyscyplinarnego. Skutkują one spadkiem efektywności automatyzacji, zagrożeniem bezpieczeństwa ludzi, instalacji czy środowiska, lub uniemożliwiają jej realizację.

Wielu autorów wskazuje, że współczesny człowiek jest przeciążony techniką, dlatego też albo jej nie rozumie, albo też nie radzi sobie organizacyjne z jej nadmiarem.

Motywacją do niniejszej pracy było rozwiązanie wybranych problemów automatyzacji poprzez zwrócenie uwagi na jej użytkownika – człowieka. Takie postawienie problemu wymagało od autora wyjścia poza paradygmat inżynierskiego i technicznego poszukiwania rozwiązań i sięgnięcia do obszarów humanistycznych. Największym problemem okazała się konieczność radzenia sobie z *miękkimi* wskaźnikami charakteryzującymi problemy humanistyczne, jak np. wiedza o automatyzacji.

Autor zaproponował zestaw umiejętności w zakresie Modelowania Obiektowego i języka UML dla Automatyzacji Wytwarzania, który został nazwany MOAW. Kompetencje te są adresowane zarówno dla automatyków jak również i nie-automatyków, stanowiąc w ten sposób platformę wymiany wiedzy o automatyzacji wytwarzania. W wyniku komunikacji powstaje jednocześnie dokumentacja archiwizująca wiedzę o wymaganiach i rozwiązaniu. Dla efektywnego użytkowania *narzędzia zarządzania wiedzą* MOAW, przedyskutowano problematykę jego wdrożenia i utrzymania.

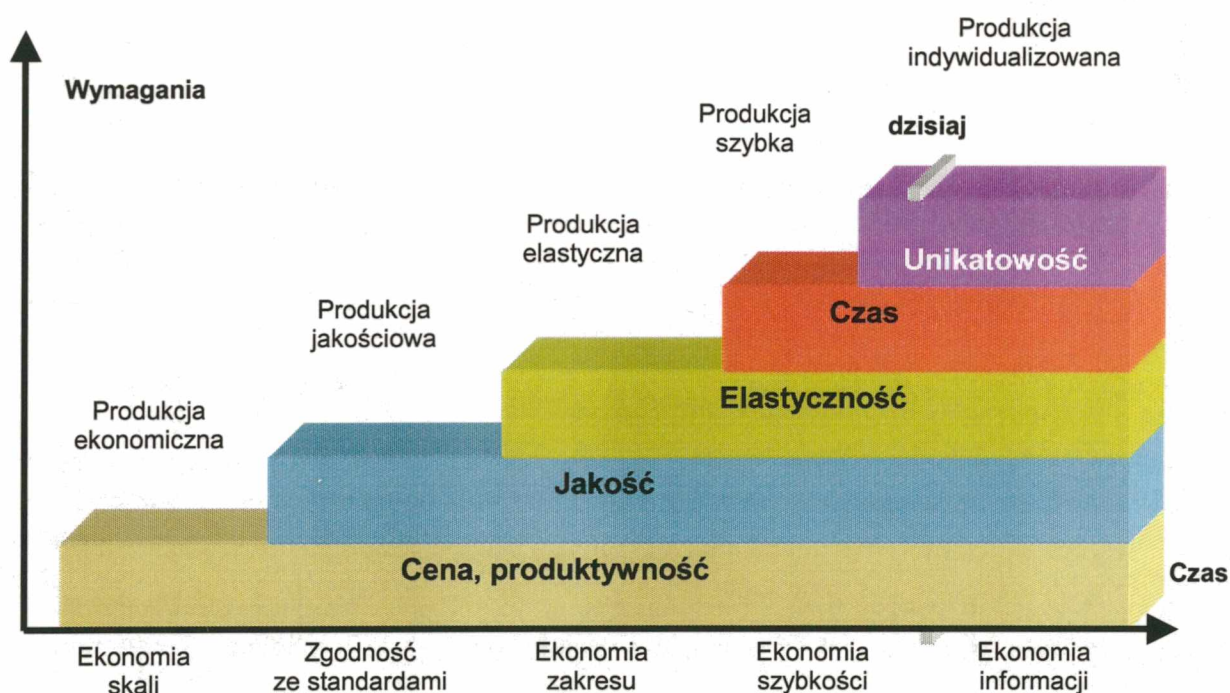
Przedstawione narzędzie MOAW i jego metodyka wdrażania zostały zweryfikowane w grupie pracowników odpowiedzialnych za utrzymanie i rozwój linii montażu silników w zakładzie produkcyjnym.

1 Automatyzacja wytwarzania

1.1 Zautomatyzowane systemy produkcyjne

Wizje nowoczesnej produkcji z 1991 roku (tzw. CIM¹ pierwszej generacji), mówiły o produkcji na zamówienie, seriach dopuszczających wielkości zaledwie jednej sztuki (produkt seryjny, nie na zamówienie) oraz intensywnym przetwarzaniu danych. Wynikała z tego jednak potrzeba częstego przeprogramowywania i rekonfigurowania linii oraz stanowisk produkcyjnych. Aktualna sytuacja dowodzi, że produkcja na zamówienie stała się faktem. Na samochód, którego wyposażenie dobierane jest indywidualnie, oczekuje się ok. 3 miesiące, a na komputer do 1 tygodnia. Japońska gospodarka kreśli wizję samochodu dostępnego po 3-5 dniach od zamówienia, a jego konfiguracja jest ustalana indywidualnie przez Internet. Na najważniejszych konferencjach wyznaczających kierunki rozwoju produkcji dyskutuje się o nieuniknionej konieczności permanentnych zmian.

Jednocześnie klient, tzn. jego oczekiwania identyfikowane są nadrzędnie, i to on wyznacza kierunek rozwoju produktów a tym samym i systemów produkcyjnych i automatyzacji [Rockwell 97].



Rys. 1.1 Wymagania dla systemów produkcyjnych [iwb 03]

Najbardziej rozpowszechniony model wymagań dla systemów produkcyjnych wskazuje na wymagania kosztowe i produktywności, jakości, elastyczności, szybkości produkcji a ostatnio unikatowości wyrobów [iwb 03].

¹ CIM Computer Integrated Manufacturing (Komputerowo Zintegrowane Wytwarzanie)

Dla bardziej szczegółowej identyfikacji wymagań względem systemów wytwórczych, przedstawia się je w kontekście użytkowników. Przykładowo dla inwestora:

- koszty,
- wydajność systemu,
- bezpieczeństwo,
- ograniczenia techniczne i polityczne,

dla procesu technologicznego:

- wydajność,
- niezawodność,

dla produkcji i technologii:

- dokumentacja,
- przyjazność,
- elastyczność,

dla utrzymania ruchu:

- diagnostyka,
- serwisowalność,

dla rozwoju:

- dokumentacja,
- otwartość,
- standaryzacja,
- rekonfigurowalność.

Wymagania dla systemów produkcyjnych przejmowane są nawet ze strategii wojskowych, jak przykładowo wymaganie „żwawości” lub „zwinności” (ang. *agility*), tj. zdolności systemu technicznego do szybkiej reakcji na wymagane zmiany.

Prócz rozszerzającego się zakresu wymagań szczególnie mocno akcentowana jest dynamika zmian.

„to nie duzi zjadają małych, ale szybcy opieszalnych” [Jennings, Haughton 02]

Możliwość spełnienia wszystkich, często przeciwstawnych wymagań, poszukiwana jest w zastosowaniu zaawansowanej automatyzacji procesów produkcyjnych.

Wskazuje się przy tym, że klasyczne scentralizowane systemy sterowania muszą zostać zastąpione systemami rozproszonymi, agentowymi [Shen, Norie], a nawet biologicznymi [Ueda 96] jak np. systemy holoniczne [Brussel, et al. 98]. Systemy holoniczne są wzorowane na organizmach biologicznych wykazujących takie cechy jak inteligencja, autonomia i kooperowanie.

Jednocześnie wskazuje się na rosnące wymagania wobec człowieka, tj. projektanta powyższych systemów, ale również ich użytkownika.

Człowiek podobnie jak systemy produkcyjne poddany jest rozbieżnym wymaganiom dyktowanym przez technologie, informacje, cele przedsiębiorstwa, kompleksowość, kreatywność, etc. [Tscheligi 01]. Stąd aktualne wskazania na konieczność rozwoju organizacji, oraz jej najważniejszego elementu tj. człowieka, jego wiedzy, motywacji i zaangażowania, przed techniką [iwb03].

1.2 Środki techniczne automatyzacji

1.2.1 Sterowniki PLC i PC

Początki sterowania elektrycznego sięgają lat 1930 i układów przekaźnikowych. Na bazie odpowiednio połączonych styków zwiernych i rozwiernych oraz cewek przekaźników, implementowane były funkcje logiczne realizujące algorytm sterowania procesem. Tego typu rozwiązania cechował duży koszt i brak elastyczności umożliwiającej modyfikację algorytmu sterowania. Wraz z opracowaniem pierwszych układów tranzystorowych, ok. roku 1960, zaczęto zastępować „logikę elektryczno-mechaniczną” przez rozwiązania elektroniczne. Prawdziwym przełomem jednak był rok 1970, i wprowadzenie do sterowania sterownika swobodnie programowalnego PLC (ang. *Programmable Logic Controller*). Urządzenie to posiadało 128 wejść/wyjść dwustanowych, pamięć 1Kbyte oraz zestaw instrukcji logicznych do programowania. O sukcesie sterownika PLC zadecydowały dwa elementy: łatwa programowalność i łatwość zagwarantowania wymagań czasowych procesu.

Sterownik PLC został wprowadzony do automatyzacji jako urządzenie, które jednak początkowo **nie zmieniło myślenia o sposobie rozwiązywania problemów sterowania**, gdyż podstawowym sposobem programowania PLC pozostało rysowanie schematu elektrycznego. Prosta składnia i zrozumiała semantyka języka drabinkowego LD (ang. *ladder diagram*) spowodowały, że jest on stosowany jeszcze do dzisiaj. Mimo, że jest pozbawiony „informatycznego myślenia” o programowaniu, to w roku 2000 używało go aż 93% automatyków, respondentów ankiety „Control Engineering” [Ctrl. Eng. 00]. Oznacza to, że **użytkownicy sterowników PLC nie są informatykami, a raczej elektrykami i technologami znającymi lepiej proces technologiczny niż sztukę programowania**.

Tak duże zastosowanie języka LD nie oznacza jednak wyłączności, gdyż podczas programowania PLC można wykorzystywać wiele języków równocześnie. Zależnie od potrzeb i specyfiki rozwiązywanych zadań stosowane są języki tekstowe na poziomie assemblera np. IL (ang. *instruction list*) lub języki wysokiego poziomu np. ST (ang. *structured text*). Jednocześnie, **coraz większego znaczenia nabierają języki graficzne** np. SFC (ang. *sequential function chart*) i FBD (ang. *function block diagram*), pozwalające zapisywać działanie systemu za pomocą symboli graficznych. Według badań Control Engineering z roku 2000, 20% respondentów używa bloków funkcyjnych, co stanowi wzrost ich liczby o 3%, w porównaniu z rokiem poprzednim [Ctrl. Eng. 00].

Różnorodność wersji, odmian i modyfikacji języków programowania sterowników PLC, doprowadziła do powstania inicjatywy ujednolicenia i standaryzacji. Międzynarodowa norma IEC 61131 [IEC61131], postawiła sobie za cel, zapewnienie standaryzowanego **interfejsu między użytkownikiem a systemem sterowania, dla użytkowników z różnym przygotowaniem programistycznym**.

Informacje ogólne	„IEC61131-1 General information”
Sprzęt	„IEC61131-2 Hardware”
Języki programowania PLC	„IEC61131-3 PLC programming languages”
Zalecenia dla użytkownika	„IEC61131-4 User guidelines”
Komunikacja	„IEC61131-5 Messaging service specification”

Standard IEC61131, bazując na praktyce inżynierskiej, nie wprowadza żadnych radykalnych zmian do programowania sterowników. Systematyzuje on jedynie pięć z najbardziej rozpowszechnionych języków tekstowych i graficznych: język drabinkowy (LD), strukturyzowany tekst (ST), lista instrukcji (IL), bloki funkcyjne (FBI) i diagram sekwencji funkcji (SFC) – patrz tabela 1.1.

Tabela 1.1 Języki programowania normy IEC61131

Ladder Diagram (LD) dla logiki (przepływ prądu)	Język oparty na schemacie elektrycznym układu logiki stycznikowej. Poszczególne sieci opisują niezależne, równoległe warunki logiczne. Preferowany przez elektryków, szczególnie rozpowszechniony w Stanach Zjednoczonych
Function Block Diagram (FBD) dla sterowania regulującego (przepływ danych)	Bloki funkcyjne łączone są podobnie jak na schematach elektrycznych, szczególnie stosowany dla aplikacji z przepływem i przetwarzaniem informacji lub danych. Umożliwia dekompozycję problemu sterowania na moduły funkcjonalne.
Structured Text (ST) dla przetwarzania informacji	Język wysokiego poziomu, wprowadzający struktury blokowe. Syntaktyka podobna do PASCAL'a, umożliwia budowę kompleksowych wyrażeń oraz zagnieżdżanie funkcji. Zawiera funkcje iteracji, funkcje warunkowe oraz matematyczne.
Instruction List (IL) dla programowania na poziomie asemblera.	Język IL bazuje na modelu przetwarzania przez pojedynczy akumulator, jedna linia zawiera jedną operację. Szczególnie rozpowszechniony w Niemczech, bazuje na języku AWL (<i>niem. Anweisungsliste</i>)
Sequential Function Chart (SFC) język sekwencji (tzw. pierwowzorem był GRAFCET)	Graficzny język dla opisu sekwencyjnego zachowania systemów sterowania za pomocą bloków akcji i warunków przejść między nimi (tranzycji). Umożliwia dekompozycję problemu sterowania na fazy technologiczne, wspiera równoległe i warunkowe przetwarzanie

Drugi element sukcesu sterowników PLC dotyczy spełnienia wymagań czasu rzeczywistego (ang. *RT real-time*) wymaganego przez sterowany proces. Dawny, równoległy układ przekaźnikowy, został odwzorowany na maszynie von Neumana, czyli jednym procesorze potokowym wyposażonym w pamięć i peryferia. Działanie PLC jest następujące: sterownik „zatrzaskuje” wektor stanów wejściowych, następnie przetwarza wszystkie warunki logiczne i w ostatnim cyklu uaktualnia stany wszystkich swoich wyjść. Dla takiego algorytmu, czas cyklu sterownika zależy od szybkości procesora i długości programu, czyli liczby instrukcji. Ponieważ oba parametry, w określonej aplikacji, są niezmiennie, to sterownik pracuje cyklicznie i deterministycznie. Stąd mówimy o **sekwencyjnym przetwarzaniu PLC, które upraszcza rzeczywiste współbieżne i niezależne występowanie zdarzeń w procesie technologicznym**.

Współczesne sterowniki PLC są rozszerzone o bogaty zestaw funkcji, nie tylko logicznych dla sterowania procesami dyskretnymi, ale również przetwarzania wartości analogowych w formacie zmiennoprzecinkowym, a nawet sterowania i synchronizacji napędów. Funkcje *fuzzy logic* oraz mechanizmy przerwań, szybkich liczników i komunikacji obiektowej *fieldbus* a także LAN (patrz rozdział 1.2.2) rozszerzyły wyraźnie obszar zastosowania PLC. Rozwój mikroelektroniki pozwolił na uzyskanie wielokrotnie, nawet 10^6 razy większych mocy obliczeniowych oraz zwiększenie zasobów pamięci tych sterowników, w porównaniu z pierwszymi rozwiązaniami z lat 70 [Smalec et al. 02]

Ciągle jednak, w przeciwieństwie do komputera PC, nie dążono i **nie dokonała się *defacto*, żadna standaryzacja sprzętu sterowników PLC**. Rozwiązania te są **podporządkowane opracowaniom firmowym**, których szczegóły nie są ujawniane. Wprowadzenie standaryzowanych interfejsów komunikacyjnych, jak np.: PROFIBUS [EN50170], Interbus [EN50245], umożliwiło wprowadzenie zestawianie heterogenicznych systemów automatyzacji,

jednak nadal **każdy sterownik PLC wymaga firmowego środowiska programowania**. W Europie, udział firmy SIEMENS wraz z jej głównym produktem w zakresie sterowników PLC SIMATIC S7, programowalnym w środowisku STEP 7, szacowany jest na 50% [Seel 00].

Wprowadzanie komputerów osobistych do automatyzacji wytwarzania uzależnione jest od wymagań, jakie mają spełniać. Komputerowe wsparcie prac inżynierskich, z uwagi na analogie do aplikacji biurowych, jest bardzo mocno reprezentowane przez systemy CAx (patrz rozdział 1.3).

Analiza porównawcza cech sterowników PLC i komputerów PC, pod kątem sterowania procesem technologicznym, zwraca uwagę na następujące odmienności [Rockwell 97b]:

Zalety rozwiązań PLC:

- niższe koszty utrzymania (organizacja infrastruktury),
- koszt sprzętu i oprogramowania optymalizowany pod kątem sterowania,
- krótki czas uruchamiania („podnoszenia”) systemów,
- odporność na warunki przemysłowe,
- wyższa niezawodność i krótsze czasy przestojów, MTBF (*ang. Mean Time Between Failures*), MTTR (*ang. Mean Time To Repair*),
- możliwość naprawy i/lub wymiany podczas działania urządzenia lub systemu,
- zintegrowana magistrala wejściowo/wyjściowa,
- zintegrowany układ nadzorujący tzw. „wachdog”,
- przemysłowe rozwiązania interfejsu z użytkownikiem (*ang. touchscreen*),
- mniejszy nakład na utrzymanie oprogramowania.

Zalety rozwiązań PC:

- niższy koszt (produkty rynku masowego),
- zintegrowana grafika,
- duża pamięć RAM i dyskowa,
- duży wybór oprogramowania komercyjnego, *shareware* i *freeware*,
- otwarta komunikacja,
- integracja z bazami danych SQL,
- duży wybór narzędzi programistycznych,
- bogate systemy operacyjne,
- **dynamiczny postęp badawczy i wdrożeniowy.**

Jak wynika z powyższego zestawienia z roku 1997, przewaga PLC nad PC wynikała głównie z kryteriów kosztowych, lepszego przygotowaniu PLC do pracy w warunkach przemysłowych i wyższej niezawodności.

Ponieważ podstawą obu sterowników jest mikroprocesor, moc obliczeniowa obu rozwiązań podlega jednakowemu rozwojowi zgodnie z prawem Morea². Jednak komputer PC dzięki

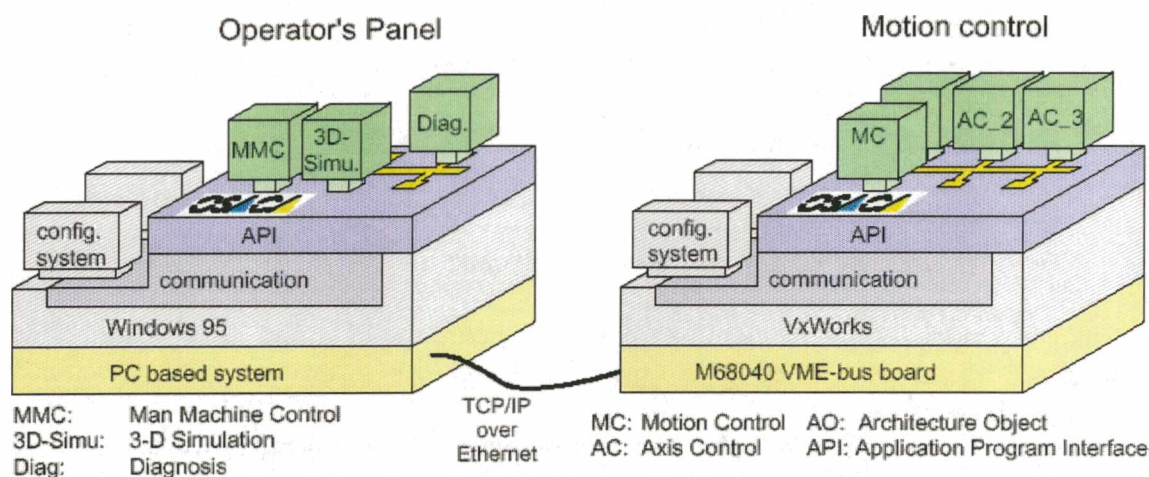
² Jedno z podstawowych praw rządzących rynkiem informatycznym tzw. prawo Morea. Zostało ono sformułowane ponad ćwierć wieku temu przez jednego z założycieli firmy Intel i wyznaczyło obszary rozwoju przemysłu komputerowego. Zgodnie ze sformułowaną wtedy tezą, moc obliczeniowa komputerów podwaja się w cyklach dwuletnich. Teza ta nieustannie sprawdza się w praktyce.[Indor]

otwartości architektury i efektowi skali, tzn. **liczbie producentów i użytkowników** oraz konkurencyjności między nimi, rozwija się niewspółmiernie dynamicznie.

Problem zastępowania PLC przez PC nie wynika dzisiaj z niedostatecznej szybkości działania, czy dostosowania do warunków przemysłowych (rozwiązania typu wbudowanego ang. *embedded*), lecz **odmiennej filozofii ich użytkowania**. „Filozofia PC” kształtowana jest głównie przez firmę Microsoft, która oferuje systemy operacyjne Windows dla użytku domowego. Systemy te zapewniają wprawdzie łatwy i przyjemny interfejs z użytkownikiem, jednak wymagania bezpieczeństwa i niezawodności traktują drugoplanowo. W takiej sytuacji zastosowanie rozwiązań biurowych i domowych do sterowania procesem technologicznym nie jest możliwe.

Pierwsze, „odpowiedzialne” zastosowania komputera PC z biurowym systemem operacyjnym dla sterowania, wykorzystywały dedykowany procesor (koprocesor) do obsługi zadań krytycznych czasowo. Karty z koprocesorem i interfejsem równoległym bądź sieciowym do wejść i wyjść procesu, były umieszczane w magistrali rozszerzeń ISA/PCI komputera. Stąd przyjęta nazwa dla powyższych rozwiązań SlotControl lub SlotPLC. W takiej konfiguracji, komputer PC pełnił rolę platformy zapewniającej interfejs z człowiekiem, tj. wizualizację, komunikację w sieci przedsiębiorstwa oraz środowisko programowania. Najistotniejszą wadą rozwiązania SlotControl był jednak brak skalowalności oraz zamknięte jądro programu sterującego, co ograniczało istotnie rozbudowę takich sterowników.

Wraz z rozwojem systemów operacyjnych i odpowiednich rozszerzeń deterministyczności systemu Windows [RTX], [InTime], [HyperKernel], [LP-Win], zrealizowano kompletny sterownik PLC na komputerze PC. Powyższe rozwiązania nazywane są SoftPLC lub SoftControl. Sposób ich programowania jest zgodny z normą IEC1131-3. Takie podejście do programowania sterowników **SoftPLC** wskazuje, że wprowadzenie PC do sterowania **nie zmieniło zasadniczo sposobu myślenia o realizacji sterowania wytwarzaniem, a rozwiązania te adresowane są bardziej dla elektryków i technologów, jak dla programistów.**



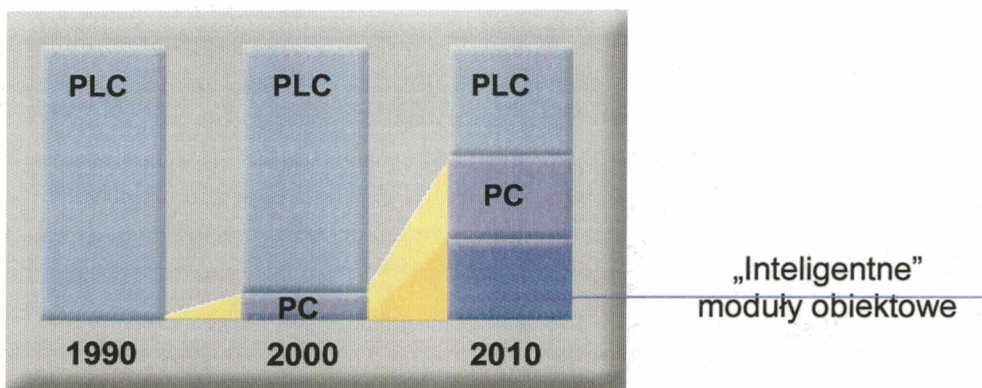
Rys. 1.2 Platforma otwartego układu sterowania OSACA [Sperling, Lutz 96]

Przykład **zupełnie nowego myślenia o sterowaniu wytwarzaniem opracowano w ramach projektu OSACA** (*Open System Architecture for Controls Association*) [OSACA 97] i [Pritschow, Daniel 96]. Opracowane rozwiązanie proponuje możliwość budowania sterowania z modułów (analogia do klocków Lego) (rys.1.2). OSACA specyfikuje architekturę otwartego systemu sterowania dla systemów automatyki oraz zestawienia wymagania softwarowych platform komunikacyjnych dla implementacji systemów zgodnych

z tą architekturą. Dla elastyczności rozwiązania, ułatwienia jego zrozumienia i integracji całego systemu, składającego się z logicznie różnych aplikacji (zwanymi też modułami funkcjonalnymi), zastosowano technologię modułową. Implementacja software'owa modułów funkcyjnych nazwana została obiektami architektury AO (ang. *Architecture Object*). Z punktu widzenia architektury system OSACA dzieli się na dwie części: oprogramowanie aplikacyjne oraz platformę systemową. Oprogramowanie aplikacyjne łączy funkcjonalność charakterystyczną dla systemów sterowania, natomiast platforma systemowa obejmuje oprogramowanie systemowe, które nie jest charakterystyczne dla typu sterowania, lecz jest niezbędne do wsparcia działania oprogramowania aplikacyjnego i bazuje na sprzęcie.

Podobną inicjatywę, do europejskiego projektu OSACA, podjęto w ramach Amerykańskiego projektu OMAC (ang. *Open Modular Architecture Controllers*) i OSEC (ang. *Open System Environment for Controllers*). Mimo szczegółowych specyfikacji rozproszonych architektur sterowania i demonstracji ich działania, **propozycje te nie spowodowały istotnej zmiany w sposobie opracowywania sterowań.**

Mimo powyższych doświadczeń, **wskazuje się na wyraźny wzrost liczby stosowanych komputerów PC i „inteligentnych” modułów obiektowych w najbliższych latach** (rys. 1.3).



Rys. 1.3 Przewidywany udział sterowników PLC i PC w automatyzacji wytwarzania [PNO 01]

1.2.2 Sieci komunikacji przemysłowej

Sieci komunikacyjne znane z informatyki biurowej, takie jak Ethernet, w latach 1990 nie mogły spełnić wymagań komunikacji przemysłowej, głównie ze względu na brak zapewnienia deterministyczności, niskie bezpieczeństwo danych i brak usług wspierających obsługę procesu. Pierwsze rozwiązania komunikacyjne dedykowane dla przemysłu to protokół MAP (ang. *Manufacturing Application Protocol*) i usługi MMS (ang. *Manufacturing Message Specification*). Rozwiązania te, w pełni zgodne z modelem komunikacyjnym ISO/OSI, okazały się jednak zbyt drogie dla zastosowań przemysłowych. O ich wysokiej cenie decydował rozbudowany protokół komunikacyjny, ale również konieczność bardzo obszernej konfiguracji, co podrażało koszty inżynierii [Smalec et al. 97], [Koch et al. 99], [Koch et al. 00].

W tym czasie przemysł oczekiwał na efektywne rozwiązanie komunikacji między sterowaniem a procesem, tzn. modułami wejść/wyjść.

Pierwsze rozwiązania określone jako sieci obiektowe - *fieldbus*, takie jak PROFIBUS-FMS, redukowały wprowadzając model komunikacyjny ISO/OSI do trzywarstwowego, ale mocno odwzorowywały usługi MMS, chcąc w ten sposób stanowić rozwiązanie uniwersalne. **Teoretycznie wprowadzały nową jakość do komunikacji, ale praktycznie rynek nie był**

gotowy do ich przyjęcia. Integratorzy wykorzystywali jedynie podstawowe funkcje odczytu i zapisu (*read/write*), pomijając pozostałe 40 usług, a z siedmiu relacji komunikacyjnych, wykorzystywane były jedynie 2-3 podstawowe. Ponadto automatycy nie zaakceptowali odmiennej „filozofii komunikacji” jak obowiązująca w branży. Przykładowo oferowane były „obiekty komunikacyjne” dla procesów ciągłych, gdy tymczasem branża ta opisywała proces za pomocą bloków funkcyjnych.

Dopiero rozwiązania komunikacyjne, bazujące na modelu odwzorowywania pamięci procesu i sterowania (ang. *memory-map*), takie jak PROFIBUS-DP, Interbus, AS-I, etc., okazały się sukcesem. Przy zastosowaniu powyższych rozwiązań, aplikacja sterująca PLC „nie widzi” różnicy w pracy z lokalnymi czy też rozmieszczonymi na obiekcie modułami wejść i wyjść (rys. 1.4). Konfigurowanie i uruchomienie sieci PROFIBUS-DP czy Interbus jest na tyle proste, że nie wymaga żadnej specjalistycznej wiedzy ani sprzętu [Mrzygłód, et al., 97].

Tabela 1.2 Rozwój systemów sieciowych

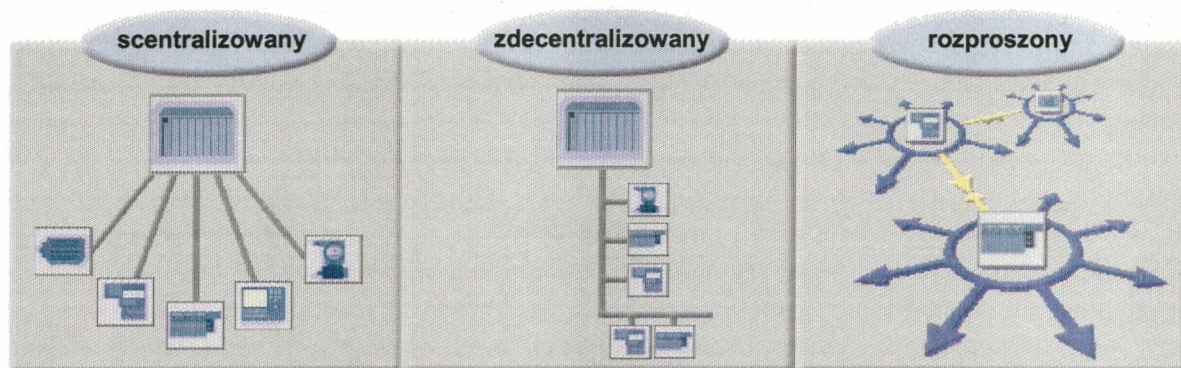
Rok	Zadanie	Realizacja
1980	LAN Transmission	Ethernet, Token-bus, EIA RS 485, Manchester
1985	Interconnectivity Interworking	MAP3.0/MMS, PROFIBUS, InterBus-S, FIP, IEC Fieldbus
1990	Interoperability	FB's, IEC-61131, UL, DDL
1995	Interchangeability Portability Reusability	Profile, Obiekty
2000	Internet, Safety, Distributed, Component-oriented	Web-Servers, PROFISafe, IEC61499, PROFINet

Oszczędności z zastosowania sieci *fieldbus*, wynikają z zastąpienia komunikacji równoległej przez szeregową (okablowanie) i zastąpienia komunikacji analogowej przez cyfrową (niezawodność, efektywność, serwisowanie) [Reiner 99]. W miarę rozbudowy funkcjonalności urządzeń obiektowych (węzłów sieciowych), rozszerzeniu i zróżnicowaniu uległ również ich interfejs komunikacyjny. W celu zagwarantowania współpracy urządzeń różnych producentów (ang. *interoperability*), wprowadzono tzw. „profile” urządzeń, czyli definicję funkcjonalną pola danych. Dzięki wspólnemu interfejsowi, na poziomie sprzętowym i reprezentacji danych, użytkownik wykorzystuje urządzenia automatyki jak „klocki”.

Podczas projektowania systemu nie musi uwzględniać szczegółów implementacyjnych interfejsu, lecz funkcjonalność typu urządzenia, np.: enkoder, panel operatorski, napęd, etc.

Akceptacja rozwiązań komunikacyjnych cyklicznego odwzorowywania pamięci, oraz chęć coraz szerszego wprowadzania do integracji poziomej jak i pionowej, wymusiła rozszerzenie funkcjonalności komunikacji *fieldbus*, o usługi acykliczne (DPV1), komunikację skrośną i szybką komunikację taktowaną zegarem z interpolatora centralnego do napędów (PROFIDRIVE), funkcje gwarantujące bezpieczeństwo na poziomie SIL3, SIL4 (PROFISafe). Równolegle z rozwojem cyklicznej komunikacji typu *master-slave*, rozwijano rozwiązania komunikacyjne tzw. zdarzeniowe. Przykładem jest CAN (ang. Controller Area Network), gdzie każdy węzeł jest uprawniony do przesyłania danych, jeśli zajdzie taka potrzeba. Koncepcja, automatyzacji rozproszonej na bazie CAN, zyskała głównie entuzjazm ośrodków uniwersyteckich, przemysł stosując sieci na bazie CAN, takie jak DeviceNet, CANopen, etc.,

wykorzystuje w dominującej części usługi komunikacji cyklicznej. Jest to powodowane **brakiem metod i narzędzi ułatwiających projektowanie i zagwarantowanie wymagań czasowych**. W przypadku mniej krytycznych czasowo aplikacji jak np. automatyzacja budynków, tzw. „inteligentne budynki”, komunikacja zdarzeniową na bazie sieci LON znajduje coraz szersze zastosowanie.



Rys. 1.4 Ewolucja architektur sterowania [PNO 01]

Zalety stosowania produktów rynku masowego do automatyzacji wytwarzania, tj. komputerów PC, biurowych systemów operacyjnych, skłaniają do coraz intensywniejszych prób wprowadzenia sieci komunikacji biurowej LAN (ang. *Local Area Network*), tj. Ethernetu, protokołu TCP/IP i usług Internetowych do komunikacji na poziomie procesu. Jako podstawowe argumenty wymienia się niską cenę takiego rozwiązania, łatwą integrację z poziomem komunikacyjnym zarządzania przedsiębiorstwem, **bogatą bazę doświadczeń w zakresie sprzętu i oprogramowania**. Wadami powyższego rozwiązania są: brak deterministyczności, niska efektywność protokołu i brak definicji warstwy aplikacyjnej.

Ważnym krokiem dla rozwoju Ethernetu okazały się szybkie urządzenia przełączające typu „switch” oraz wprowadzenie komunikacji dwukierunkowej (ang. *ful-duplex*). Dzięki nim architektura sieci przemieniła się z magistrali w gwiazdę. Switch pełni rolę systemu operacyjnego czasu rzeczywistego, który dla każdego węzła przydziela odpowiednie pasmo transmisyjne. W ten sposób, możliwe jest zagwarantowanie wymagań deterministyczności czasowej dla komunikacji na poziomie procesu wytwarzania [Skura 02].

Dążenie do wprowadzenia Internetu na poziom sterowania wynika również z chęci **wykorzystania tanich i znanych, ogólnie dostępnych narzędzi Internetowych**, jak np.: przeglądarki WEB oraz usług: HTTP, FTP i SMTP (e-mail), czyli warstwy aplikacyjnej. Projekty przyszłej automatyzacji badają urządzenia obiektowe wyposażone w WEB serwery, które **udostępniają narzędzia konfiguracyjne i dokumentację**. Przy tak częstej modyfikacji oprogramowania firmowego, a więc funkcjonalności, jest to uznawane za jedyny **sposób zagwarantowania bezproblemowej ich obsługi**. Mimo, że usługi internetowe (FTP, WWW, e-mail) ciągle jeszcze nie oferują funkcjonalności obsługi danych procesowych (stany, alarmy, diagnostyka), to wspólne ustalenia producentów urządzeń automatyki w tej sprawie są opieszale. Sytuacja przypomina „wojnę sieci fieldbus lat 90.”, gdy pojawiło się szereg konkurencyjnych propozycji, nieznacznie różniących się technicznie. Ostatecznie na rynku pozostały rozwiązania jedynie najsilniejsze, których stabilność jest gwarantowana poprzez standaryzację. Przewodzące organizacje użytkowników sieci *fieldbus* takie jak PNO (niem. *Profibus Nutzer Organization*), InterbusClub, chcąc zachować klientów, proponują odwzorowanie usług własnych sieci /IP. Zupełnie nowatorskie rozwiązania komunikacyjne,

jak np. JetWeb [JetWeb] zrywające z „zaszłością fieldbus”, z trudem zdobywają rynek zdominowany przez dużych producentów urządzeń automatyki.

Integracja z wykorzystaniem sieci *fieldbus* i LAN, mimo heterogeniczności, prowadzi do konsolidacji i zacierania granic między warstwami i poziomami przedsiębiorstwa.

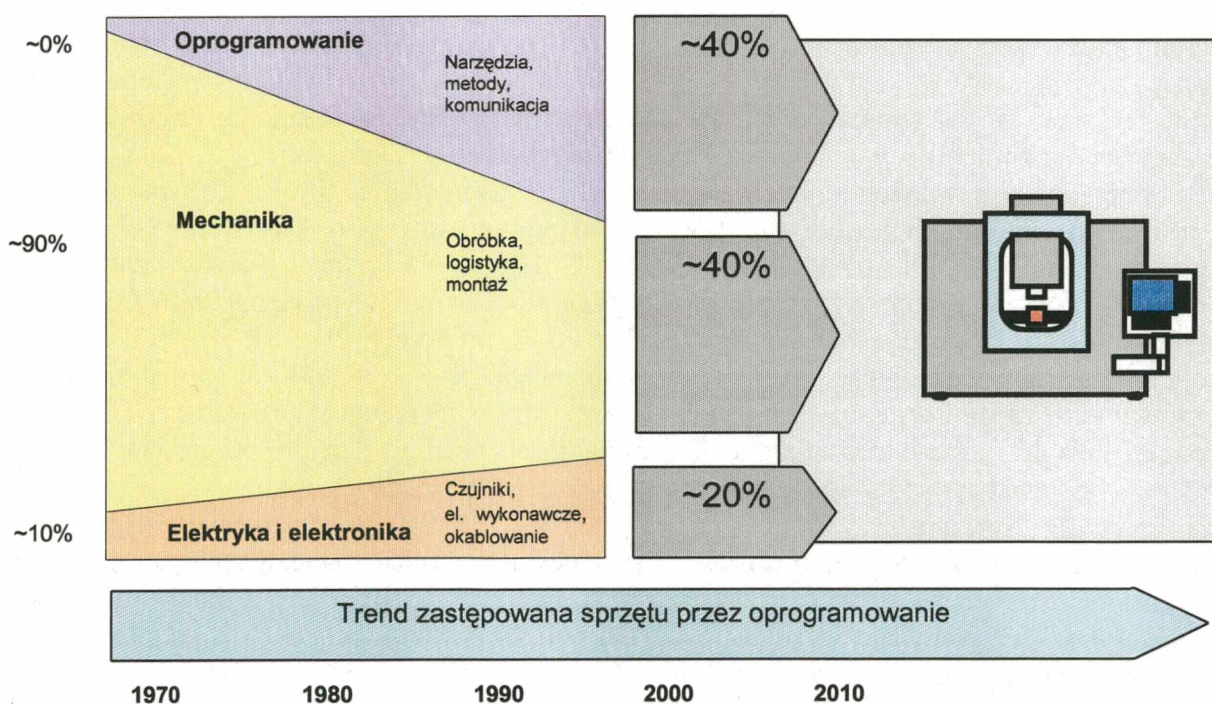
Miniaturyzacja i spadek cen węzłów sterowniczych sprawia, że „inteligentne moduły obiektowe” charakteryzuje coraz większa funkcjonalność i autonomia działania. Węzły te w połączeniu z siecią komunikacyjną są podstawą rozproszonych systemów sterowania.

Dzięki nim staje się możliwe projektowanie systemów wyższej Efektywne projektowanie zdecentralizowanych systemów sterowania, ich serwisowanie i rozwój wymagają jednak specjalnych umiejętności od inżynierii.

Otwartość systemów jest zapewniana dzięki standaryzacji interfejsów, czyli systemów komunikacyjnych. Dawna niemiecka norma DIN19245, obejmująca specyfikację PROFIBUS, została zastąpiona normą europejską EN50170 obejmującą standardy PROFIBUS, FIP i P-NET. Aktualnie trwają prace nad przyjęciem normy międzynarodowej dla standardów komunikacyjnych IEC61158 (ang. *Communication Layers*) oraz IEC61784 (ang. *Communication Profiles*). Nowe normy będą obejmowały aż dziesięć specyfikacji standardów komunikacyjnych, tworząc 4500 stronicowy dokument! Historia standardów komunikacyjnych MMS i FMS pokazała, że zbyt rozbudowana specyfikacja może hamować rozwój i upowszechnianie rozwiązań technicznych. W takiej sytuacji inwestycje na inżynierię, czyli wiedzę i umiejętności (ang. *know-how*) stają się dominującym składnikiem kosztów.

1.3 Realizacja automatyzacji i jej użytkowanie

Stosowanie sterowników, komputerów, „inteligentnych modułów obiektowych” i sieci komunikacyjnych, scharakteryzowanych w poprzednim rozdziale, uzasadnia rosnący udział kosztów oprogramowania w zautomatyzowanych systemach wytwórczych. Bender i Pritschow wskazują, że **udział oprogramowania zrównuje się z mechaniką i osiąga 40%** (rys. 1.5).



Rys. 1.5 Tendencja zastępowania sprzętu przez oprogramowanie [Pritschow 00]

Oprogramowanie sterujące dla systemu wytwórczego, w przeciwieństwie do automatyzacji biura, nie jest jednak produktem masowym, lecz rozwiązaniem specyficznym, które odzwierciedla proces technologiczny [Koch, Weck 96]. Z tego względu, nie jest możliwy wybór oprogramowania „z półki”, które odzwierciedlałoby wymagania procesu technologicznego.

Coraz dobitniej podkreśla się, że tworzenie oprogramowania wykorzystywanego w sterowaniu, jest znaczącą częścią maszyn, i dlatego wymaga ono systematycznych procedur [Storr 95] oraz metodologii takich jak Inżynieria Oprogramowania (ang. *Software Engineering*).

„Produkcja oprogramowania” biurowego, mająca dużo dłuższą tradycję i lepsze wyposażenie metodyczne i narzędziowe, jak programowanie sterowników, dostarcza przerażających faktów:

„Windows 2000 kosztował 1 miliard \$, (...) w dniu premiery miał zauważonych już ok. 63.000 błędów” [Reuters 00]

Stąd weryfikacja oprogramowania podczas fazy użytkowania i konieczność ciągłych poprawek i rozszerzeń dla polepszenia bezpieczeństwa danych i stabilności systemu. Brak możliwości całkowitego przetestowania oprogramowania wynika głównie z jego złożoności. Quality Systems & Software [QSS 98] podaje jednocześnie, że aż 45% funkcji zaimplementowanych w systemach programistycznych nie jest użytych ani razu, co dowodzi, że **wymagania użytkowników i dziedzina problemu nie są należycie znane.**

W ramach projektu EPS SES [EPSSSES] „Opracowanie, produkcja i serwisowanie oprogramowania dla systemów wbudowanych dla produkcji” (niem. „*Entwicklung, Produktion und Service von Software für eingebettete Systeme in der Produktion*”), dokonano obszernej analizy procesów opracowywania oprogramowania typu *embedded*. Projekt ten był prowadzony przez sześć niemieckich instytutów naukowych oraz sześć największych niemieckich firm m.in. SIEMENS, Bosch, DaimlerChrysler. W celu uchwycenia stanu zagadnienia w roku 1999, przeprowadzono ankietę, w której wypowiedziało się 40 firm o specjalnościach: automatyzacja, budowa maszyn i systemy produkcyjne, elektrotechnika i technika pomiarowa, oprogramowanie oraz transport. Pytania dotyczące pracy grupowej, cykli życia projektów i dokumentacji wskazały na szereg problemów. Na podstawie raportu z projektu dokonano obszernego skrótu, który zamieszczono w Tabeli 1.3 „Problemy przy opracowywaniu systemów *embedded*”.

Tabela 1.3 Problemy przy opracowywaniu systemów *embedded* [jr na podstawie EPSSSES]

Stan obecny	Problemy
Praca grupowa	
• Zespoły interdyscyplinarne, różna terminologia, kłopoty komunikacyjne dla dużych projektów • Małe grupy projektowe - nieformalny przepływ informacji, niewielkie problemy • Duże grupy projektowe – problemy ze spójnością formatów danych, brak mechanizmów zgodności wersji • Kierownictwo zespołów wywodzi się zwykle z dawnych kierunków studiów, ma słabe rozumienie specyfiki oprogramowania	• Kłopoty komunikacyjne przy dużych projektach, szczególnie, gdy kierownictwo nie jest odpowiednio przeszkolone • Kłopoty komunikacyjne powiększa terytorialne oddalenie partnerów oraz dla grup międzynarodowych cechy mentalności • Szkolenie kadry z zakresu inżynierii oprogramowania systemów wbudowanych jest niedostateczne. • Braki w zakresie organizacyjnym i metodycznym są największe

Modele postępowania	
• Typowo stosowany jest model fazowy: analiza i specyfikacja; projektowanie architektury systemu i komponentów; implementacja HW i SW; zapewnienie jakości i testowanie; wdrożenie, serwisowanie i utrzymanie • Większość firm koncentruje się na implementacji i testowaniu • Najczęściej stosowany model „wodospadowy” lub <i>top-down</i>, rzadko „V-model” lub „cykliczny” • Opis HW dość kompletny, SW raczej przypadkowy • HW wykorzystuje standardowe komponenty, SW czasami ponowne wykorzystanie kodu • Niewielkie zastosowanie abstrakcyjnych modeli dla opisu SW • Braki zgodności w opracowywaniu HW i SW	• Brak odpowiednich standaryzowanych modeli postępowania • Problemy szacowania kosztów i zasobów szczególnie dla projektów o cechach funkcjonalności indywidualnej • Braki w inżynierskich standaryzowanych sposobach opisu • Szczątkowe stosowanie komponentów • Niewielkie stosowanie standaryzowanych architektur SW i HW • Późne uwzględnianie wymagań dla oprogramowania
Analiza wymagań	
• Opracowywanie specyfikacji wymagań bez zaleceń struktury • Czasami specyfikacja zobowiązań obejmująca jedynie HW • Bardzo nieprecyzyjne określenie funkcjonalności SW • Dokumenty analizy nie obejmują nadrzędnie ujęcia HW, SW, QA	• Niepełny opis wymagań dla oprogramowania • Brak podstaw dla kontraktu/spec. zobowiązań • Mała przydatność dla QA • Trudność analizy spójności • Brak systematyki tworzenia spec. zobowiązań • Bardzo ograniczone ponowne wykorzystanie w fazie analizy
Projektowanie	
• Faza projektowania zwykle nie jest zintegrowana z innymi etapami projektu • Projektowanie często ogranicza się do architektury systemu (ogólnie) • Dokumenty projektowe bardzo ogólnie opisują SW • Zastosowanie wzorców projektowych bardzo rzadkie • Ponowne wykorzystanie ograniczone do architektury systemu • Generowanie i prototypowanie prawie nie jest stosowane	• Brakująca systematyka dla prowadzenia fazy projektowania • Brak metod i narzędzi opisu, a w szczególności: -brak mechanizmów kontroli konsystencji -niewielki udział generacji (symulacja prototypy) -niewielka umiejętność realizacji zaleceń QA • Brak możliwości wstecznego śledzenia dokumentacji projektowej aż do specyfikacji wymagań • Brak systematycznego wykorzystania komponentów
Implementacja	
• Standardowe środowiska programistyczne znajdują szerokie zastosowanie • Zalecenia implementacyjne ustalane są firmowo • Ponowne wykorzystanie ogranicza się do parametryzowanych fragmentów kodu • Automatyczna generacja kodu jest niemal nieużywana • Systemy operacyjne czasu rzeczywistego znajdują niewielkie użycie	• Zachowanie zaleceń programowania nie jest konsekwentne • Ponowne wykorzystanie ogranicza się jedynie do niemethodycznego użycia kodów • Automatyczna generacja kodu, z powodów wydajnościowych lub niedopasowania do platformy, nie jest stosowana • Definicja faz i przejść jest tylko częściowa • Systemy operacyjne RT znajdują niewielkie zastosowanie (koszty, skomplikowanie)
Serwisowanie i utrzymanie	
• Bardzo różne traktowanie zagadnienia, od bardzo istotnego, do zupełnie nieistotnego • Dla systemów <i>high-tech</i> stosowany jest telemonitoring, teleserwis • Dla niektórych aplikacji archiwa działania i nieprawidłowości tzw. <i>logfile</i> • Dla najprostszych systemów brak możliwości analizy przyczyn niesprawności	• Błędy i diagnostyka uchwyczone podczas eksploatacji często nie są analizowane i nie trafiają do działu opracowania • Częste zmiany wersji oprogramowania są trudne do ogarnięcia przez techników • Serwis i utrzymanie powierza się osobom, które nie są odpowiednio przeszkolone

Zapewnienie jakości	
- Ograniczone planowanie i brak metodyki systematycznego zapewnienia jakości - Brak kompetentnych działów testowania, programista testuje sam, - Kontrola wydajności czasowej jest prowadzona, jednak brak dla nich metod i narzędzi - Niezbędne systematyczne rewizje projektu traktowane jako dowód zgodności QA ze specyfikacją wymagań - Testowanie jako dynamiczny środek QA prowadzony w różnych fazach - Brak rozumienia potrzeby poprawy procesu opracowywania - Zastosowanie narzędzi do testowania bardzo ograniczone	- Brak systemowego planowania i prowadzenia QA - Przedłużanie projektów skraca fazę testowania - Rzadko istnieje wydzielona komórka QA - Opracowywanie planów testowych jest niemethodyczne i zależne od doświadczenia - Brak odpowiednich i jednolitych środków opisowych dla planów testów - Narzędzia do testowania spełniają wymagania systemów wbudowanych tylko częściowo - Niedostateczna możliwość dopasowania narzędzi testowania do procesu opracowywania i warunków firmy
Zarządzanie konfiguracją	
- Dla małych projektów (3 osoby) żadnych narzędzi zarządzania konfiguracją - Dla średnich projektów (30 osób) proste narzędzia CVS - Dla dużych projektów (100 osób) ClearCase - Zarządzaniu podlega kod źródłowy i ewentualnie dokumenty projektowe - Brak nadrzędnej kontroli konsystencji typów dokumentów - Zastosowanie nieodpowiednich formatów danych bardzo utrudnia konfigurację	- Częściowe braki centralnych kartotek projektowych i systemów archiwizacji - Nie wszystkie fazy są obejmowane - Stare projekty nie są wprowadzane do systemu - Trudne zarządzanie prawami dostępu dla pojedynczych dokumentów - Trudność koordynacji dokumentów nadbudowywanych - Brak skrzynek informacyjnych Analizy całościowe projektu są niemożliwe Zarządzanie konfiguracją zorientowane jest na działy organizacyjne
Techniki i narzędzia opisu (dokumentacji)	
- Wybór narzędzi prezentacji i technik opisu zależy od oceny projektanta - Techniki opisu są stosowane głównie podczas analizy i projektu ogólnego - Faworyzowane techniki opisu to: automat stanów, przedstawienie architektury i diagram klas - Modularyzacja opracowywania następuje jedynie nieformalnie - Brak efektywnego użycia generatorów kodu - Wymagania czasowe odgrywają dużą rolę, nie są jednak metodycznie rozważane	Różnorodne wymagania użytkowników dot. techniki opisu Niedostateczne informacje o możliwościach Decyzja o stosowanych technikach opisu zależna od projektanta - Niedostateczne wsparcie narzędzi komputerowych - Niedostateczna integracja narzędzi - Wysokie koszty wprowadzenia i utrzymania - Niedostateczne rozpoznanie rynku narzędzi - Niska jakość narzędzi Niedostatek narzędzi do analizy i testowania - Brak możliwości nadrzędnego śledzenia etapów projektu Przeciążenie fazy szczegółowego projektowania i implementacji Problemy jakości, niewielkie ponowne użycie, wysokie koszty

Powyższe studium dowodzi, że to **niedostatki organizacyjne i metodyczne, dotyczące wszystkich faz i etapów rozwoju oprogramowania, a nie techniczne, są przyczyną kłopotów podczas realizacji projektów systemów wbudowanych.**

Badania Standish Group w roku 1999, przeprowadzone na 352 firmach programistycznych [Shannon et al. 99] wykazały, że 56% wszystkich błędów ma przyczynę w specyfikacji wymagań. Usuwanie błędów poczynionych w specyfikacji wymagań jest najkosztowniejsze, a niejednokrotnie „rujnuje całe przedsięwzięcie”.

Jako najistotniejsze przyczyny przekroczeń budżetów, terminów realizacji, a nawet przerywania projektów, identyfikowane są problemy związane z odbiorcą produktu oraz realizacją. Nierealistyczne i zmieniające się wymagania, niskie zaangażowanie, brak wsparcia

Zapewnienie jakości	
- Ograniczone planowanie i brak metodyki systematycznego zapewnienia jakości - Brak kompetentnych działów testowania, programista testuje sam, - Kontrola wydajności czasowej jest prowadzona, jednak brak dla nich metod i narzędzi - Niezbyt systematyczne rewizje projektu traktowane jako dowód zgodności QA ze specyfikacją wymagań - Testowanie jako dynamiczny środek QA prowadzony w różnych fazach - Brak rozumienia potrzeby poprawy procesu opracowywania - Zastosowanie narzędzi do testowania bardzo ograniczone	- Brak systemowego planowania i prowadzenia QA - Przedłużanie projektów skraca fazę testowania - Rzadko istnieje wydzielona komórka QA - Opracowywanie planów testowych jest niemethodyczne i zależne od doświadczenia - Brak odpowiednich i jednolitych środków opisowych dla planów testów - Narzędzia do testowania spełniają wymagania systemów wbudowanych tylko częściowo - Niedostateczna możliwość dopasowania narzędzi testowania do procesu opracowywania i warunków firmy
Zarządzanie konfiguracją	
- Dla małych projektów (3 osoby) żadnych narzędzi zarządzania konfiguracją - Dla średnich projektów (30 osób) proste narzędzia CVS - Dla dużych projektów (100 osób) ClearCase - Zarządzaniu podlega kod źródłowy i ewentualnie dokumenty projektowe - Brak nadrzędnej kontroli konsystencji typów dokumentów - Zastosowanie nieodpowiednich formatów danych bardzo utrudnia konfigurację	- Częściowe braki centralnych kartotek projektowych i systemów archiwizacji - Nie wszystkie fazy są obejmowane - Stare projekty nie są wprowadzane do systemu - Trudne zarządzanie prawami dostępu dla pojedynczych dokumentów - Trudność koordynacji dokumentów nadbudowywanych - Brak skrzynek informacyjnych Analizy całościowe projektu są niemożliwe Zarządzanie konfiguracją zorientowane jest na działy organizacyjne
Techniki i narzędzia opisu (dokumentacji)	
- Wybór narzędzi prezentacji i technik opisu zależą od oceny projektanta - Techniki opisu są stosowane głównie podczas analizy i projektu ogólnego - Faworyzowane techniki opisu to: automat stanów, przedstawienie architektury i diagram klas - Modularyzacja opracowywania następuje jedynie nieformalnie - Brak efektywnego użycia generatorów kodu - Wymagania czasowe odgrywają dużą rolę, nie są jednak metodycznie rozważane	Różnorodne wymagania użytkowników dot. techniki opisu Niedostateczne informacje o możliwościach Decyzja o stosowanych technikach opisu zależna od projektanta - Niedostateczne wsparcie narzędzi komputerowych - Niedostateczna integracja narzędzi - Wysokie koszty wprowadzenia i utrzymania - Niedostateczne rozpoznanie rynku narzędzi - Niska jakość narzędzi Niedostatek narzędzi do analizy i testowania - Brak możliwości nadrzędnego śledzenia etapów projektu Przeciążenie fazy szczegółowego projektowania i implementacji Problemy jakości, niewielkie ponowne użycie, wysokie koszty

Powyższe studium dowodzi, że to **niedostatki organizacyjne i metodyczne, dotyczące wszystkich faz i etapów rozwoju oprogramowania, a nie techniczne, są przyczyną kłopotów podczas realizacji projektów systemów wbudowanych.**

Badania Standish Group w roku 1999, przeprowadzone na 352 firmach programistycznych [Shannon et al. 99] wykazały, że 56% wszystkich błędów ma przyczynę w specyfikacji wymagań. Usuwanie błędów poczynionych w specyfikacji wymagań jest najkosztowniejsze, a niejednokrotnie „rujnuje całe przedsięwzięcie”.

Jako najistotniejsze przyczyny przekroczeń budżetów, terminów realizacji, a nawet przerywania projektów, identyfikowane są problemy związane z odbiorcą produktu oraz realizacją. Nierealistyczne i zmieniające się wymagania, niskie zaangażowanie, brak wsparcia

1.4 Podsumowanie rozdziału pierwszego

Wymagania stawiane systemom produkcyjnym są odwzorowaniem oczekiwań klientów wobec produktów. Stąd też konieczność redukcji kosztów produkcji, zwiększania produktywności, elastyczności i skracania czasów produkcji, które zostały ostatnio rozszerzone o poszukiwanie unikatowości wyrobów, możliwość indywidualnego ich kształtowania, a nawet ich zdolność do uczenia się. Spełnianie powyższych wymagań wymaga coraz bardziej wyrafinowanych systemów produkcyjnych, w których automatyzacja odgrywa dominującą rolę.

Konieczność zapewnienia różnorodności produktów oraz unowocześnianie procesu technologicznego wprowadzają dodatkowy wymóg szybkiej odpowiedzi na zmieniające się wymagania, określanej jako „zwinność zautomatyzowanego systemu wytwórczego”.

Zastosowanie najnowocześniejszych środków technicznych dla automatyzacji, takich jak sterowniki PLC, komputery PC, „inteligentne” moduły obiektowe i sieci komunikacyjne, wymaga od projektantów wysokich kompetencji technicznych, jak również społecznych. Bez odpowiedniej wiedzy projektantów, nowoczesne rozwiązania techniczne mogą zagrażać bezpieczeństwu ludzi, instalacji czy środowiska. Kompetencje społeczne są zaś podstawą pracy grupowej oraz zidentyfikowania oczekiwań klienta, co jest wskazywane jako podstawa sukcesu biznesowego.

Bender i Pritschow wskazują, że udział kosztowy oprogramowania i mechaniki zrównują się osiągając 40%, w takiej sytuacji o jakości systemu wytwórczego decyduje w coraz większym stopniu proces tworzenia oprogramowania. Jak wykazała analiza, praktyka inżynierii oprogramowania wskazuje na wiele braków i niedostatków głównie organizacyjnych. Wśród nich zidentyfikowano grupę problemów dotyczących komunikacji, tj. wymiany wiedzy wewnątrz zespołu projektowego oraz z klientem. Wskazany problem jest przyczyną niekompletności wymagań, niskiego zaangażowania klienta i jego nierealistycznych wymagań oraz zmiany oczekiwań. Analiza cyklu życia projektów potwierdziła potrzebę intensyfikacji początkowych faz opracowywania oprogramowania, a więc i automatyzacji wytwarzania, celem redukcji kosztów związanych z usuwaniem błędów.

2 Postawienie problemu pracy doktorskiej

2.1 Obszar pracy, beneficjenci

W wyniku studium przeprowadzonego w rozdziale 1, zidentyfikowano zagadnienie **komunikacji wiedzy w automatyzacji między jej uczestnikami**, jako mający istotny wpływ na efektywność projektów automatyzacji. Problem efektywności wprowadzania, utrzymywania i rozwijania zautomatyzowanych systemów nabiera rosnącego znaczenia wraz ze wzrostem skali zakresu automatyzacji i wzrostem częstości jej prowadzenia. Powodowane jest to liczebniejszym zespołem projektowym a tym samym rozproszeniem wiedzy. Komunikacja wiedzy dotyczy zarówno zespołów projektowych (komunikacja wewnętrzna) jak również relacji użytkownik (technolog) i dostawca (projektant) określanej jako komunikacja zewnętrzna. W obu przypadkach partnerzy komunikacji reprezentują różne kompetencje techniczne.

Beneficjenci:

- Użytkownicy automatyzacji: technolodzy, „jakościowcy”, serwisanci, planiści
- Dostawcy automatyzacji: analitycy, projektanci, programiści

2.2 Cele pracy

Strategicznym celem realizowanej pracy doktorskiej jest poprawa efektywności automatyzacji wytwarzania. Dla osiągnięcia założonego celu przewiduje się:

Cele pracy doktorskiej:

- Przeanalizować procesy automatyzacji w ujęciu wiedzy,
- Opracować narzędzie zarządzania wiedzą dla poprawy efektywności automatyzacji,
- Przeanalizować proces wdrażania opracowanego narzędzia i jego utrzymywania,
- Poddać weryfikacji opracowane narzędzie w środowisku praktyki przemysłowej

Cele szczegółowe dla poszukiwanego „narzędzia” skupiają się na poprawie stanu współpracy zespołów interdyscyplinarnych, a w szczególności rozumienia wymagań i rozwiązań, komunikacji międzyludzkiej i poprawie dokumentacji. Są one zatem pre-rekwizytami zarządzania wiedzą.

2 Postawienie problemu pracy doktorskiej

2.1 Obszar pracy, beneficjenci

W wyniku studium przeprowadzonego w rozdziale 1, zidentyfikowano zagadnienie **komunikacji wiedzy w automatyzacji między jej uczestnikami**, jako mający istotny wpływ na efektywność projektów automatyzacji. Problem efektywności wprowadzania, utrzymywania i rozwijania zautomatyzowanych systemów nabiera rosnącego znaczenia wraz ze wzrostem skali zakresu automatyzacji i wzrostem częstości jej prowadzenia. Powodowane jest to liczniejszym zespołem projektowym a tym samym rozproszeniem wiedzy. Komunikacja wiedzy dotyczy zarówno zespołów projektowych (komunikacja wewnętrzna) jak również relacji użytkownik (technolog) i dostawca (projektant) określanej jako komunikacja zewnętrzna. W obu przypadkach partnerzy komunikacji reprezentują różne kompetencje techniczne.

Beneficjenci:

- Użytkownicy automatyzacji: technolodzy, „jakościowcy”, serwisanci, planiści
- Dostawcy automatyzacji: analitycy, projektanci, programiści

2.2 Cele pracy

Strategicznym celem realizowanej pracy doktorskiej jest poprawa efektywności automatyzacji wytwarzania. Dla osiągnięcia założonego celu przewiduje się:

Cele pracy doktorskiej:

- Przeanalizować procesy automatyzacji w ujęciu wiedzy,
- Opracować narzędzie zarządzania wiedzą dla poprawy efektywności automatyzacji,
- Przeanalizować proces wdrażania opracowanego narzędzia i jego utrzymywania,
- Poddać weryfikacji opracowane narzędzie w środowisku praktyki przemysłowej

Cele szczegółowe dla poszukiwanego „narzędzia” skupiają się na poprawie stanu współpracy zespołów interdyscyplinarnych, a w szczególności rozumienia wymagań i rozwiązań, komunikacji międzyludzkiej i poprawie dokumentacji. Są one zatem pre-rekwizytami zarządzania wiedzą.

Poprawa rozumienia wymagań technologii i jakości systemów wytwórczych przez dostawców rozwiązań automatyzacji (automatyków)

Beneficjentami przyszłego zautomatyzowanego systemu wytwarzania są jego użytkownicy (operatorzy i inżynierowie ruchu i planiści). Zwykle jednak, ich uczestnictwo w procesie automatyzacji ogranicza się wyłącznie do ogólnego nakreślenia wymagań i końcowego użytkowania rozwiązania. W takiej sytuacji weryfikacja ich wymagań jest bardzo późna.

Jak wskazano w rozdziale 1.3, aż 45% funkcjonalności oprogramowania nie jest wykorzystywanych. Oznacza to, że **rozwiązanie zostało „przewymiarowane funkcjonalnie”, więc jest nieekonomiczne**. Czy dostawcy (automatycy) naprawdę wiedzą, jakie są wymagania użytkowników automatyzacji systemu wytwarzania? Jest to bardzo trudne, w sytuacji, gdy wymagania się zmieniają, a wiele potrzeb jest identyfikowanych i formułowanych dopiero podczas „życia systemu”. Jak wskazano na rys. 1.6, koszt usuwania błędów rośnie eksponencjalnie wraz z oddalaniem się od fazy formułowania założeń projektowych, dlatego **początkowe fazy projektu odgrywają szczególną rolę**. Jak wskazano w rozdziale 1.3, **dopasowanie obowiązujących procesów technologicznych i ludzi do systemu informacyjnego jest nieefektywne, a systemy, bardziej zorientowane na technikę, niż na proces i pracę ludzką, traktujące człowieka przedmiotowo, odbierają mu motywację i utożsamianie się z problemem**.

Poprawa komunikacji użytkownik – dostawca rozwiązań automatyki

Wiedza ekspertów, technologów, inżynierów zapewnienia jakości, planistów i serwisantów, a więc użytkowników systemu automatyzacji, uznawana jest za największy kapitałem przedsiębiorstwa. Od każdej z grup **specjalistów wymaga się innych umiejętności (kompetencji) oraz innej odpowiedzialności za proces technologiczny**. Stosowanie dedykowanych metod, narzędzi i norm **utrwała żargon i specyficzne pojmowanie problemów przez pryzmat własnej specjalizacji**. Dodatkowo **lawinowy przyrost informacji**, wspierany technologią informatyczną, powoduje jej powielanie i niespójności. **Technologia informatyczna**, taka jak serwery plików, bazy danych, poczta e-mail są tylko narzędziami, które **nie zapewniają jednak, że specjaliści z różnymi kompetencjami potrafią „się dogadać”**.

Poprawa jakości dokumentacji o wymaganiach i rozwiązaniu automatyzacji

Konieczność skracania czasów cyklu rozwoju produktu (ang. *time-to-market*) i jego wdrażania (ang. *time-to-customer*) oraz poszukiwanie oszczędności, negatywnie wpływają na **dokumentację projektową**. Ponieważ jej **jakość nie jest możliwa do obiektywnej oceny**, stąd częste sprowadzanie **dokumentacji do plików z kodami źródłowymi**. Kod źródłowy, z uwagi na swoją szczegółowość i obszerność, nie jest jednak efektywnym źródłem informacji o koncepcji systemu, jego architekturze, funkcjonalności, mechanizmach bezpieczeństwa i gwarantowanej wydajności.

Bez rzetelnej dokumentacji lub wiedzy o zastosowanych rozwiązaniach na poziomie założeń projektowych i zastosowanych rozwiązań, modyfikacja i rozbudowa istniejącego zautomatyzowanego systemu wytwórczego nie będzie efektywna, może zagrażać bezpieczeństwu, a nawet może okazać się niemożliwa.

Poprawa rozumienia wymagań technologii i jakości systemów wytwórczych przez dostawców rozwiązań automatyzacji (automatyków)

Beneficjentami przyszłego zautomatyzowanego systemu wytwarzania są jego użytkownicy (operatorzy i inżynierowie ruchu i planiści). Zwykle jednak, ich uczestnictwo w procesie automatyzacji ogranicza się wyłącznie do ogólnego nakreślenia wymagań i końcowego użytkowania rozwiązania. W takiej sytuacji weryfikacja ich wymagań jest bardzo późna.

Jak wskazano w rozdziale 1.3, aż 45% funkcjonalności oprogramowania nie jest wykorzystywanych. Oznacza to, że **rozwiązanie zostało „przewymiarowane funkcjonalnie”, więc jest nieekonomiczne**. Czy dostawcy (automatycy) naprawdę wiedzą, jakie są wymagania użytkowników automatyzacji systemu wytwarzania? Jest to bardzo trudne, w sytuacji, gdy wymagania się zmieniają, a wiele potrzeb jest identyfikowanych i formułowanych dopiero podczas „życia systemu”. Jak wskazano na rys. 1.6, koszt usuwania błędów rośnie eksponencjalnie wraz z oddalaniem się od fazy formułowania założeń projektowych, dlatego **początkowe fazy projektu odgrywają szczególną rolę**. Jak wskazano w rozdziale 1.3, **dopasowanie obowiązujących procesów technologicznych i ludzi do systemu informacyjnego jest nieefektywne, a systemy, bardziej zorientowane na technikę, niż na proces i pracę ludzką, traktujące człowieka przedmiotowo, odbierają mu motywację i utożsamianie się z problemem**.

Poprawa komunikacji użytkownik – dostawca rozwiązań automatyki

Wiedza ekspertów, technologów, inżynierów zapewnienia jakości, planistów i serwisantów, a więc użytkowników systemu automatyzacji, uznawana jest za największy kapitałem przedsiębiorstwa. Od każdej z grup **specjalistów wymaga się innych umiejętności (kompetencji) oraz innej odpowiedzialności za proces technologiczny**. Stosowanie dedykowanych metod, narzędzi i norm **utrwała żargon i specyficzne pojmowanie problemów przez pryzmat własnej specjalizacji**. Dodatkowo **lawinowy przyrost informacji**, wspierany technologią informatyczną, powoduje jej powielanie i niespójności. **Technologia informatyczna**, taka jak serwery plików, bazy danych, poczta e-mail są tylko narzędziami, które **nie zapewniają jednak, że specjaliści z różnymi kompetencjami potrafią „się dogadać”**.

Poprawa jakości dokumentacji o wymaganiach i rozwiązaniu automatyzacji

Konieczność skracania czasów cyklu rozwoju produktu (ang. *time-to-market*) i jego wdrażania (ang. *time-to-customer*) oraz poszukiwanie oszczędności, negatywnie wpływają na **dokumentację projektową**. Ponieważ jej **jakość nie jest możliwa do obiektywnej oceny**, stąd częste sprowadzanie **dokumentacji do plików z kodami źródłowymi**. Kod źródłowy, z uwagi na swoją szczegółowość i obszerność, nie jest jednak efektywnym źródłem informacji o koncepcji systemu, jego architekturze, funkcjonalności, mechanizmach bezpieczeństwa i gwarantowanej wydajności.

Bez rzetelnej dokumentacji lub wiedzy o zastosowanych rozwiązaniach na poziomie założeń projektowych i zastosowanych rozwiązań, modyfikacja i rozbudowa istniejącego zautomatyzowanego systemu wytwórczego nie będzie efektywna, może zagrażać bezpieczeństwu, a nawet może okazać się niemożliwa.

Ponieważ zaproponowany zestaw umiejętności (kompetencji) dotyczy pracy ludzkiej, przewiduje się, że proces rozwijania powyższych kompetencji wymagać będzie zarządzania. W przeciwnym razie „nowe kompetencje” nie zostaną efektywnie wdrożone.

Na podstawie powyższych wniosków postawiono następującą tezę pracy doktorskiej:

Teza pracy doktorskiej:

„Poprawę efektywności automatyzacji wytwarzania można osiągnąć dzięki odpowiednim kompetencjom w zakresie modelowania obiektowego i języka UML oraz poprzez odpowiednie nimi zarządzanie”

Rozprawa doktorska będzie studium nad problemem efektywności automatyzacji, oraz sformułowaniem zakresu kompetencyjnego dla poprawy efektywności komunikacji w grupach interdyscyplinarnych. Udowodnienie tezy będzie jednocześnie odpowiedzią na cele częściowe postawione wyżej dla poszukiwanego „narzędzia”.

Tak sformułowana teza jest próbą przełamania tradycyjnego sposobu myślenia i postępowania w automatyzacji, które poszukuje poprawy efektywności automatyzacji głównie w rozwoju technicznym.

Modelowanie w automatyzacji jest wykorzystywane głównie jako język implementacji. Automatyzacja rozwija własne podejście do modelowania bazujące na blokach funkcyjnych (IEC61131, IEC61499). Obiektość kojarzona jest z językami programowania obiektowego np. C++ [Strosłup 96], Java, etc, lub „nieprzydatną abstrakcją”. Język UML, stający się standardem *defacto* obiektowego opisu systemów informatycznych, jest jednak wykorzystywany w bardzo ograniczonym stopniu do komunikacji z klientem podczas specyfikacji wymagań.

2.4 Plan struktury dokumentu pracy doktorskiej

Rozdział	Obiekt badań	Problemy i pytania badawcze
1	Automatyzacja wytwarzania	- Jakie wymagania dla automatyzacji stawiają systemy produkcyjne? - Jak środki techniczne wpływają na wymagania i realizację automatyzacji? - Jak przebiega proces automatyzacji?
2	Postawienie problemu pracy doktorskiej	- Jaki problem badawczy powinien być rozwiązany? - Kto będzie beneficjentem rozwiązania? - Jakie cele zdefiniować w pracy? - Jak sformułować tezę? - Jak przeprowadzić rozprawę?
3	Analiza automatyzacji wytwarzania w ujęciu wiedzy (cel tezy: poprawa efektywności automatyzacji wytwarzania)	- Jak zamodelować proces automatyzacji w ujęciu wiedzy? - Od czego zależy i jak jest określana efektywność w automatyzacji? - Jak przebiega i czym charakteryzuje się łańcuch komunikacyjny między klientem a dostawcą podczas automatyzacji?
4	Kompetencje w zakresie modelowania obiektowego i UML dla automatyzacji wytwarzania (pierwszy postulat tezy: odpowiednie kompetencje w zakresie modelowania obiektowego i UML)	- Jak komunikować idee? - Jak przebiega modelowanie? - Jaki paradygmat i języki graficzne znajdują zastosowanie w automatyzacji? - Czym charakteryzuje się paradygmat obiektowy i język UML? - Jak zastosować UML w automatyzacji wytwarzania?
5	Wdrażanie modelowania obiektowego i UML dla automatyzacji (drugi postulat tezy: odpowiednie zarządzanie modelowaniem obiektowym i UML)	- Jak wdrożyć zaproponowane narzędzie zarządzania wiedzą do praktyki inżynierskiej? - Jak użytkować wdrożone narzędzie zarządzania wiedzą?
6	Weryfikacja modelowania obiektowego i UML	- Jak zaproponowane rozwiązanie zostanie przyjęte przez praktyków? - Czy zaproponowana metoda wdrożenia prowadzi do sukcesu? - Jakie użytkowe wyniki osiągnął zespół po wdrożeniu?
7	Wnioski	- Jakie wnioski naukowe wynikają z pracy? - Jakie wnioski użytkowe wynikają z pracy? - Jakie proponowane są dalsze kierunki rozwoju proponowanych rozwiązań?

3 Analiza automatyzacji wytwarzania w ujęciu wiedzy

Dla zbadania słuszności postawionej tezy, w rozdziale 3 skoncentrowano się na jej założeniu:

„Poprawę efektywności automatyzacji wytwarzania można osiągnąć dzięki kompetencjom w zakresie modelowania obiektowego i języka UML oraz poprzez odpowiednie zarządzanie nimi.”

Dla Rozdziału 3 postawiono następujące pytania badawcze:

3.1 Automatyzacja wytwarzania:

- *Czym jest automatyzacja wytwarzania (AW Automatyzacja Wytwarzania)?*
- *Jaki zakres obejmuje i kiedy występuje AW?*
- *Jak przebiega scenariusz realizacji AW?*

3.2 Efektywność automatyzacji

- *Jakie miary określają efektywność AW?*

3.3 Produkty automatyzacji jako nośniki wiedzy

- *Jaką rolę pełnią produkty automatyzacji w przenoszeniu wiedzy?*
- *W jaki sposób interakcja między uczestnikami AW wpływa na jej efektywność?*

3.4 Komunikacja wiedzy między aktorami

- *Co charakteryzuje ogniwa łańcucha komunikacyjnego?*
- *Jaką rolę odgrywa paradygmat w komunikowaniu wiedzy?*

3 Analiza automatyzacji wytwarzania w ujęciu wiedzy

Dla zbadania słuszności postawionej tezy, w rozdziale 3 skoncentrowano się na jej założeniu:

„Poprawę efektywności automatyzacji wytwarzania można osiągnąć dzięki kompetencjom w zakresie modelowania obiektowego i języka UML oraz poprzez odpowiednie zarządzanie nimi.”

Dla Rozdziału 3 postawiono następujące pytania badawcze:

3.1 Automatyzacja wytwarzania:

- *Czym jest automatyzacja wytwarzania (AW Automatyzacja Wytwarzania)?*
- *Jaki zakres obejmuje i kiedy występuje AW?*
- *Jak przebiega scenariusz realizacji AW?*

3.2 Efektywność automatyzacji

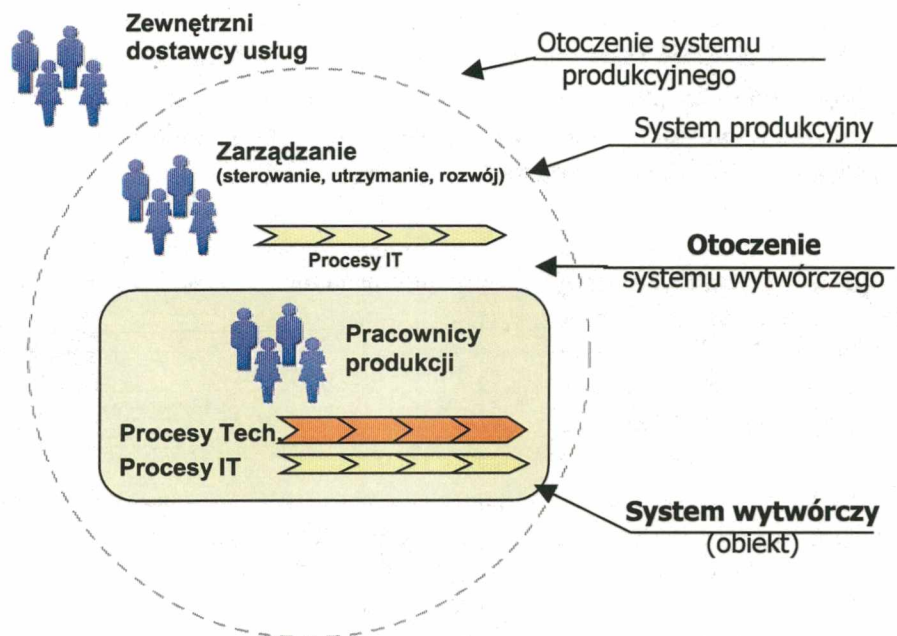
- *Jakie miary określają efektywność AW?*

3.3 Produkty automatyzacji jako nośniki wiedzy

- *Jaką rolę pełnią produkty automatyzacji w przenoszeniu wiedzy?*
- *W jaki sposób interakcja między uczestnikami AW wpływa na jej efektywność?*

3.4 Komunikacja wiedzy między aktorami

- *Co charakteryzuje ogniwa łańcucha komunikacyjnego?*
- *Jaką rolę odgrywa paradygmat w komunikowaniu wiedzy?*



Rys. 3.2 Składniki, zakres i otoczenie systemu wytwórczego [jr]

Jak wskazano na powyższych modelach (rys. 3.1 i 3.2), człowiek stanowi integralną część systemów biznesowych, produkcyjnych czy wytwórczych. Stąd pojęcie systemu socjo-technicznego, będące dostrzeżeniem nie tylko fizycznej, ale również intelektualnej roli człowieka. Współcześnie człowiek nie jest „swoistym wyposażeniem” produkcyjnym, czy też „tanią siłą roboczą”, ale dostrzegane są jego kompetencje, doświadczenie, odpowiedzialność, współpraca, motywacja, zaangażowanie, etc. Autorzy i³ [Eversheim, et al. 02] zadają filozoficzne pytanie: „czy człowiek jest przedłużeniem ramienia robota, czy robot przedłużeniem ramienia człowieka?” Podnoszą w ten sposób problem nadrzędności w podejmowaniu decyzji oraz odpowiedzialności za jakość, niezawodność i bezpieczeństwo. Problemy filozofii technicznej, psychologii, socjologii i inżynierii przeplatają się wzajemnie odkrywając bardzo złożony świat systemów socjo-technicznych dla inżyniera. Czy aparat techniczny inżynierii jest wystarczający do zapanowania nad skomplikowaniem a może nawet złożonością takich systemów?

„Wyzwaniem za 20 lat nie będzie szybkość, koszty czy wydajność, ale problem kompleksowości” (Bill Raduchel, Chief Strategy Officer, Sun Microsystems)

„Kompleksowość jest głównym problemem postępu” Ken Crater [Ctrl. Eng. 99] „...w sytuacji, gdy cykle rynkowe ulegają ciągłemu skracaniu, technologia jest bezużyteczna, jeśli użytkownicy nie mogą jej błyskawicznie wdrożyć”

Pojęcie „złożoność” (ang. *complexity*), pojawia się z coraz to większą częstotliwością dla wskazania przyczyn problemów rozwoju. Co oznacza kompleksowość? J.Horgan w *Scientific American* 1995 [Mikuleck web] wymienia aż 31 definicji złożoności. Mikuleck podkreśla, że najczęściej złożoność definiowana jest przez samą siebie. Proponuje własną definicję (tłumaczenie własne):

„Złożoność jest cechą systemu świata rzeczywistego, która przejawia się w niemożności podania formalizmu, który potrafiłby uchwycić wszystkie jego cechy. Wymaga to znajdowania wyraźnie różnych sposobów interakcji z systemem. Wyraźnie różnych w takim sensie, że podczas

tworzenia poprawnego modelu, systemy formalne potrzebne do opisu każdego z różnych aspektów nie są pochodnymi innych"

Złożoność jest zatem funkcją jakościową systemu, a nie skali, jak wskazywał Neuman. Charakterystykę porównawczą systemów złożonych i skomplikowanych (trudnych), opracowaną na podstawie Mikuleck, przedstawiono w tabeli 3.1.

Tab. 3.1 Porównanie cech systemów złożonych i skomplikowanych [jr na podstawie Mikulecky]

System złożony	System skomplikowany
Brak jednego dużego modelu	Jeden duży model
Całość to więcej niż suma części	Całość jest sumą składników
Relacje skutkowo-przyczynowe bardzo rozległe i posplatane	Relacje skutkowo-przyczynowe wyraźne
Generyczny	Nie-generyczny
Analityczny < syntetyczny	Analityczny = syntetyczny
Niepodzielny	Podzielny
Nieprzeliczalny	Obliczeniowy
Świat rzeczywisty	System formalny

3.1.2 Cykl życia zautomatyzowanego systemu wytwórczego

Cykl życia systemów wytwórczych obejmuje okres od powstania zapotrzebowania na dany system aż do wycofania tego systemu z eksploatacji. W makro skali wyróżniane są dwa zasadnicze etapy cyklu życia systemów wytwórczych, tj.: inwestycji oraz późniejszej eksploatacji (rys.3.3). Faza inwestycji jest nabywaniem dóbr inwestycyjnych. Eksploatacja wymaga utrzymywania maszyn i urządzeń w stanie umożliwiającym realizację procesu produkcyjnego.



Rys. 3.3 Uproszczony cykl życia zautomatyzowanego systemu wytwórczego [jr]

Czas cyklu życia produktów ulega ciągłemu skracaniu. Dotyczy to zarówno etapów ich projektowania i realizacji (inżynieria współbieżna), jak również ich użytkowania. Tendencja ta jest szczególnie widoczna dla produktów rynku masowego z dużym udziałem elektroniki i oprogramowania, o których mówi się, że „starzeją się bardzo szybko”. Aktualnie czas życia komputerów PC szacowany jest na 2-3 lat, a serwerów 3-5 lat. Zgodnie z prawem Moore’a można estymować, że moc obliczeniowa „nowego” komputera po modernizacji (ang. *upgrade*), będzie czterokrotnie większa, a serwera aż dziesięciokrotnie większa w stosunku do poprzednika. Tendencja ta dotyczy nie tylko komputerów i oprogramowania, ale większości wyrobów branży elektro-maszynowej, tj. przemysłu maszynowego, motoryzacyjnego, sprzętu gospodarstwa domowego, etc.

Możliwość jak najdłuższego eksploataowania systemów wytwórczych identyfikowana jest jednocześnie jako warunek konkurencyjności kosztowej. Stąd poszukiwanie rozwiązań adaptacyjnych i elastycznych, za pomocą których będzie można odpowiadać na zmieniające się wymagania rynku.

3.1.3 Automatyzacja wytwarzania

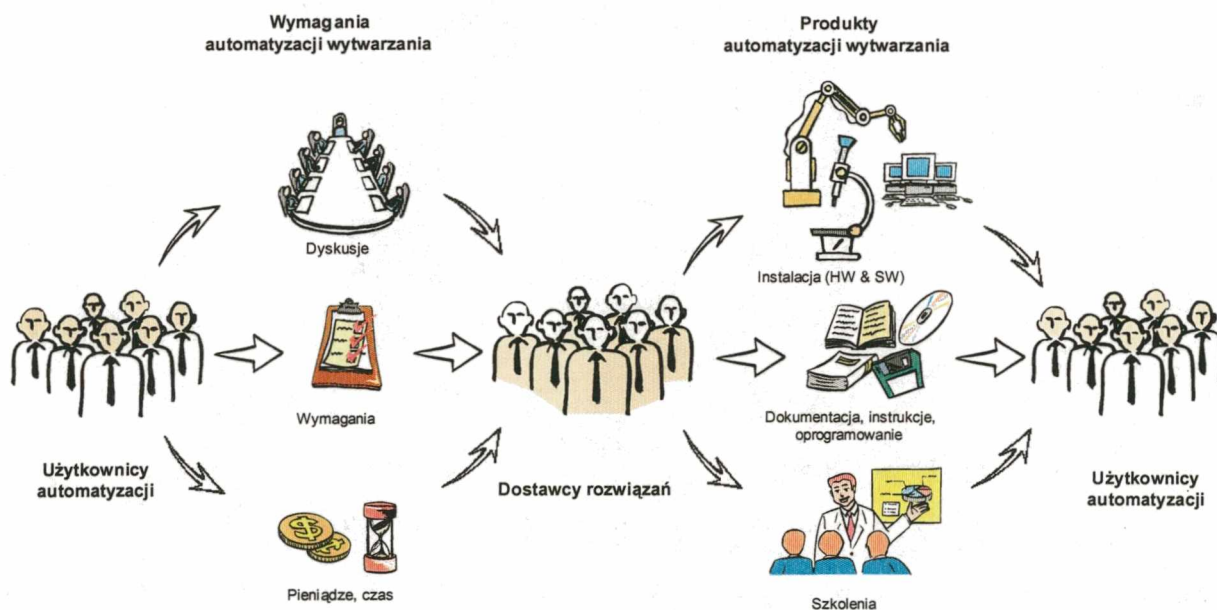
Czym jest automatyzacja wytwarzania? Czy obejmuje ona pierwszy etap cyklu życia zautomatyzowanego systemu wytwórczego przedstawiony na rysunku 3.3?

automatyzacja, wprowadzanie w różnych dziedzinach życia urządzeń automatycznych, (automatów) w celu samoczynnego sterowania obiektami technicznymi; ogranicza bezpośredni udział człowieka w pracy fiz. i umysłowej (mechanizacja zastępuje tylko pracę fiz.). [PWN]

Zgodnie z polską definicją, automatyzacja jest procesem, którego celem jest doskonalenie różnych dziedzin życia poprzez wprowadzanie rozwiązań technicznych. Automatyzacja, w przeciwieństwie do mechanizacji, ułatwia lub zastępuje również czynności umysłowe człowieka. Angielskie słowo *automation* jest szersze pojęciowo. Obejmuje ono automatyzację i automatykę, czyli naukę o sterowaniu różnymi procesami bez udziału, lub przy ograniczonym udziale człowieka. *Automation* występuje również w znaczeniu rzeczownikowym, tj. dla określenia samodzielnego działania lub sterowania urządzeniami, procesami lub systemami, jak również jako technika lub wyposażenie używane do osiągnięcia automatycznego działania [MBS 98].

Głównym wynikiem automatyzacji jest opracowanie lub modyfikacja rozwiązań technicznych. Współczesna automatyzacja, jak to zostało scharakteryzowane w Rozdziale 1, oparta jest głównie na zastosowaniu elektroniki i oprogramowania, których udział stale rośnie. Obsługa urządzeń staje się wprawdzie coraz łatwiejsza, jednak zakres oferowanej funkcjonalności i konfigurowalności rozszerza się. W takim przypadku zapotrzebowanie na wiedzę eksploatacyjną i serwisową rośnie. Podstawą przekazywania takiej wiedzy staje się dokumentacja oraz szkolenia, które stanowią integralną część wyników automatyzacji.

Scenariusz realizacji projektu automatyzacji został symbolicznie przedstawiony na rysunku 3.4.



Rys. 3.4 Scenariusz realizacji projektu automatyzacji [jr]

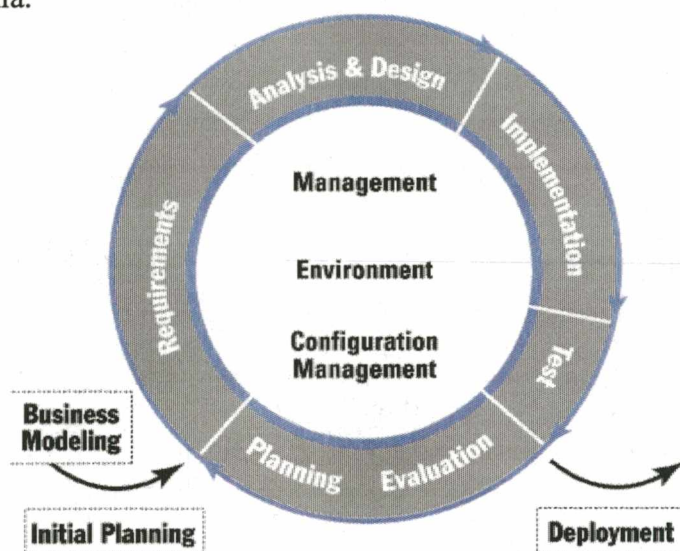
Scenariusz automatyzacji z rysunku 3.4 wskazuje na potrzebę współpracy użytkowników i dostawców. Zwraca on również uwagę na różnorodność produktów automatyzacji, wyróżniając instalację (sprzęt i oprogramowanie), dokumentację oraz szkolenia.

Zadanie automatyzacji - wprowadzanie usprawnień technicznych, może być rozważane w kontekście „problem – rozwiązanie”. Dziedzinę problemu stanowi proces podlegający

automatyzacji, a dziedzinę rozwiązania obejmują techniki i urządzenia sterujące. Reprezentantami powyższych dziedzin są role klienta i dostawcy. Rola lub aktor są pojęciami symbolicznymi opisującymi kompetencje. Rolę klienta charakteryzuje znajomość problemu, natomiast rolę dostawcy charakteryzuje znajomość rozwiązania. Dla małych projektów obie te role mogą być odgrywane przez tą samą osobę lub zespół, a w przypadku dużych projektów są to wyspecjalizowane grupy reprezentowane przez użytkowników i automatyków.

W tym przypadku, gdy role klienta i dostawcy są pełnione przez zespoły z różnych firm, to współpraca między nimi regulowana jest na podstawie umów dwustronnych określających prawa i obowiązki każdej ze stron. Typowo, strona dostawcy zobowiązuje się do wykonania danego dzieła, zaś strona zamawiająca do zapłaty wynagrodzenia z chwilą jego wykonania. Istotną sprawą jest zagadnienie odpowiedzialności za wynik projektu automatyzacji. W przypadku umowy o dzieło, odpowiedzialność leży po stronie wykonawcy, w przypadku umowy zlecenia, nazywanej „umową starannego działania”, wykonawca nie bierze odpowiedzialności za jej wynik, ale wyłącznie za przebieg realizacji.

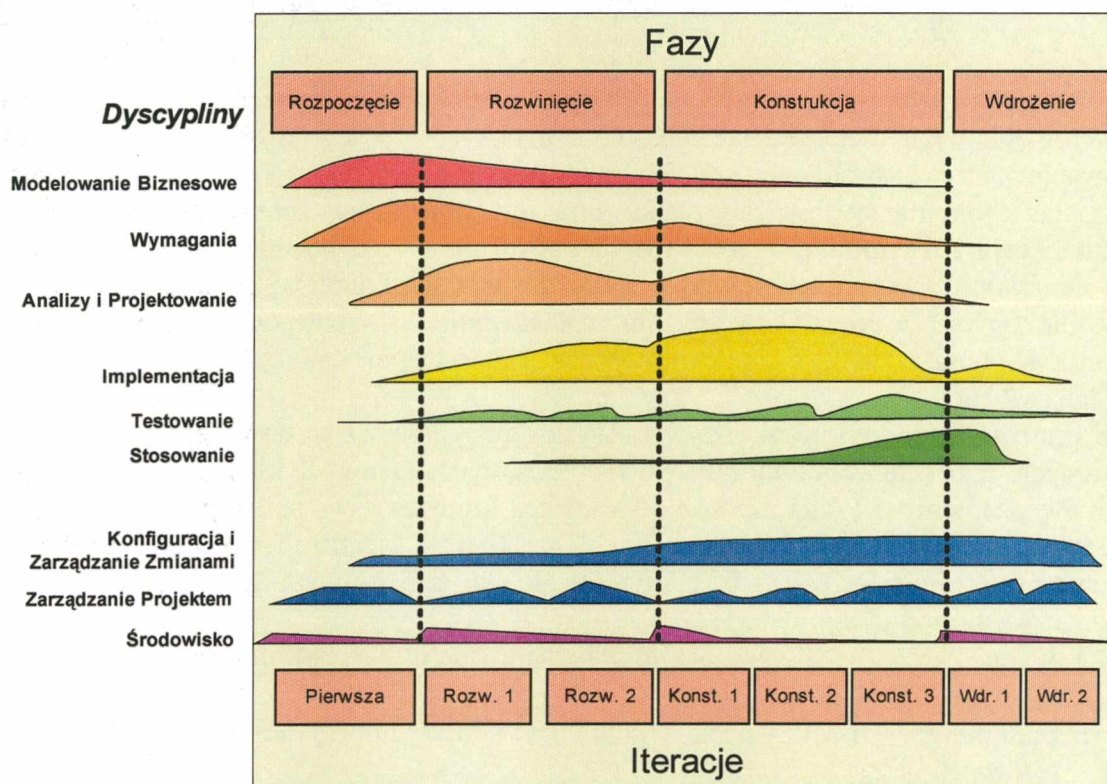
Jeśli zadanie automatyzacji charakteryzują unikatowe wymagania i ograniczone zasoby, to w myśl metodyki PMBOK (ang. *Project Management Body of Knowledge*) [PMBOK 00] jest ona projektem. Cykl życia projektu automatyzacji, tj. podział na poszczególne fazy, nie jest jednoznacznie zdefiniowany. Zależnie od celu analizy, wyróżniana jest różna liczba faz oraz stosowane różne nazewnictwo. Bazowy przebieg projektu wyznacza „model wodospadowy”. Opiera się on na dobrze zdefiniowanym celu i etapach jego osiągnięcia, dzięki czemu jego przebieg jest liniowy i sekwencyjny. Definicja projektu podkreśla jednak jego unikatowość. Oznacza to, że sprecyzowanie celu lub poszukanie drogi jego osiągnięcia jest przedmiotem projektu. W takiej sytuacji, prowadzenie projektu zgodnie z modelem wodospadowym jest trudne do utrzymania.



Rys. 3.5 Cykliczny przebieg projektów [Rational 01]

Modele wspierające uczenie, tj. precyzowanie celu i poszukiwanie dróg osiągnięcia celu, bazują na przebiegu cyklicznym spiralnym (patrz rys. 3.5). Przykładowo, model poprawy Deminga wyróżnia cztery etapy, nazywając je odpowiednio: planowanie (ang. *plan*), realizacja (ang. *action*), weryfikacja (ang. *check*), wprowadzenie (ang. *action*). Metodyka ROPES (ang. *Rapid Object-Oriented Process*) [Douglas 99] rozwoju oprogramowania obiektowego dla zastosowań typu wbudowanego (ang. *embedded*), definiuje poszczególne fazy takie jak: analiza, projektowanie, kodowanie i testowanie. Natomiast Rockwell [Rockwell 97] wyróżnia aż sześć etapów w inwestycyjnym modelu cyklu życia

zautomatyzowanego systemu (ang. *The Automation Investment Life Cycle Model*”, nazywając je kolejno: uzasadnienie (ang. *justify*), wprowadzenie (ang. *apply*), instalacja (ang. *install*), użytkowanie (ang. *operate*), utrzymanie ruchu (ang. *maintain*) i doskonalenie (ang. *improve*). Główną wadą powyższych modeli jest brak silnych mechanizmów kontroli osiągania sukcesu, co utrudnia zarządzanie takimi projektami. Oznacza to, że cykliczny przebieg projektu może zaniedbać główne wymaganie biznesowe – tj. obiektywne wskaźniki sukcesu: finanse, czas i funkcjonalność. Niebezpieczeństwo takich niekończących się projektów nazywane jest „spiralą śmierci”.

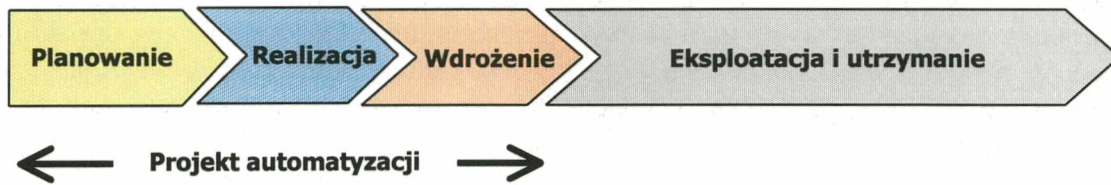


Rys. 3.6 Model Zunifikowanego Procesu Rational (RUP) [Rational 01]

Metodyka RUP (ang. *Rational Unified Process*) [Rational 01] przedstawia proces opracowywania oprogramowania w postaci modelu dwuwymiarowego, identyfikując wymiar zadaniowy, w postaci procesów, oraz wymiar czasowy w postaci faz realizacji (patrz rys. 3.6). RUP wskazuje, że główne procesy projektowe, takie jak modelowanie biznesowe, wymagania, analiza i projektowanie, implementacja, testowanie, instalowanie oraz procesy wspierające, takie jak zarządzanie zmianami i konfiguracją, zarządzanie projektem oraz przygotowanie i pielęgnacja środowiska projektowego przebiegają współbieżnie. Są one jednak wzajemnie synchronizowane w ramach faz kontrolowanych przez przepływ aktywności i dokumentów (ang. *workflow*). Poszczególne fazy, tj. rozpoczynanie, rozwijanie, konstruowanie i przenoszenie, dopuszczają pętle iteracyjne. Model RUP jest zatem integracją zalet modelu „inkrementacyjno – iteracyjnego” i zalet modelu „wodospadowego”. Pierwszy z nich wspiera kreatywność, uczenie, wcześniejsze weryfikowanie i motywowanie poprzez sukces. Natomiast drugi z tych modeli wspiera biznesowe wymagania projektu, tj. łatwość zarządzania projektem, jego zakresem prac, czasem i kosztami.

Należy zwrócić uwagę również na „rozmycie” poszczególnych działań, których granice i poziomy aktywności nie są ilościowo sprecyzowane. Oznacza to, że akceptowana jest miara jakościowa dla przebiegu projektu, który opisany jest w sposób szkieletowy.

Zgodnie ze scenariuszem automatyzacji rys. 3.4, dla perspektywy klienta i dostawcy, w modelu cyklu życia zautomatyzowanego systemu wytwórczego, wyróżniono planowanie, realizację i wdrożenie – patrz rys.3.7.



Rys. 3.7 Cykl życia zautomatyzowanego systemu wytwórczego [jr]

Etap planowania inicjowany jest „potrzebą”. Potrzeba może wynikać ze zidentyfikowanego problemu lub wizji ulepszenia czegoś, co nie jest w istocie klasyfikowane jako problem. Innowacje oraz wprowadzanie nowych rozwiązań nie zawsze są łatwe do uzasadnienia w kryteriach ekonomicznych. Etap planowania jest intensywną interakcją między klientem poszukującym rozwiązania, a dostawcą mogącym takie rozwiązanie wytworzyć. Celem tej fazy jest sformułowanie spójnej wizji przyszłego systemu. Ponieważ stopień osiągnięcia wizji jest obiektywnie nieweryfikowalny, to ustanawiane są biznesowe kryteria osiągnięcia sukcesu. Kryteria takie są formułowane przez dostawcę, jako specyfikacja zobowiązań oraz plan ich realizacji.

Etap realizacji jest zasadniczą aktywnością dostawcy i prowadzi do urzeczywistnienia wizji. W ramach tego etapu mogą następować uszczegółowienia z klientem, lub prezentacje wyników pośrednich. W ten sposób wykonawca, poprzez tzw. pętle sprzężenia zwrotnego, stara się minimalizować ryzyko niezadowolenia klienta. Mechanizm taki wprowadza jednak utrudnienia w procesie realizacji. Główne z nich to zmiana wymagań i oczekiwań klienta, który z czasem modyfikuje własne wizje i oczekiwania.

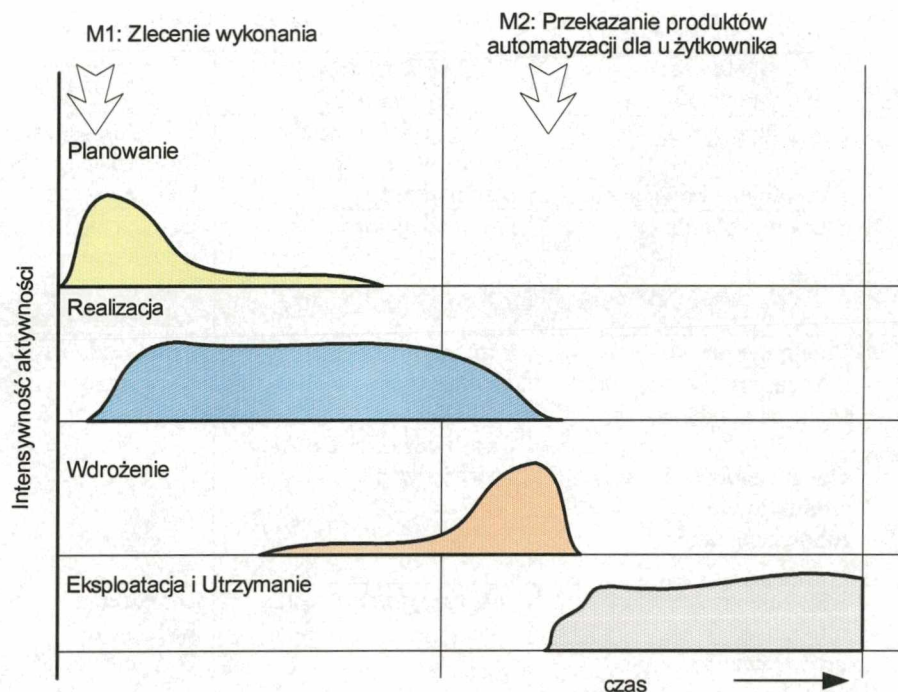
Produkty końcowe fazy realizacji, jak to przedstawiono na rys. 3.4, można sklasyfikować jako trzy grupy:

- zautomatyzowane rozwiązanie techniczne - sprzęt i oprogramowanie (ang. *hardware i software*),
- dokumentacja - użytkownika, techniczna i kody źródłowe oprogramowania,
- szkolenia (usługa) – dla użytkowników i serwisantów.

Podczas etapu wdrożenia następuje uruchomienie opracowanych rozwiązań technicznych oraz ich integracja z otoczeniem. Szczególną formą integracji nowych rozwiązań z otoczeniem jest szkolenie pracowników. Typowo obejmuje ono dwa poziomy, tj. szkolenie podstawowe z zakresu obsługi i interpretacji błędów oraz szkolenie zaawansowane dla konfiguracji i serwisowania urządzeń.

Na podstawie modelu RUP opracowano jakościowy diagram aktywności w cyklu życia automatyzacji, co przedstawiono na rysunku 3.8. W modelu tym wyszczególniono etapy planowania, realizacji, wdrożenia, eksploatacji i utrzymania, których natężenie aktywności jest rozmyte. Jako kamienie milowe wspierające zarządzanie projektem automatyzacji wyznaczono:

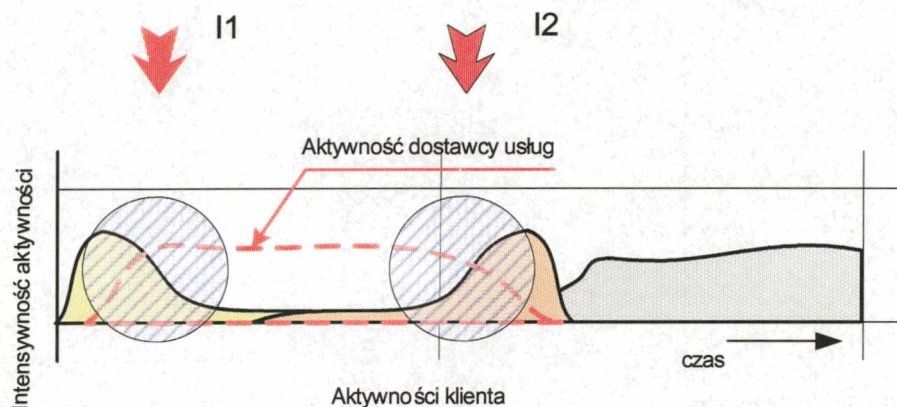
- M1: Zlecenie wykonania,
- M2: Przekazanie produktów automatyzacji do użytkowania.



Rys. 3.8 Aktywności w poszczególnych etapach automatyzacji wytwarzania [jr]

Osiągnięcie kamienia milowego M1 jest potwierdzane sformułowaniem specyfikacji zobowiązań w formie pisemnej. Osiągnięcie statusu projektu M2 wymaga natomiast wytworzenia produktów automatyzacji.

Na podstawie diagramu aktywności przedstawionych na rys. 3.8, dokonano rzutowania aktywności ról klienta i dostawcy usług, co przedstawia rys. 3.9. Uzyskany wynik wskazuje, że podczas fazy planowania (obszar I1) następuje przekazanie aktywności ze strony klienta do dostawcy usług. Realizacja jest prowadzona głównie przez rolę dostawcy, który w fazie wdrażania przekazuje swoją aktywność roli klienta (obszar I2).



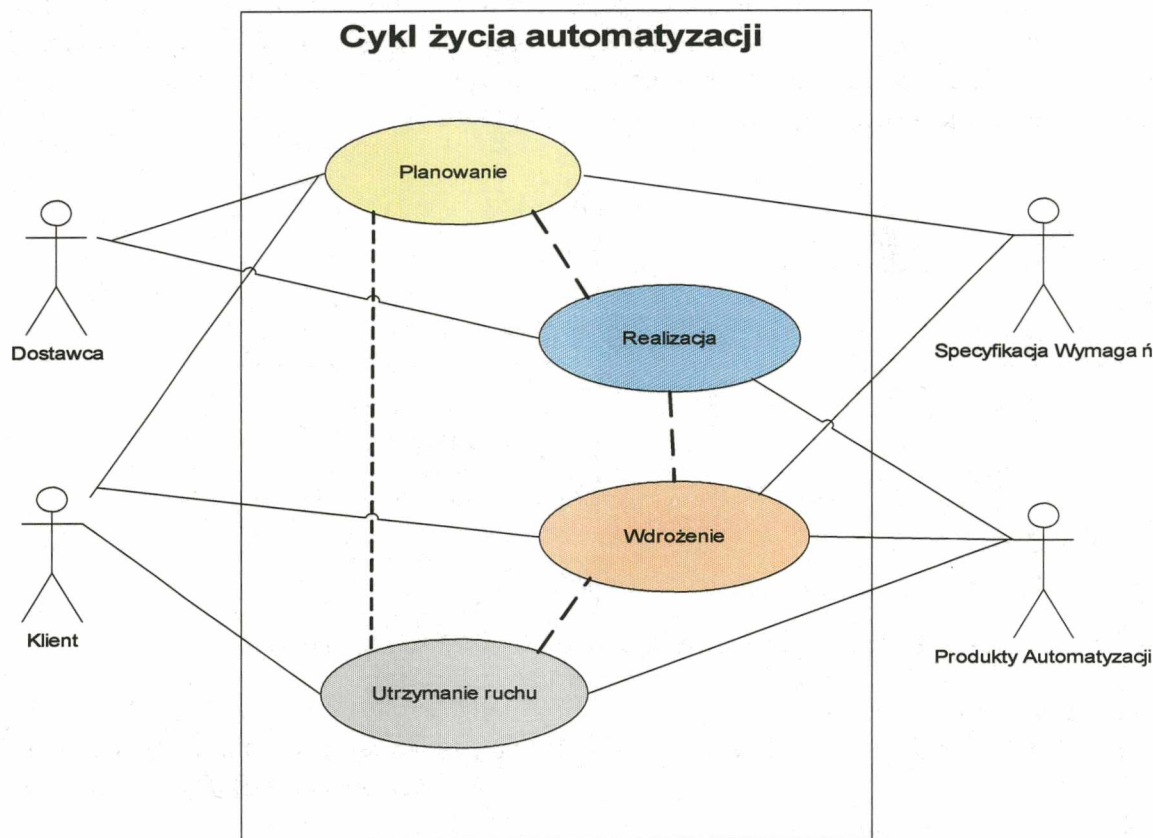
Rys. 3.9 Obszary przekazywania aktywności klient-dostawca usług [jr]

Jak wskazano powyżej, człowiek występuje w roli klienta (użytkownika) lub dostawcy usług. Każdą z tych ról charakteryzują określone kompetencje i zadania do wykonania. Specyfikację roli klienta i dostawcy oraz produkty wyjściowe w kontekście każdego z procesów przedstawiono w tabeli 3.2.

Tab.3.2 Wyniki analizy interfejsów między etapami cyklu życia automatyzacji

Faza automatyzacji	Rola klienta (użytkownika)	Rola dostawcy (dostawcy)	Produkty wyjściowe
Planowanie	Formułuje wymagania Akceptuje zobowiązania	Formułuje zobowiązania Opracowuje plan realizacji	Wizja rozwiązania Specyfikacja zobowiązań Plan realizacji
Realizacja	Zleca wykonanie	Opracowuje produkty automatyzacji	Urządzenia i systemy wytwórcze Dokumentacja Szkolenia i doradztwo
Wdrażanie	Przejmuję produkty automatyzacji (uczy się obsługi i działania) Ocenia zgodność produktów ze zobowiązaniami	Przedstawia produkty automatyzacji Szkoli w zakresie obsługi i serwisowania	Urządzenia i systemy wytwórcze Dokumentacja
Eksploatacja i utrzymanie ruchu	Użytkuje produkty automatyzacji, usuwa awarie, rekonfiguruje, inicjuje cykl modernizacji	(nie uczestniczy)	Urządzenia i systemy wytwórcze Dokumentacja

Dla całościowego przedstawienia powiązań między poszczególnymi etapami automatyzacji, jej uczestnikami (aktorami) i produktami, opracowano diagram przypadków użycia – rysunek 3.10. Diagram ten oparto na syntaktyce i semantyce języka UML, która zostanie omówiona w Rozdziale 4.



Rys. 3.10 Diagram przypadków użycia - Cykl Życia Automatyzacji [jr]

3.1.4 Eksploatacja i utrzymanie

W literaturze anglojęzycznej znajdujemy wspólne określenie „*maintenance*” dla działań prowadzonych w zakresie mechanicznym, elektrycznym i informatycznym w celu zapewnienia ciągłości działania urządzenia, procesu lub systemu. Poszukiwanie spójnej definicji „*maintenance*” wskazuje na przenikanie pojęć: instalacja, naprawa, wsparcie serwisowe, modyfikacje, rozszerzanie, a nawet rozwój.

W języku polskim, w kontekście urządzeń mechanicznych mówi się o „utrzymaniu ruchu”, którego zadaniem są naprawy i konserwacje niezbędne do zapewnienia niezawodnej i wydajnej eksploatacji. Obok zachowawczego utrzymania ruchu coraz większego znaczenia nadaje się zapobieganiu (prewencji), bazującej na estymowaniu krytycznych wskaźników zużycia. Prewencyjne utrzymanie ruchu, wykorzystujące strategie planistyczne lub zmiany stanu, identyfikowane jest jako najważniejsze źródło poprawy dostępności systemów produkcyjnych. Wymagania adaptacji lub nowej funkcjonalności urządzeń mechanicznych leżą zwykle w zakresie zagadnień związanych z działalnością planistyczną i są realizowane jako samodzielne projekty modernizacji.

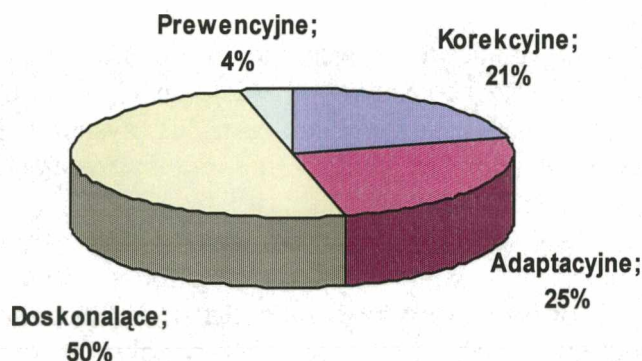
Pielęgnowanie, konserwacja (angielskie *maintenance*), czynności związane z wprowadzaniem poprawek do oprogramowania, dostosowywaniem go do nowych wymagań użytkowników, uaktualnianiem, tj. wymianą starszych wersji oprogramowania na nowsze. Pielęgnowanie dużych systemów wymaga też dbałości o utrzymywane banki danych, ochrony na wypadek awarii, itp. [Płoski 99]

Jako odpowiednik „*maintenance*” w kontekście oprogramowania, w literaturze polskojęzycznej przyjęło się pojęcie „pielęgnacji oprogramowania”. Rozkłada ono jednak inaczej, niż w utrzymaniu ruchu, priorytety zakresu zadań. Pielęgnacja oprogramowania wyróżnia cztery główne zakresy działań:

- korekcyjne (ang. *corrective maintenance*) – gdy usuwane są błędy,
- adaptacyjne (ang. *adaptive maintenance*) – gdy wprowadzane są zmiany niezbędne do dalszego użytkowania systemu,
- doskonalące (ang. *perfectioning maintenance*) – gdy użytkownik wymaga nowej funkcjonalności lub proces większej wydajności,
- prewencyjne (ang. *preventive maintenance*) – w celu poprawy pielęgnacji lub minimalizacji ryzyka wystąpienia awarii czy przeciążenia systemu.

Rozkład udziału poszczególnych klas pielęgnacji przedstawiono na rys. 3.11. Ocenia się, że udział kosztowy zadań adaptacyjnych i doskonalących, a więc mieszczących się w definicji automatyzacji, w utrzymaniu oprogramowania sięga aż 75%.

Utrzymanie systemu (z angielskiego *housekeeping, system maintenance*), ogół czynności wykonywanych w celu niezawodnej i wydajnej eksploatacji systemu informatycznego, obejmujący takie prace jak składowanie, defragmentacja dysków, usuwanie zbędnych danych, uaktualnianie dokumentacji, unowocześnianie sprzętowych i programowych składowych systemu, dbałość o bezpieczeństwo, doglądanie systemu, przełamywanie blokad, usuwanie skutków awarii, itp. Utrzymanie dużego systemu należy do obowiązków osoby nazywanej administratorem systemu. [Płoski 1999]



Rys. 3.11 Typy pielegnacji oprogramowania [Vliet 1999]

W kontekście systemów informatycznych, tj. oprogramowania i platform sprzętowych, w języku polskim funkcjonuje pojęcie „utrzymania systemu”. Zatem „utrzymanie ruchu” zautomatyzowanego systemu produkcyjnego będzie obejmowało działania prowadzone w zakresie mechanicznym, elektronicznym i informatycznym.

Szacuje się, że w przypadku oprogramowania, koszty poniesione w czasie eksploatacji mogą sięgać aż 400% kosztów inwestycyjnych [DES 00].

Awarie

Podczas normalnej obsługi urządzeń powstają sytuacje awaryjne. Konieczność błyskawicznego usunięcia przestojów pojmowana jest nadrzędnie. Naprawy doraźne, które nie zostały poprzedzone wnikliwą analizą skutków i przyczyn błędów, prowadzą zwykle do ich ponownego wystąpienia a nawet eskalacji. Jeśli automatyzacja to „zastępowanie manualnych lub kognitywnych funkcji człowieka przez automat”, to „zaufanie” w tej współpracy odgrywa istotną rolę. System automatyczny o niskim poziomie ufności staje się uciążliwą przeszkodą w realizacji zadań produkcyjnych człowieka. Człowiek błędnie informowany ignoruje źródło takich informacji, a więc przestaje na nie reagować. Skutkiem tego pojawia się „dziura w bezpieczeństwie”, np. jakościowym zautomatyzowanego systemu wytwórczego. Gruntowne badanie skutków i przyczyn błędów jest więc bezwzględny warunkiem utrzymywania ruchu, szczególnie w przypadku kompleksowych systemów automatyzacji wytwarzania. Spełnienie takiego warunku wymaga jednak dostatecznej wiedzy o zastosowanych rozwiązaniach wewnętrznych. Zarządzanie jakością zaleca wówczas stosowanie metod i narzędzi wspierających analizę skutkowo-przyczynową oraz podejmowanie decyzji, np. FMEA³ procesu [McDermott, et al 96], [Humpries et al. 98].

Podczas utrzymywania ruchu systemów produkcyjnych zachodzi także konieczność ich przekonfigurowywania. Dlatego też zautomatyzowane urządzenia wytwórcze wyposażane są w coraz obszerniejsze możliwości konfiguracyjne, dzięki wystarczającym zasobom sterowników PLC/PC i niskim kosztom powielania oprogramowania. Funkcjonalność taka jest odpowiedzią na wyzwanie elastyczności „konfigurować zamiast programować”. Tak rozumiana elastyczność umożliwia łatwą integrację automatu z linią technologiczną, użytkowanie go mimo wystąpienia awarii, czy zmianę programu produkcyjnego. Zapewnienie bezpieczeństwa i niezawodności leży jednak w opozycji do wyżej określonej elastyczności. Przy tak wielu kombinacjach konfiguracyjnych, sam wytwórca (dostawca) nie

³ Analiza Skutków i Przyczyn Błędów (ang. *Failure Mode Effect Analysis*)

jest w stanie ich w pełni przetestować. Dlatego też odpowiedzialność za działanie automatu, czy linii produkcyjnej „spada” więc na użytkownika.

Adaptacja

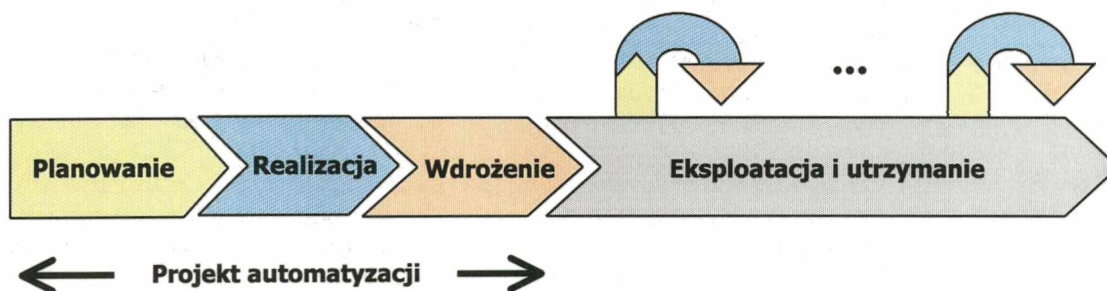
Z uwagi na bezpieczeństwo instalacji, bezpieczeństwo danych, a szczególnie bezpieczeństwo ludzi i środowiska, zwykle wprowadza się ochronę dostępu do krytycznych funkcji systemu. Dzięki poziomom uprawnień i ochronie hasłami, nieautoryzowana konfiguracja lub zmiana parametrów działania stacji automatycznej jest niemożliwa. Tego rodzaju ograniczenia w dostępie do funkcji konfiguracyjnych „usztyniają” jednak wykorzystanie systemu.

Zmieniające się konstrukcje wytwarzanych produktów, jak również nowe metody ich produkcji, czy sposoby poprawy jakości, wymagają wprowadzania zmian w systemie. Gdy zmiany te przekraczają elastyczność oferowaną przez użytkowane produkty automatyzacji, to konieczna staje się modernizacja. O modernizacji można mówić również w stosunku do specyfikacji wymagań i specyfikacji zobowiązań. W takim przypadku zarządzanie zmianami (ang. *change management*) następuje równolegle z całym cyklem życia zautomatyzowanego systemu wytwórczego, już od pierwszej fazy jego planowania. Modernizacja jest zatem „uruchomieniem” cyklu zmian, który obejmuje: planowanie, realizację i wdrożenie. Bazuje ona na stanie wiedzy o aktualnej realizacji. Gdy stan ten nie jest znany, lub utracono spójność między rzeczywistymi obiektami a dokumentacją techniczną, to efektywność i bezpieczeństwo automatyzacji jest zagrożone.

Doskonalenie

Systemy mechaniczne wymagają głównie konserwacji i wymiany zużywających się elementów, zaś każda modernizacja wykracza poza zakres rozumienia pojęcia utrzymanie ruchu.

W kontekście systemów informatycznych doskonalenie należy do zadań utrzymania ruchu. Migracja, czyli przejście na nowszą wersję oprogramowania, obejmuje zaplanowanie przebiegu zmiany, dostosowanie sprzętu, instalację oprogramowania i szkolenia użytkowników. W pkt. 3.1.2 wskazano, że sprzęt komputerowy charakteryzuje się krótkim cyklem życia (3-5 lat). Dlatego też w razie awarii podzespołu komputerowego jego wymiana często związana jest z zastosowaniem nowocześniejszej wersji. W takim przypadku, współpraca nowoczesnego modułu z systemem operacyjnym lub aplikacją nie zawsze jest gwarantowana. Potrzeba doskonalenia i uaktualniania oprogramowania aplikacyjnego, a szczególnie systemów operacyjnych wynika z konieczności zapewnienia bezpieczeństwa. „Dziury” w bezpieczeństwie systemów operacyjnych „łatane są” na bieżąco poprzez pakiety serwisowe (ang. *service pack*). Poprawki te, będące formą wsparcia technicznego, oferowane są wyłącznie dla tzw. aktualnych wersji oprogramowania, tj. podczas cyklu życia produktu, określonego przez producenta.



Rys. 3.12 Cykl życia zautomatyzowanego systemu wytwórczego wraz z doskonaleniem [jr]

Jak wskazano w Rozdziale 1, zautomatyzowane systemy wytwórcze charakteryzuje coraz większy udział oprogramowania. Wynika to z tego, że **wraz ze wzrostem udziału oprogramowania, konieczna jest również zmiana podejścia do procesów utrzymania zautomatyzowanych systemów wytwórczych. Zmiana ta wymaga akceptacji poglądu, że doskonalenie, tj. poprawa, staje się zadaniem pierwszoplanowym utrzymania ruchu instalacji** [Reiner, et al. 03].

Zapobieganie

Zapobieganie, prewencja lub predykcja należą do najważniejszych zadań utrzymania ruchu. W przypadku systemów mechanicznych zapobieganie obejmuje np. optymalizację smarowania, czy też wymianę elementów podlegających zużyciu. Monitorowanie zużycia oraz modele matematyczne, opisujące zachodzące procesy podczas eksploatacji systemu wytwórczego, są podstawą do zaplanowania czynności serwisowych. Dostępnych jest wiele modułów programowych wspierających proces nadzorowania i planowania remontów maszyn. W kontekście oprogramowania problem taki nie zachodzi. Inżynieria oprogramowania wykorzystuje pielęgnację prewencyjną, głównie do minimalizacji czasu potrzebnego do lokalizacji i usunięcia potencjalnych usterek. Pielęgnacja prewencyjna w kontekście oprogramowania obejmuje zatem porządkowanie kodów źródłowych i dokumentacji z nimi związanych. Działaniem prewencyjnym w zakresie sprzętu komputerowego jest natomiast dbanie o integralność danych, oraz analiza zmian wydajności systemu, wynikająca ze zwiększania objętości danych w systemie informatycznym.

3.2 Efektywność automatyzacji

Przy próbie formalnego zdefiniowania „efektywności” napotykamy na problemy. Słownik języka polskiego wskazuje na efektywność ekonomiczną, tj. porównanie wartości uzyskanych efektów do nakładu czynników użytych do ich uzyskania. Wskazuje on także, że w prakseologii efektywność utożsamiana jest ze skutecznością. Nieprzydatny jest również słownik polsko-angielski [Collins], który tłumaczy „efektywność” jako „*efficiency*”, zaś w drugą stronę wskazuje na „sprawność” lub „wydajność”. Angielskie znaczenia „*efficiency*” definiuje je jako „jakość lub miara sprawności lub wydajności” (ang. „*efficient*”). Sprawnie bądź wydajnie, to działać lub produkować skutecznie z jak najmniejszą stratą wysiłku i zasobów.

W przypadku efektywności organizacji, Elmuti [Elmuti 97] identyfikuje sześć cech: samodzielność w zarządzaniu (ang. *self-managed*), wyniki (ang. *performance*), satysfakcja (ang. *satisfaction*), produktywność (ang. *productivity*), jakość (ang. *quality*) i efektywność (ang. *effectivness*) – patrz tabela 3.3.

Tabela 3.3. Cechy efektywności organizacyjnej [Elmuti 97]

Cecha	Definicja
Samodzielność w zarządzaniu	Grupy posiadające samodzielność w zarządzaniu to zespoły (5-12 osobowe), z umiejętnościami technicznymi jak również uprawnieniami w zakresie kierowania i zarządzania własnym zespołem (Bowen, Lawler 1994), (Pasmore 1994), (Stokes, Stewart 1991).
Wyniki	Opłacalność odnosząca się do całkowitej działalności. Opłacalność może być wyrażana w kategoriach dochodu netto, dochodów na udział, rentowności. Inne ogólne cele to wzrost, wielkości produkcji, udziałach w rynku, rozwój innowacji pracowników, zarządzanie zasobami (Daft 1995).
Satysfakcja	Zadowolenie pracowników ze współpracowników, pracy, przełożonych, organizacji w odniesieniu do innych, zarobków, dokonanego rozwoju organizacji i możliwości dalszego rozwoju (Likert 1973).

Produktywność	Postrzeganie wykonanej pracy przez pracowników w dziale lub wydziale, w kontekście ilości i jakości oraz efektywności jej realizacji (Mott 1972).
Jakość	Postrzeganie wykonanej pracy w organizacji przez pracowników, w kontekście wymagań w celu zaspokojenia potrzeb i satysfakcji użytkowników. Jakość jako zgodność z wymaganiami (Crosby 1979).
Efektywność	Organizacje efektywne, to takie, które produkują więcej i wyższej jakości produktów i dostosowują się efektywniej do zmian otoczenia, w tym samym czasie utrzymując wysoki poziom satysfakcji swoich członków (Likert 1973).

Jak wynika z zestawienia przedstawionego w tabeli 3.3, jedynie „wyniki” stanowią miarę liczbową, która poprzez porównanie z wzorcem może być obiektywnym wskaźnikiem sukcesu. Wszystkie pozostałe cechy są jedynie opisowym scharakteryzowaniem złożoności problemu i nie podają żadnych miar ilościowych.

Przed nastaniem recesji w USA, wiele firm postrzegało technologię informatyczną (IT) jako strategiczny priorytet. Dziś widać już wyraźnie, że motorem napędowym rozwoju technologii informatycznej jest strategia biznesowa, a nie odwrotnie [CXO 03]. Aż 82% badanych firm przyznało, że strategia biznesowa ma znaczący lub zasadniczy wpływ na strategię w obrębie IT. Strategia biznesowa wymaga jednak konkretnych wyników ilościowych; w jaki sposób ocenić koszt wprowadzenia i użytkowania systemów informatycznych oraz rentowność tej inwestycji?

Dla wskazania całkowitych kosztów rzeczywistych i ukrytych operuje się pojęciem kosztu posiadania TCO (ang. *Total Cost of Ownership*) [TCO web].

Przykładowo, badania porównawcze TCO dla systemów operacyjnych Windows i Linux, przeprowadzone przez dwie niezależne firmy analityczne przynoszą zupełnie rozbieżne wyniki. Firma RFG dowodzi, że w perspektywie 3 letniej TCO systemu Linux jest o 50% niższe niż TCO dla systemu Windows (patrz tab. 3.4). Firma IDC rozszerza wprowadzone okres użytkowania do 5 lat, jednak TCO systemu Linux wyznacza o średnio 15% wyższe jak TCO systemu Windows (patrz tab. 3.5). Skąd wynikają tak duże rozbieżności? Szczegółowa analiza metodyki przeprowadzonych badań oraz przyjętych składników kosztów wskazuje, że nie wynikają one z różnej długości okresu eksploatacji, lecz istotnego udziału czynników tzw. „miękkich”. Komentatorzy tych wyników pytają, skąd podstawy, aby przyjąć w badaniach IDC, że koszt personelu, szacowany na 60% TCO, jest dla systemu Linux o 30% wyższy?

Tabela 3.4 Porównanie 3 letnich TCO dla, Linux, Solaris, Windows [RFG 02]

System operacyjny	TCO (3 lata)
Linux	74,475 USD
Solaris	561,520 USD
Windows	190,662 USD

Tabela 3.5 Porównanie 5 letnich okresów TCO dla Windows i Linux [IDC 02]

Zastosowanie	Linux (TCO – 5 lat)	Windows 2000 (TCO – 5 lat)	Relacja kosztów
Serwery sieciowe	13,263 USD	11,787 USD	+12,5%
Serwery plików	114,381 USD	99,048 USD	+15,4%
Serwery drukarek	106,989 USD	86,849 USD	+23,2%
Wer-Serwery	30,600 USD	32,305 USD	-5,6%
Zastosowania chronione	90,975 USD	70,495 USD	+29,1%

Rozstrzygnięcie powyższego sporu leży poza zakresem niniejszej pracy, ale przytoczony przykład dowodzi, że formułowanie sukcesu projektu wdrożeniowego, poprzez kryteria

ilościowe jest niezwykle trudne i kosztowne, a błąd, jakim może to być obarczone, kwestionuje sensowność jego prowadzenia.

„Skoro wydajesz prawdziwe pieniądze - powinieneś stosować prawdziwe miary do oceny ich zagospodarowania. To z kolei przekłada się na prostą konstatację: jeśli nie znajdziesz jakiejś miary w starym podręczniku swojego kierownika do spraw finansowych - zaniechaj jej i nie oceniaj na jej podstawie inwestycji technologicznych”

... stwierdza we wstępie do analizy cROI i uśrednionego ROI analityk Nucleus Research Rebecca Wettemann [CXO 03].

Obraz ten jest zasadniczo różny od tego, jaki wyłania się z badań ankietowych firmy CFO, sponsorowanych przez Getronics i przeprowadzonych w 2001 r. przez firmę Research Services. Wynika z nich, że 86% szefów finansów firm amerykańskich i europejskich bierze pod uwagę przynajmniej jeden miernik przy zatwierdzaniu wydatków na IT. W CFO oceniono inwestycje technologiczne w oparciu o poziom lub okres zwrotu z inwestycji, natomiast indagowani obecnie pracownicy CIO (menadżerowie informatyki, ang. *chief information officer* [cio web]) twierdzą, że dla nich najważniejsze jest, aby przedsięwzięcia były kończone na czas i bez przekroczenia założonego budżetu [CXO 03].

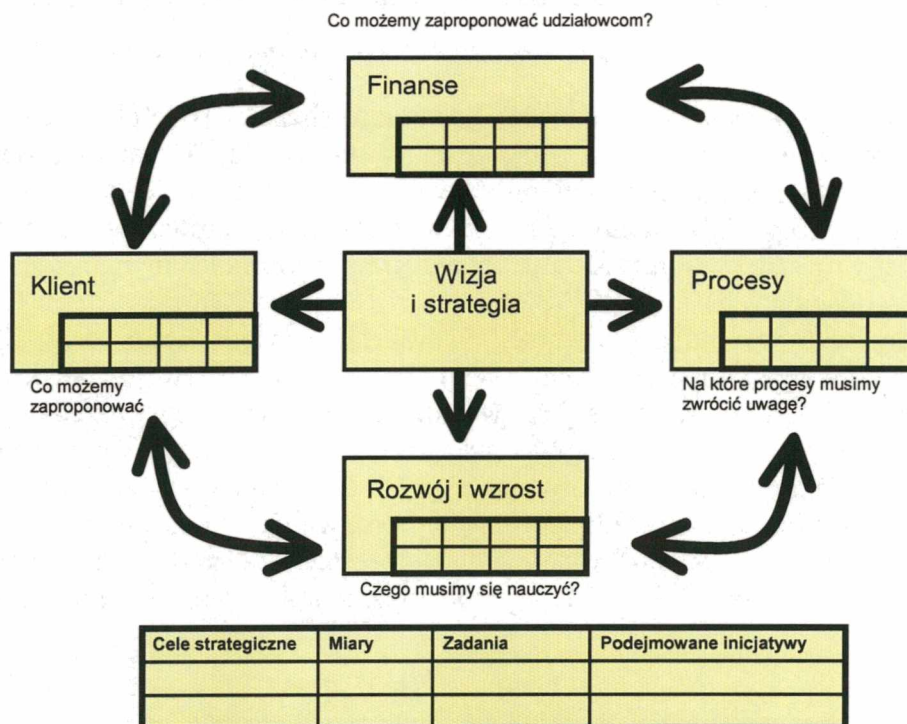
W kontekście „miękkich kryteriów sukcesu” należy zwrócić uwagę także na „jakość”. Jest ona pierwszoplanowym wymogiem dla współczesnych produktów, procesów, organizacji, etc. Jak wynika z tabeli 3.6 „jakość” nie jest cechą wyłącznie ilościową, lecz jej definicje wskazują na ekonomiczne i psychologiczne wartościowane jakości.

Tab.3.6 Różne definicje jakości

Jakość jest to pewien stopień doskonałości	Platon
Jakość jest jedną z dziesięciu kategorii, które umożliwiają podział wszystkich pojęć na grupy logiczne (czas, miejsce, ilość,)	Arystoteles
Przewidywany stopień jednorodności i niezawodności przy możliwie niskich kosztach i dopasowaniu do wymagań rynku	Edwards Deming
Zgodność z wymaganiami	Philip Crosby
Jakość to właściwości wyrobu określone przede wszystkim z punktu widzenia własności chemicznych, fizycznych oraz wartości użytkowej	Mała Encyklopedia Ekonomiczna
Przydatność użytkowa	Joseph Juran
Zbiorcza charakterystyka wyrobu lub usługi z uwzględnieniem marketingu, projektowania, wykonania i utrzymania, która powoduje, że dany produkt lub usługa spełniają oczekiwania użytkownika	Armand Feigenbaum
Jakość może być określana tylko w odniesieniu do wzajemnych powiązań, jakie występują pomiędzy produktem lub usługą, dostawcą i odbiorcą (klientem)	W.E.Deming, 1960
Jakość to zgodność z wymaganiami użytkowników	K.Ishikawa, 1980
Jakość jest tym, czego brak oznacza straty dla wszystkich	G.Taguchi, 1980
Jakość to zero braków	P.B.Crosby, 1985
Jakość to ogół cech produktu lub usługi, które stanowią o jego lub jej zdolności do zaspokojenia wyraźnie określonych lub przewidywanych potrzeb	PN-ISO8402, 1986

Ogół właściwości obiektu, wiążących się z jego zdolnością do zaspakajania potrzeb stwierdzonych i oczekiwanych	ISO 8402
Stwierdziliśmy, że tak jak inne ważne idee, idea jakości jest bardzo prosta - tak prosta, że aż trudna do zrozumienia	Roger Hale, Dyrektor Naczelny Tennant Company, USA
Jakość to wszystko co można poprawić	nieznany
Jakość to nic. Ale wszystko jest niczym bez jakości	nieznany
Jakość to stopień uwolnienia wyrobu od wad i błędów	nieznany

Kategorię jakości znajdujemy po raz pierwszy u Platona (427-347 p. n. e.), Jest ona określona greckim terminem *poiotes*. W kontekście systemu filozoficznego, twórcy idealizmu obiektywnego jakość czegoś konkretnego (przedmiotu, rzeczy) oznaczała stopień osiągniętej doskonałości. Giovanni Quaglia – Dyrektor ds. Rozwoju Modelu EFQM⁴ prezentując model doskonałości podczas „I Międzynarodowej Szkoły Jakości 2000” stwierdził, że jakość ma różne znaczenia, w różnych kulturach. I tak np. w Niemczech jakość to „zgodność ze specyfikacją”, we Francji „luksus”, w Japonii „perfekcja”, w USA „jeśli to działa”, zaś we Włoszech „styl” [Nieścior Web]. Jakość nie może być zatem rozpatrywana wyłącznie w kategoriach ekonomicznych. Stąd też poszukiwanie modeli obejmujących obok finansów, perspektywę klienta, pracownika, strategię przedsiębiorstwa, procesy biznesowe. Jednym z podejść, integrujących perspektywę finansową procesów produkcyjnych klientów i pracowników, jest Metoda Zrównoważonej Karty Wyników BSC (ang. *Balanced Score Card*) [Horvath 01] dla strategicznego zarządzania przedsiębiorstwem, rys. 3.13.

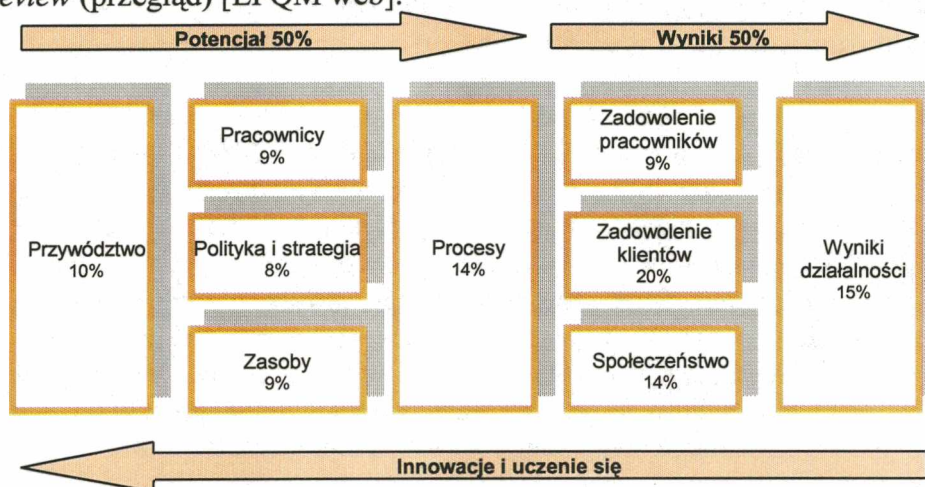


Rys. 3. 13 Zrównoważona Karta Wyników BSC [BCS 00]

Obszerniejszy model dla doskonalenia zarządzania promuje Europejska Fundacja Zarządzania Jakością EFQM (ang. *European Foundation for Quality Management*) rys. 3.13. Organizacja ta powstała w 1988r. z inicjatywy 14 dużych przedsiębiorstw europejskich w odpowiedzi na

⁴ ang. *European Foundation for Quality Management*

ustanowienie rok wcześniej w USA Narodowej Nagrody Jakości Malcolma Baldrige'a (ang. *Malcolm Baldrige National Quality Award*). Model EFQM, bazując na filozofii TQM (ang. *Total Quality Management*) czyli Zarządzania przez Jakość, próbuje zdefiniować główne składniki sukcesu w organizacji, takie jak: koncentracja na kliencie, ciągłe uczenie się, doskonalenie i innowacje, rozwój i zaangażowanie pracowników, zarządzanie przez procesy i fakty, przywództwo i stałość celów, rozwój partnerstwa, odpowiedzialność publiczna, orientacja na wyniki. Model powyższy definiuje 9 kryteriów głównych i podkryteria, które uwzględnia się przy ocenie. Tabele punktacji przyporządkowane poszczególnym podkryteriom oparte są na układzie logicznym o nazwie RADAR - akronimie angielskich słów: *Results* (wyniki), *Approach* (podejście), *Deployment* (wdrożenie) oraz *Assesment* (ocena) i *Review* (przegląd) [EFQM web].



Rys. 3.14 Model Europejskiej Fundacji Zarządzania Jakością EFQM

Krytyczną dyskusję pojęcia „doskonałość” podejmuje Nieścior [Nieścior web] jako rezultat konferencji dotyczącej jakości. Analizując aktualny stan wiedzy w tym zakresie, sięgając do prac Platona i Arystotelesa, dochodzi on do wniosku, że nie ma jednoznacznych definicji doskonałości, a próby dążenia do nich „za wszelką cenę” prowadzą do absurdu. Autor przestrzega, aby rozsądek i logika nie zostały przyćmione ortodoksyjnym dążeniem do modeli i wskaźników ilościowych.

Podobna konkluzja wynika z prezentacji i dyskusji podczas konferencji „Profesjonalne Zarządzanie Wiedzą – Doświadczenia i Wizje” w Lucernie (Szwajcaria), kwiecień 2003. Podczas sesji dotyczącej oceny projektów zarządzania wiedzą nie przedstawiono żadnej propozycji obiektywnych wskaźników sukcesu. Projekty zarządzania wiedzą zostały przyrównane do istniejących procesów biznesowych firm, takich jak marketing, kontroling i kadry, które również nie uzasadniają swojej rentowności [Farnschlaeder, et al 03]. A propozycji mierników, takich jak np. zwiększenie zakresu dokumentacji spotkań projektowych z 20% do 80% [WM 03], nie można traktować jako poważnego głosu w dyskusji.

3.3 Wiedza i automatyzacja

Zarządzanie wiedzą jest tak stare jak ludzkość. Już w czasach kamienia łupanego ludzie przekazywali sobie informacje o trujących roślinach i innych niebezpieczeństwach. Największe wynalazki w dziejach ludzkości, takie jak pismo, druk, telefon, komputer i Internet, wiąże wspólna cecha – poprzez informacje i komunikację wspierają one rozwój wiedzy.

Prekursorami pojęcia „zarządzanie wiedzą” są Druckner [Druckner 93], Probst [Probst, et al. 97], Senge [Senge 98] Nonaka i Takeuchi [Nonaka, Takeuchi 00]. Tematyka zarządzania wiedzą stała się szczególnie modna w ostatnich 3 latach. Wielu autorów wskazuje na rewolucję jaka się dokonuje, tj. przejście z gospodarki industrialnej do postindustrialnej, tzn. opartej na wiedzy. Rozumienie pojęcia „wiedza” nie jest jednak jednolite.

Wiedza, w szerokim rozumieniu ogół treści utrwalonych w umyśle ludzkim w wyniku kumulowania doświadczenia oraz uczenia się. Obejmuje wszystkie formy świadomości społecznej: naukę, ideologię, religię, magię. W takim ujęciu na wiedzę składa się każdy typ myślenia - od wyobrażeń potocznych do twierdzeń naukowych. Może być prawdziwa lub fałszywa, racjonalna lub irracjonalna. W węższym znaczeniu wiedza stanowi osobisty stan poznania człowieka w wyniku oddziaływania na niego obiektywnej rzeczywistości.

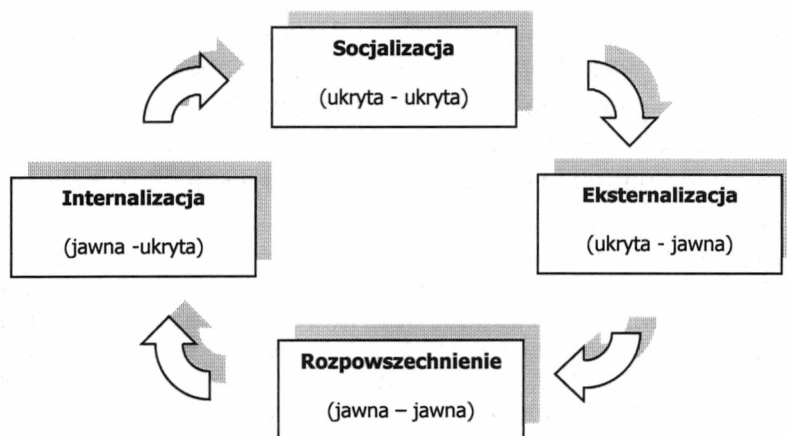
Wyróżnia się dwa rodzaje wiedzy:

- 1) *praktyczną (użyteczną), opartą na doświadczeniu i pozwalającą zmieniać rzeczywistość,*
- 2) *teoretyczną (naukową), opisującą poszczególne aspekty rzeczywistości. [wiem 01]*

Jak wynika z powyższej definicji, „wiedza” związana jest z ludźmi i znajduje się wyłącznie w umysłach ludzi. Wiedza przekazywana poprzez dokumenty czy zapisy elektroniczne to już informacja, a informacja pozbawiona struktury i kontekstu to dane.

Taka interpretacja zakresu znaczeniowego wiedzy, informacji i danych nie jest jednak ogólnie przyjęta. Inna definicja dzieli wiedzę na jawną (ang. *explicite*), zawartą w podręcznikach, dokumentacji, etc. i wiedzę ukrytą (ang. *implicit, tacit*) obejmującą umiejętności. Cempel [Cempel web] uzasadnia, że różne pojmowanie teorii wiedzy (epistemologii) i jej egzemplifikacji wynika z różnej tradycji intelektualnej Zachodu i Japonii. Podczas, gdy na Zachodzie przeważają poglądy atomistyczne i mechanistyczne, to podejście japońskie jest kolektywne i organiczne. Dowodem takiego podejścia są charakterystyczne modele wdrażania zarządzania wiedzą, poszukiwanie biznesowych wskaźników dla projektów wiedzy, *debriefing* oraz modelowanie procesów wiedzy np. K-Modeler [Gronau 03]. W przeciwieństwie do tego Nonaka stawia tezę, że głównym czynnikiem sukcesu organizacji jest umiejętność i biegłość w organizacyjnym tworzeniu wiedzy. Toeffler [Toeffler 96 z Cempel web] pisze „wiedza jest najbogatszym źródłem władzy i kluczem do jej zdobywania, przestała być dopełnieniem władzy opartej na pieniądzu i przemocy, stała się jej rdzeniem”. Stąd obawa, a nawet opór uczestników projektów wiedzy, którzy wychowani i funkcjonujący w tradycji Zachodu są motywowani, głównie finansowo, do dzielenia się wiedzą. Wiara, że nowoczesna technika, tj. systemy informatyczne, jest rozwiązaniem problemów nie przynosi pozytywnych doświadczeń. Wraz z doświadczeniem wzrasta się przekonanie, że zarządzanie wiedzą może być prowadzone bez techniki [Sturz web]. Krytycznym elementem w zaakceptowaniu takiego paradygmatu stają się jednak wymagania biznesowe, tj. potrzeba kosztowej kalkulacji projektów zarządzania wiedzą, co omówiono w Rozdziale 3.2.

Nonaka [Nonaka 00] zwraca uwagę na proces obiegu wiedzy w organizacji, wyróżniając cztery fazy przechodzenia między jej formą jawną a niejawną – patrz rys. 3.15.

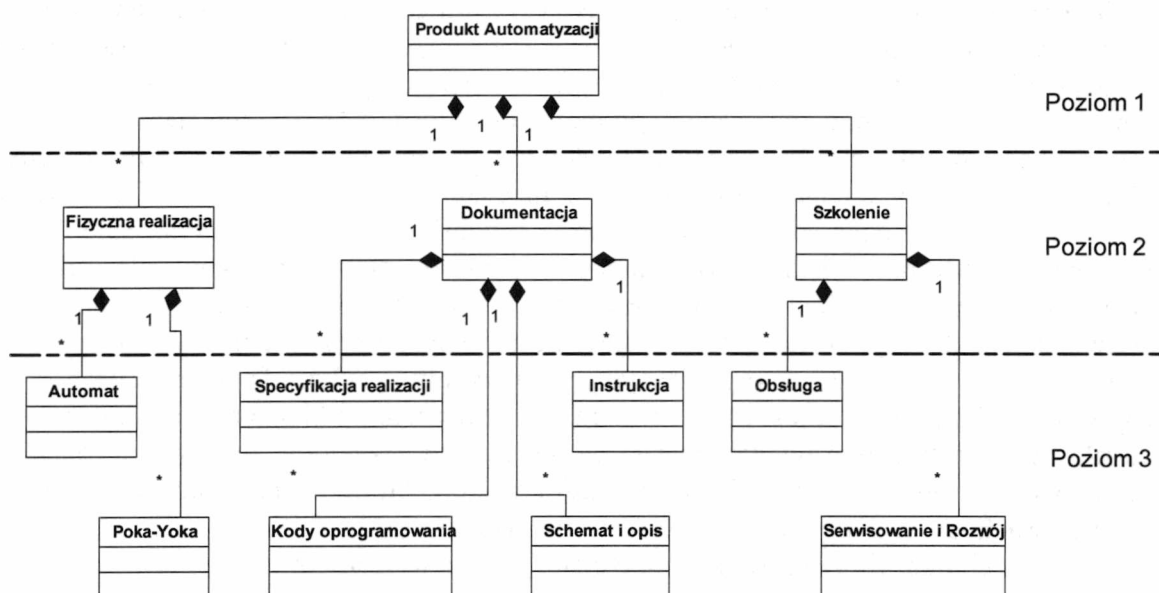


Rys. 3.15 Cykl wiedzy [CTR 96]

Model ten podkreśla konieczność transformacji i komunikacji wiedzy.

3.3.1 Produkty automatyzacji

Klasyfikację produktów automatyzacji przedstawionych na rys. 3.4, rozwinięto jako obiektowy diagram klas – rys. 3.16.



Rys. 3.16 Obiektowy model klas produktów automatyzacji [jr]

Diagram na rys 3.16 nie stanowi kompletnego uszczegółowienia poszczególnych typów produktów, lecz identyfikuje jedynie produkty reprezentatywne. Dalsza ich analiza, w kontekście wiedzy, będzie prowadzona jedynie na drugim poziomie szczegółowości.

Fizyczna realizacja

Fizyczna realizacja, tj. np. zautomatyzowany system produkcyjny, jest głównym produktem automatyzacji. Produkt ten ma przynieść wymierne korzyści biznesowe. Do jego zaprojektowania i wykonania niezbędna była wiedza wykonawcy oraz wiedza użytkownika, sformułowana w postaci wymagań. Produkt ten jest wynikiem wykorzystania wiedzy.

Czy fizyczna realizacja jest nośnikiem wiedzy? Poprzez inżynierię odwrotną (ang. *reverse engineering*) jest możliwy dostęp do „wiedzy ukrytej”, choć w ograniczonym zakresie. Wykonawca, będący autorem rozwiązania docelowego, ma również prawo zastrzec pewne własne rozwiązania. Ochrona taka może być realizowana na drodze technicznej, poprzez kodowanie fragmentów oprogramowania, lub na drodze prawnej tj. poprzez ochronę licencyjną lub zakaz inżynierii wstecznej.

Kiedy dostęp do „wiedzy produktu” jest elementem specyfikacji wymagań, to osiąga się to przykładowo poprzez intuicyjność interfejsu operatora oraz wbudowane elementy pomocy. Decydują one o „przyjazności” urządzenia dla użytkownika. Cecha „*user-friendly*” jest jednak określeniem subiektywnym, dla którego brak miar ilościowych. To, co jeden użytkownik określi „*user-friendly*”, wcale nie musi tak być ocenione przez innego użytkownika.

Dokumentacja

Dokumentacja jest bezpośrednią formą archiwizowania i przekazywania wiedzy o procesie automatyzacji i jej produktach. Z uwagi na jej rozległość jest ona klasyfikowana wg adresatów, tj. ról poszczególnych użytkowników, tworząc różne zestawy tematyczne. Typowo obejmuje ona instrukcje obsługi, dokumentacje techniczne oraz kody źródłowe oprogramowania.

Instrukcje obsługi kierowane są do osób bezpośrednio pracujących z maszynami i urządzeniami. Dotyczą one głównie opisów interfejsów użytkownika, specyfikacji błędów i procedur obsługi.

Dokumentacja techniczna jest skierowana także jako pomoc dla serwisujących urządzenia techniczne. Zwykle za najważniejszy składnik dokumentacji technicznej uznawane są kody źródłowe oprogramowania. Są one niezbędne do re-instalacji, w razie poważnej awarii sterowników lub komputerów. Kody źródłowe są jednak tak szczegółową specyfikacją zaimplementowanych algorytmów i struktur danych, że ich interpretacja celem zrozumienia działania systemu, jest niezwykle trudna, a często niemożliwa.

Choć istnieją zalecenia dot. zawartości dokumentacji technicznej, np. IEC/TC 1131-4:1995, to są one na tyle ogólne, że poziom merytoryczny i „jakość dokumentacji” opisującej zastosowane rozwiązania techniczne nie są możliwe do obiektywnej oceny. Problem ten wynika z braków mierników ilościowych dla informacji, którą zawiera dokumentacja.

Szkolenia i konsultacje

Bezpośrednią formą przekazywania wiedzy o automatyzacji są szkolenia i konsultacje. Efektywność zapoznawania się przyszłych użytkowników z nowym, zautomatyzowanym systemem produkcyjnym nie jest ilościowo możliwa do określenia. Szkolenia takie, wykorzystując bezpośredni kontakt szkolonych z ekspertami rozwiązania, zapewniają najbardziej efektywny transfer wiedzy. Podstawą ich efektywności jest przygotowanie słuchacza oraz odpowiednia metodyka przekazu. Szkolenia podnoszą jednak poziom wiedzy wyłącznie indywidualnej jedynie tych, którzy bezpośrednio w nich uczestniczyli. Wiedzę organizacyjną budują zadania zespołowe, np. ćwiczenia, warsztaty (ang. *workshops*) i projekty grupowe. Formą zapewnienia dostępności do wiedzy w dłuższym okresie czasowym jest tzw. wsparcie techniczne, np. telefoniczne, lub doradztwo i konsultacje.

Wytworzenie każdego z produktów automatyzacji generuje koszty. Trudności z uzasadnieniem ekonomicznym dla inwestycji „miękkich”, powodują oszczędności inwestycyjne właśnie dla tych produktów. Stąd dokumentacja i szkolenia stają się obszarami oszczędności zarówno po stronie wykonawców jak i kupujących.

Wymaganiem koniecznym dla wszystkich trzech produktów automatyzacji, tj. Fizycznej realizacji, dokumentacji i szkoleń, jest ich spójność. Oznacza to, że nie może być

rozbieżności między wynikającą z nich wiedzą. Brak konsystencji pojawia się, gdy następują zmiany techniczne, które nie znajdują odzwierciedlenia w dokumentacji albo wiedza o nich zostaje ograniczona tylko do jednostki, która te zmiany wprowadziła. Planowanie zmian, czy modyfikacje, na podstawie nieaktualnego stanu wiedzy są efektywne, a nawet mogą zagrażać bezpieczeństwu systemu.

Jak wynika z analizy cyklu życia zautomatyzowanego systemu wytwórczego, zapotrzebowanie na wiedzę jest zależne od rozwiązywanych problemów, tj. usuwanie awarii, adaptacja, doskonalenie i zapobieganie awariom etc. Potrzeba zapewnienia dostępu do wiedzy, oraz redukcja kosztów jej pozyskania, znajduje odzwierciedlenie w stosowaniu otwartych rozwiązań automatyki. Rozwiązania otwarte to takie, których specyfikacja jest dostępna. Dzięki niej możliwy jest dalszy rozwój systemu niezależnie od producenta. Przykładami otwartych rozwiązań technicznych dla automatyzacji wytwarzania są obiektowe sieci komunikacyjne *fieldbus* (patrz rozdział 1.2).

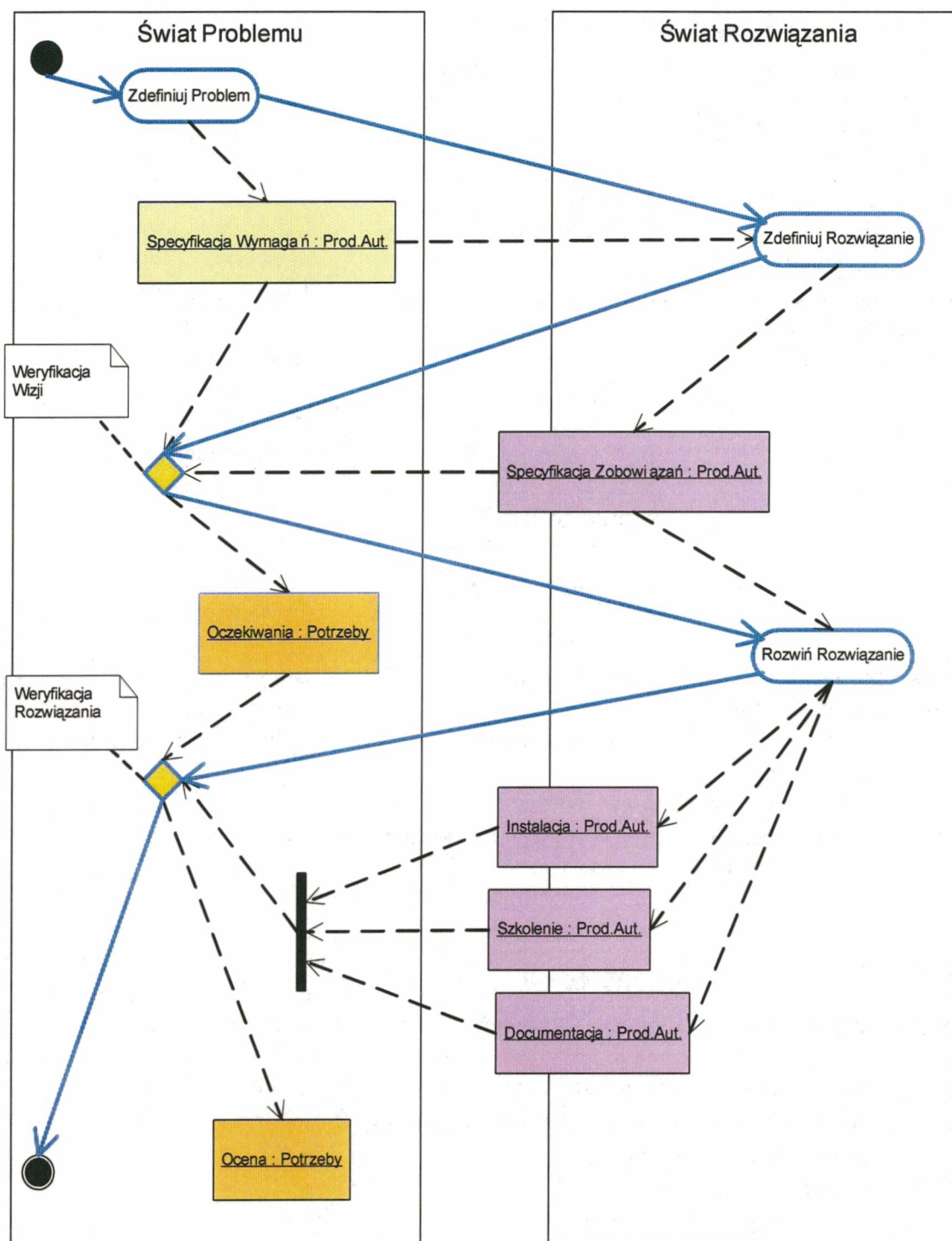
3.3.2 Interakcja między aktorami

Jak wskazano w powyższych rozdziałach, automatyzacja jest procesem „dogadywaniem się” między „tym”, co potrzebuje rozwiązania a „tym”, który może je zaoferować.

„Dogadywanie się” może być analizowane jako proces wymiany wiedzy, informacji i danych.

Jeśli cel strategiczny automatyzacji zostanie określony jako „zadowolenie klienta” (zyski finansowe są pochodną zadowolenia), to o jego osiągnięciu decyduje wynik porównania produktów automatyzacji, oznaczony na rys. 3.17 jako „Weryfikacja wizji” i „Weryfikacja rozwiązania”.

Zwykle klient nie jest ekspertem z dziedziny rozwiązania, a swoją wizję i wymagania formułuje językiem „użytkownika” – jako specyfikację wymagań. Natomiast dostawca usług nie jest specjalistą z dziedziny problemu i swoje rozwiązanie formułuje w języku „realizacji” – jako specyfikację zobowiązań. Taka sytuacja wymusza konieczność porównania przez klienta dwóch specyfikacji. Jeśli przedmiotem porównania staną się szczegóły implementacyjne lub szczegóły poziomu logicznego, a zabraknie gruntownych podstaw strategicznych, to mimo osiągniętego porozumienia, nie będzie ono gwarancją sukcesu. Rozwijane produkty będą wprawdzie zawierały omówione elementy, ale to nie one będą podstawą weryfikacji rozwiązania. Stąd wynika potrzeba abstrakcyjnego formułowania obu specyfikacji. O poprawności porównania nie decyduje zawartość merytoryczna lecz sposób ich przedstawienia. O wyniku porównania, gdy zapewniono jego poprawność, może decydować zawartość merytoryczna. Oznacza to, że specyfikacja zobowiązań może być sformułowana również w języku klienta, ale dalsze jej przetwarzanie będzie wymagało transformacji.



Rys. 3.17 Diagram aktywności – cykl życia automatyzacji i jej produkty [Reiner, Koch 03]

W przypadku weryfikacji rozwiązania, klient dokonuje porównania i oceny tego czego oczekiwał, z tym co otrzymał. W celu ułatwienia weryfikacji, prowadzonej przez klienta, można wyłącznie modyfikować język wymagań klienta, gdyż produkt zdefiniowany jest w języku rozwiązania. Jeśli między rozwiązaniem przekazany w postaci produktów automatyzacji, a oczekiwaniem klienta są rozbieżności, to ich przyczyną są problemy w komunikacji między nimi, lub w ich wewnętrznych procesach. Przyczyny rozbieżności między wizją a wynikiem sklasyfikowano w tabeli 3.7.

Tab.3.7 Przyczyny braku akceptacji dla wyniku realizacji projektu automatyzacji

Przyczyna	Dostawca	Klient
Wizja (potrzeby) nie została zrozumiana należycie przez dostawcę	Zaniedbanie należytego zrozumienia wymagań	Brak weryfikacji rozumienia własnych wymagań
Wizja (rozwiązania) nie została należycie zrozumiana przez klienta	Zastosowano niezrozumiały język rozwiązania	Brak przygotowania do rozumienia języka rozwiązania (wizji)
Wizja nie została dotrzymana przez klienta		Nie udokumentowano wizji, zezwalając na jej równoległy rozwój, niezależnie od procesu realizacji (dwupostaciowość)
Wizja nie została dotrzymana podczas realizacji	Zastosowano zbyt wiele transformacji, które doprowadziły do odstępstwa, wizja klienta nie była modyfikowana	
Produkt nie został zrozumiany przez klienta	Zastosowano niezrozumiały język rozwiązania	Brak przygotowania do rozumienia języka rozwiązania (produktów)

Powyższe przyczyny problemów wskazują na konieczność bliższej analizy łańcucha komunikacyjnego „klient-dostawca”, gdyż to komunikacja jest podstawą przekazywania informacji i wymiany wiedzy.

Powyższe rozważania dowodzą, że ilościowa ocena procesów biznesowych, których podmiotem jest człowiek, jest ciągle nierozwiązana. Procesy te charakteryzują „miękkie” wskaźniki, których wielkość wpływu na wskaźniki finansowe jest subiektywna.

Z uwagi na brak obiektywnych mierników „efektywności automatyzacji”, postulowanej w tezie pracy, nie będą poszukiwane żadne absolutne miary ilościowe. Cele pracy:

- poprawa komunikacji między członkami zespołu,
- ułatwienie zrozumienia problemu,
- ułatwienie zrozumienia rozwiązania,
- poprawa dokumentacji problemu,
- poprawa dokumentacji rozwiązania,

zostaną ocenione subiektywnie poprzez członków zespołu uczestniczących w pilotażowym wdrożeniu zaproponowanego rozwiązania. Przebieg wdrożenia i jej wyniki zostaną zaprezentowane w Rozdziale 5.

3.3.3 Komunikacja w systemie socjo-technicznym

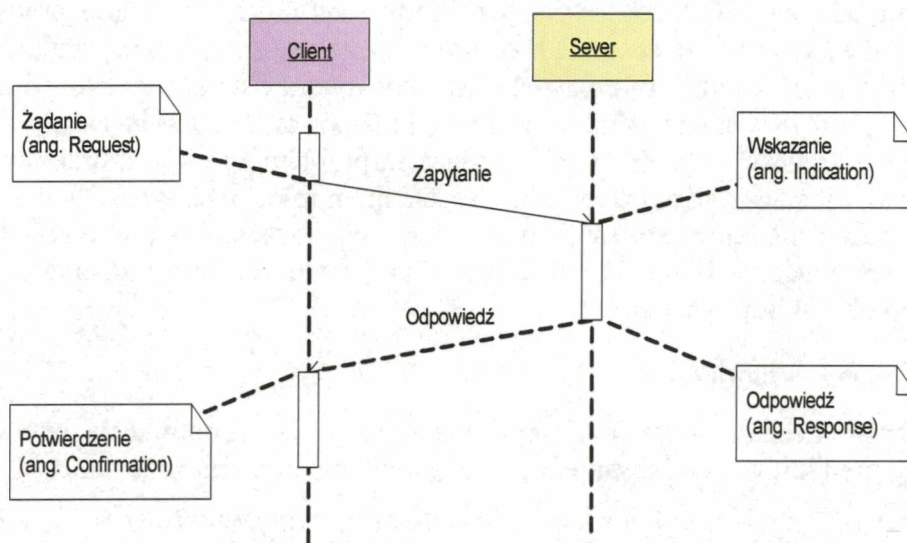
Komunikacja maszyna-maszyna

Rozwiązania techniczne komunikacji sieciowej omówione w Rozdziale 1, tj. sieci obiektowe *fieldbus*, komunikacja na Ethernet TCP/IP i z wykorzystaniem usług Internetu, bazują na

obszernym dorobku teorii komunikacji. Wśród podstaw teoretycznych komunikacji sieciowej szczególnie istotny jest statyczny model ISO/OSI oraz modele typów interakcji. Model ISO/OSI systematyzuje zagadnienia związane z komunikacją w środowisku heterogenicznym, przedstawiając je jako stos siedmiowarstwowy (warstwa fizyczna, łącza danych, sieciowa, transportowa, sesji, prezentacji, aplikacji) [Koch, et al, 97]. Dla każdej z warstw zdefiniowana jest funkcjonalność po stronie nadawcy i odbiorcy oraz interfejsy do warstwy poprzedzającej i następnej. Model ten jest podstawą również dla komunikacji *fieldbus* i Internet, choć nie wszystkie warstwy zostały w nich zastosowane.

Uzupełnieniem modelu statycznego ISO/OSI są modele dynamiczne interakcji między partnerami komunikacji (węzłami sieciowymi). Do podstawowych z nich należy komunikacja typu klient-serwer i *publisher-subscriber* (wydawca-prenumerator).

W modelu klient-serwer, klient jest partnerem komunikacji, korzystającym z wyników dostarczanych przez serwer. Szczegółowy przebieg takiej relacji przedstawiono na rys. 3.18.



Rys. 3.18 Model komunikacyjny klient-serwer

Klient wysyła żądanie (ang. *request*), które po stronie serwera przyjmowane jest jako wskazanie (ang. *indication*). Odpowiedź przygotowana po stronie serwera (ang. *response*) trafia do klienta jako potwierdzenie (ang. *confirmation*). Komunikacja sieciowa przewiduje dodatkowe rozszerzenia powyższej interakcji o potwierdzenia odebrania komunikatu, jak również interakcję pomocniczą, służącą przygotowaniu ostatecznej odpowiedzi.

Drugi podstawowy typ komunikacji to tzw. komunikacja rozgłoszeniowa typu *publisher-subscriber*. Wydawca rozsyła komunikat bez wskazywania konkretnego adresata. Prenumeratorzy to te węzły sieci, które są zainteresowane odbiorem informacji nadawanych przez konkretnego wydawcę. Żaden z prenumeratorów nie potwierdza jednak odebrania wiadomości, co w porównaniu do komunikacji klient-serwer zmniejsza poziom bezpieczeństwa. Zaletą komunikacji typu *publisher-subscriber* jest możliwość dotarcia do szerszego grona odbiorców mniejszym kosztem (efektywność komunikacji) i w krótszym czasie.

Wraz z rozwojem wymagań dla technicznych środków komunikacji maszyna-maszyna, pojawia się zapotrzebowanie na komunikowanie informacji. Oznacza to, że system komunikacyjny powinien mieć funkcjonalność przekazywania nie tylko danych, ale również ich znaczenia. Rozwiązania takie bazują na etykietach definiujących znaczenie przesyłanych danych, np. SGML (ang. *Standard Generalized Markup Language*) [SGML web].

Komunikacja człowiek-maszyna

Drugim typem komunikacji w systemach socjo-technicznych jest komunikacja człowiek-maszyna. Problem ten nie może być uproszczony wyłącznie do zagadnień technicznych, tj. monitor, klawiatura, mikrofon, głośnik, czy ergonomicznych. Obejmuje on również zagadnienia filozoficzne i psychologiczne wykraczające poza sztuczną inteligencję. Zagadnienie komunikacji człowiek - maszyna jawi się pierwszoplanowo dla systemów informatycznych. Zgodnie z przyjętą definicją, że wiedza jest związana z umysłem ludzkim, a informacja z nośnikami technicznymi, to komunikacja człowiek - maszyna wykorzystuje interfejs „wiedza - informacja”. Zgodnie z modelem Cyklu Wiedzy (patrz rys. 3.15) zagadnienia eksternalizacji (ukryta - jawna) oraz internalizacji (jawna - ukryta) to interfejs człowiek - maszyna.

Podczas eksternalizacji sposób przekazywania informacji do maszyny jest ograniczony jej możliwościami. Informatyczne systemy uczące się, np. rozpoznawanie pisma, bazują na ograniczonym zakresie reguł. Ich zakres jest limitowany mocą obliczeniową współczesnych mikroprocesorów, które nie są w stanie konkurować z przetwarzaniem mózgu ludzkiego.

Podczas internalizacji zwykle to człowiek jest aktywnym poszukiwaczem informacji, której potrzebuje w celu pozyskania wiedzy. Dzięki globalnej sieci Internet zasoby informacyjne powiększają się w postępie lawinowym. Aktualnym problemem nie jest brak informacji, lecz jej nadmiar i wynikająca z tego trudność z wyszukaniem informacji właściwej. Rozwiązań dla strukturyzowania i modelowania rozległych informacji poszukuje się w ontologiach [WOW 03]. Sieci semantyczne stanowią teoretyczną podstawę dla prowadzenia wnioskowania zawężającego obszar odpowiedzi.

Komunikacja człowiek-człowiek

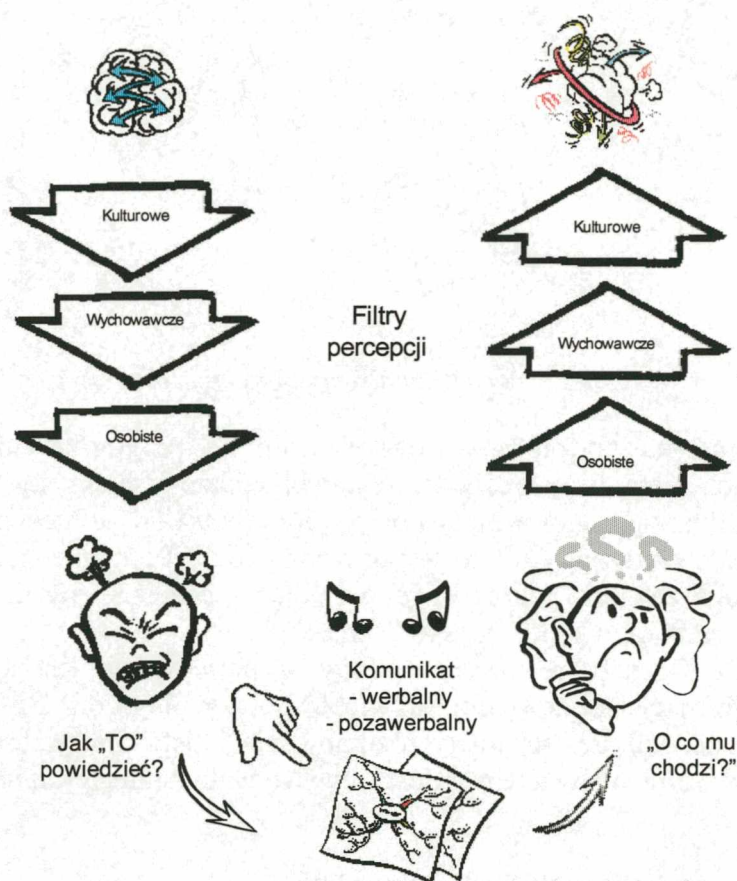
Najważniejszym mechanizmem przekazywania wiedzy jest komunikacja interpersonalna, z inżynierskiego punktu widzenia wspierana przez telefonię oraz multimedia.

Komunikacja interpersonalna (z łaciny communicatio - rozmowa, wymiana, łączność), proces, podczas którego ludzie dążą do dzielenia się znaczeniami za pośrednictwem symbolicznych (dźwięki, litery, słowa) informacji (komunikatów). Najprostszy model komunikowania się polega na przekazywaniu przez nadawcę komunikatu (werbalnego lub niewerbalnego) i odebraniu go przez odbiorcę. Komunikowanie się może być realizowane przez wypowiedzi ustne, pisemne i różne formy wizualne oraz tzw. mowę ciała (z angielskiego body language), np. różnego rodzaju gesty, barwę i ton głosu, mimikę twarzy. Komunikowanie się może być jednokierunkowe - nadawca przekazuje informacje bez oczekiwania ich potwierdzenia przez odbiorcę, lub dwukierunkowe - nadawca uzyskuje potwierdzenie przekazanej informacji, np. w formie pytań zadawanych przez odbiorcę. [Smolski et al. 99]

Komunikacja jest procesem kodowania i dekodowania informacji. Autor, chcąc wyrazić swoją koncepcję mentalną, najpierw tworzy obraz przestrzenny w umyśle. Obraz ten powstaje na podstawie przyjętego celu, dostępnego zestawu elementów języka, często również z uwzględnieniem predyspozycji odbiorcy. Następnie, obraz przestrzenny jest materializowany poprzez szkic, zapis lub wypowiedź. Uzyskana forma prezentacji w dużym stopniu zależy od predyspozycji technicznych lub językowych nadawcy. Komunikat dociera do odbiorcy, mającego swoje oczekiwania i nastawienia, co determinuje dekodowanie. Proces komunikacyjny podlega zakłócaniu, poprzez czynniki obiektywne np. niedosłyszenia, hałas i subiektywne tj. filtry percepcji (rys. 3.19). Filtry percepcji kształtowane są przez wyobrażenia

i wartości kulturowe, wyobrażenia i wartości przejęte w procesie wychowania oraz własne doświadczenia. Dla określenia tych cech używane jest pojęcie „osobistego modelu świata”. Powyższy proces można zatem określić jako dwupoziomowe kodowanie informacji. Proces u odbiorcy jest odwrotny. Najpierw odbiorca tworzy własny model wizualno-przestrzenny, który podlega przetwarzaniu i zapamiętywaniu tworząc koncepcję mentalną odbiorcy. Mechanizmy takie zachodzą podczas komunikacji zarówno wizualnej jak i werbalnej, opierając się w obu przypadkach na zbiorze wspólnych założeń. Założenia takie obejmują syntaktykę i semantykę języka. Zatem, dla każdej z form komunikacji nie istnieje bezpośrednie powiązanie między systemem oryginalnym (źródłowym) a systemem mentalnym odbiorcy. Proces kodowania - dekodowania z wielostopniową interpretacją, jest zależny od wiedzy kontekstowej, czynników kulturowych oraz predyspozycji indywidualnych autora i odbiorcy i nie jest powtarzalny. Dlatego też filozoficzne rozumienie komunikacji wymaga rozszerzenia prezentacji o kanał zwrotny. W ten sposób autor jest informowany zwrotnie i może adaptacyjnie wpływać na swój proces kodowania. Poprawa w dowolnym ogniwie łańcucha ma wpływ na proces komunikacyjny.

Doświadczenia kulturowe – są charakterystyczne dla miejsca, w którym żyjemy. Wynikają one z ogólnej koncepcji świata, która jest opisana w meta-zasadach, ustalają hierarchię wartości, powołują postacie symboliczne, kształtują pojęcie czasu, grupy i jednostki. Na tym poziomie, widzenie świata jest spójne i całkowite, oferując człowiekowi poczucie więzi, celu i sensu.



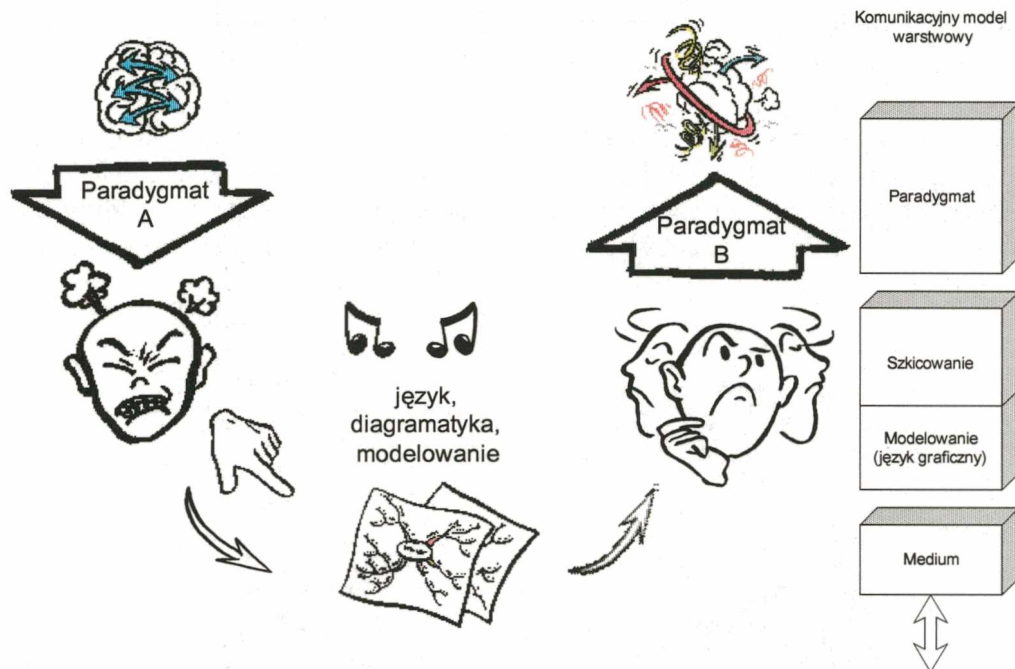
Rys. 3.19 Filtry percepcji w komunikacji międzyludzkiej [jr]

Doświadczenia wychowawcze – to normy i wartości kręgu kulturowego. W stosunku do pierwszego filtra (kulturowego) tu funkcjonują informacje przetworzone, uproszczone i

zniekształcone, a czasem odstępstwa na rzecz potocznej interpretacji. Na tym poziomie subiektywnie definiowane są pojęcia pracy, pieniędzy, bogactwa, honoru, odpowiedzialności, czystości, sprawiedliwości, prawdy, dobra, uczciwości, warunki szczęścia, itd. oraz relacje „my - świat” i „ja - świat”.

Doświadczenie osobiste – to indywidualny bagaż doświadczeń, który powstał w wyniku sukcesów i porażek, efekt tego, czego się nauczyliśmy i doświadczaliśmy. Jest ono zależne od właściwości osobniczych i neurologicznych preferencji oraz potrzeb.

Zestawienie filtrów percepcji w komunikacji międzyludzkiej (rys. 3.19) pozwala zamodelować ją na podobieństwo modelu warstwowego ISO/OSI. Łańcuch taki zawiera warstwę paradygmatu, szkicowania i modelowania, co przedstawiono na rys. 3.20.



Rys. 3.20 Łańcuch komunikacji interpersonalnej [jr]

Jeśli wiedza automatyka koncentruje się wyłącznie na programowaniu, to jego język komunikacji z otoczeniem w zakresie tej tematyki będzie wykorzystywał specjalistyczne elementy. Stąd wynika popadanie w „bity i bajty”, które z uwagi na szczegółowość nie mogą być zrozumiałe dla szerszego grona współrozmówców. Podobnie język i wymagania technologiczne, mogą zawierać tak duży zakres pojęć specjalistycznych, że komunikacja z nieprzygotowanym partnerem będzie nieskuteczna.

Paradygmat określa sposób w jaki tworzony i pojmowany jest obraz mentalny i interpretowane informacje z zewnątrz: dźwięki, obrazy. Jest on kształtowany poprzez posiadaną już wiedzę, kulturę, schematy rozumowania i listy priorytetów. Oznacza to, że paradygmat może być zmieniany a nawet dostosowywany do istniejących potrzeb.

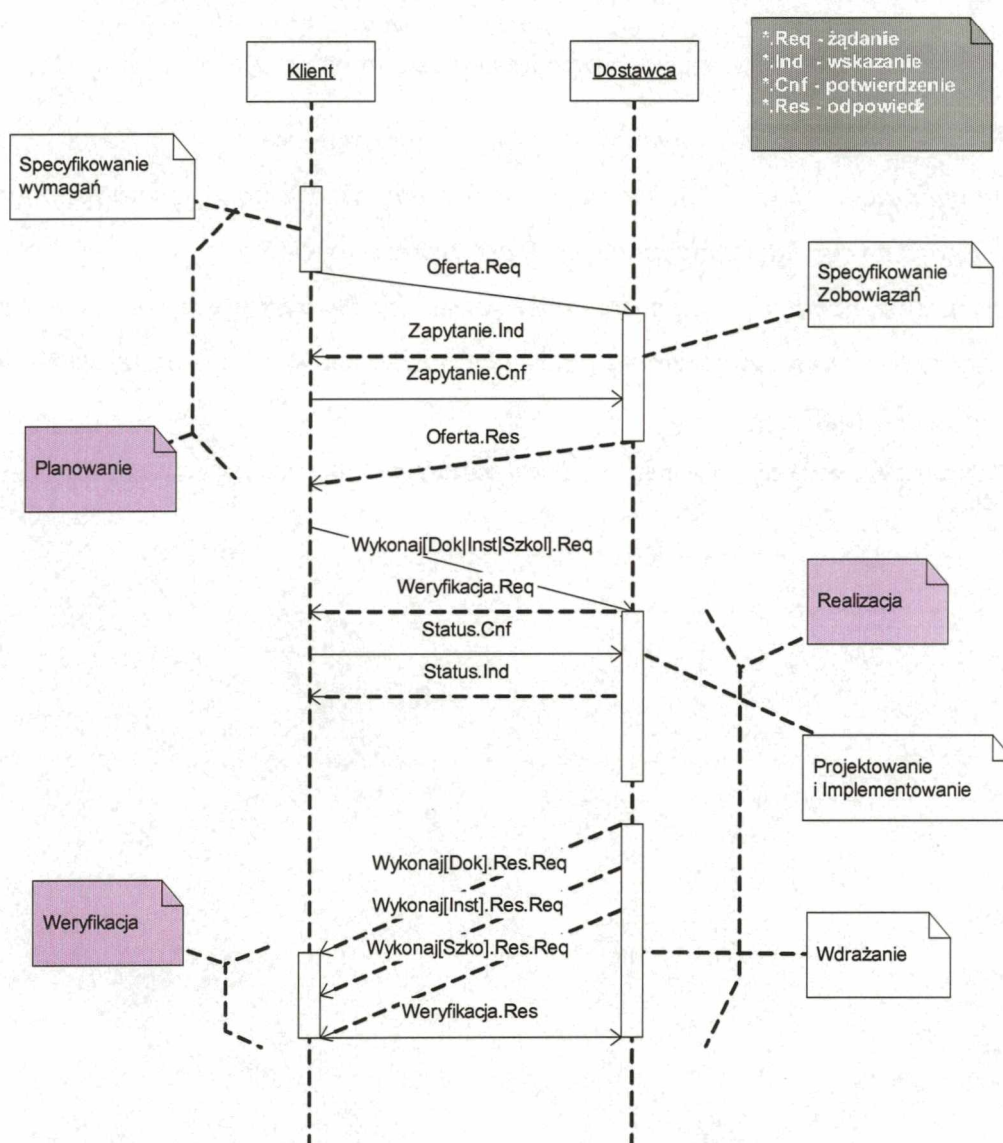
3.3.4 Komunikacja podczas cyklu życia automatyzacji

Analizując komunikację między aktorami w cyklu życia automatyzacji wytwarzania, zauważamy analogię do systemów technicznych, wyróżniających dwa jej kierunki:

- komunikację poziomą (wewnętrzna) – w ramach jednego zespołu,
- komunikację pionową (zewnętrzna) – między zespołami.

Podczas eksploatacji i utrzymania ruchu zautomatyzowanego systemu wytwórczego występuje głównie komunikacja wewnętrzna w grupach opiekujących się zautomatyzowanym systemem produkcyjnym. Typowo są to zespoły interdyscyplinarne, reprezentowane przez role technologa, inżyniera jakości, automatyka, serwisanta i planisty. Wielkość zespołów jest zależna od skali obsługiwanej instalacji. Każda z ról charakteryzowana jest przez posiadane kompetencje merytoryczne, tj. wiedzę specjalistyczną. Nieodzowne są również kompetencje społeczne określające umiejętności współpracy w grupie, rozwiązywania konfliktów, etc. Zadania scharakteryzowane w Rozdziale 3.1.4, dotyczącym eksploatacji i utrzymania, wymagają doskonałej współpracy różnych specjalistów.

Bazując na modelu klient - serwer dla komunikacji sieciowej, opracowano scenariusz interakcji komunikacyjnych klient – dostawca usług. Scenariusz przedstawiony na rys. 3.21 obrazuje interakcje komunikacji pionowej (zewnętrznej) podczas cyklu życia automatyzacji.



Rys. 3.21 Scenariusz interakcji klient – dostawca podczas automatyzacji [jr]

Pary relacji *Zapytanie.ind* (wskazanie) i *Zapytanie.cnf* (potwierdzenie) oraz *Status.cnf* (potwierdzenie) i *Status.ind* (wskazanie) (rys. 3.21), nie określają pojedynczych interakcji typu pytanie odpowiedź, lecz całe pakiety informacyjne.

Opracowany diagram modeluje przepływ aktywności i dokumentów (ang. *workflow*) podczas realizacji projektu automatyzacji. Może on być wykorzystany jako szablon dla wspierających systemów informatycznych.

3.4 Podsumowanie rozdziału trzeciego

W powyższym rozdziale dokonano analizy cyklu życia automatyzacji wytwarzania, zwracając uwagę na jej użytkowników i produkty. Analizę produktów automatyzacji przeprowadzono pod kątem wiedzy. Przeprowadzona analiza interakcji między obszarem problemu a rozwiązaniem, pod kątem sukcesu automatyzacji, wskazała na szczególnie istotną rolę komunikacji interpersonalnej. W zamodelowanym łańcuchu komunikacyjnym wyróżniono paradygmat obejmujący filtry percepcji, język i medium komunikacyjne. Jak wykazano, ogniwa te decydują o efektywności porozumiewania się między członkami zespołów.

W wyniku badań przedstawionych w Rozdziale 3 wykazano, że:

- 1) Automatyzacja jest kompleksowym procesem w systemie socjo-technicznym, podczas którego następuje intensywne przetwarzanie wiedzy.*
- 2) Produkty automatyzacji są nośnikami informacji i źródłem wiedzy o automatyzacji.*
- 3) Podstawą efektywności automatyzacji jest posiadana wiedza oraz jej komunikacja między aktorami.*
- 4) Efektywna komunikacja wymaga zgodnych paradygmatów oraz języka.*

4 Kompetencje dla modelowania obiektowego i UML

Dla zbadania słuszności postawionej tezy, w Rozdziale 4 skoncentrowano się na pierwszej części postulatu:

„Poprawę efektywności automatyzacji wytwarzania można osiągnąć dzięki kompetencjom w zakresie modelowania obiektowego i języka UML oraz poprzez odpowiednie nimi zarządzanie.”

Dla Rozdziału 4 postawiono następujące pytania badawcze:

4.1 Komunikowanie wiedzy

- *W jaki sposób komunikować abstrakcję?*
- *Jak wykorzystać modelowanie do komunikacji wiedzy?*

4.2 Programowanie i modelowanie w automatyzacji

- *Co to jest program?*
- *Na czym polega idea bloków funkcyjnych?*
- *Czy paradygmat automatyzacji jest efektywny?*

4.3 Obiektowość i język UML

- *Co charakteryzuje paradygmat obiektowy?*
- *Jakie są podstawowe elementy języka UML?*

4.4 Obiektowość i UML dla automatyzacji wytwarzania

- *Jak wykorzystać UML do komunikacji i dokumentacji w automatyzacji wytwarzania?*
- *Jak modelować biznesowe procesy automatyzacji?*
- *Jak modelować logiczną strukturę procesów?*
- *Jak modelować zakres i funkcjonalność procesów?*
- *Jak modelować dynamikę systemu?*
- *Czy MOAW jest narzędziem zarządzania wiedzą?*

4.1 Komunikowanie wiedzy

4.1.1 Diagramatyka

Szkice i rysunki należą z pewnością do najstarszych form ludzkiej komunikacji. Jednakże, nie są one używane wyłącznie do wyrażania i uzewnętrzniania idei oraz utrwalania przelotnych myśli. Oczywiście, język pisany też to potrafi, lecz szkice mają tę przewagę, że przenoszą wizualno-przestrzenną ideę bezpośrednio, poprzez użycie elementów i relacji przestrzennych na papierze, do komunikacji z elementami i relacjami przestrzennymi w świecie. Ponadto, szkice potrafią przenosić abstrakcyjne idee metaforycznie, prezentując informację niejawną. Wyrażanie idei poprzez medium wizualno-przestrzenne ułatwia rozumienie i wnioskowanie, w porównaniu do bardziej abstrakcyjnych mediów, jak np. języki. Zewnętrzność szkiców i podobnych narzędzi poznawczych stymuluje pamięć, dostarczając zapisu niezależnego od zawodności ludzkiej pamięci. Stanowią one również materiał dla mózgu, który równocześnie zapamiętuje go i przetwarza [Tversky 02]. Publiczna cecha szkiców udostępnia je także

innym osobom, dając możliwość obserwowania, komentowania i oceny ich w zewnętrznej reprezentacji.

Szkice, podobnie jak język mówiony czy pisany, stanowią formę komunikacji. Jednak szkice, bardziej jak mówienie czy pisanie, są formą komunikacji z samym sobą. Jedną z ról szkiców jest kontrola kompletności i wewnętrznej konsystencji idei, szczególnie idei przestrzennych. Szkic może być zatem dosłownym modelem idei i dowodem jej istnienia. Dzięki szkicom, ich autorzy odkrywają nowe relacje i obrazy, co z kolei stymuluje powstawanie nowych idei [Tversky 02].

Najbardziej powszechnym szkicem jest mapa. Niezależnie od tego, czy rozważamy mapę regionów czy połączeń, zawsze jest to schemat, który zawiera informację ważną dla swego przeznaczenia, pomijając to, co nie jest istotne. Szkice często jednak upraszczają, a nawet wypaczają informację źródłową. Struktura uchwycona przez szkice nie jest zatem strukturą rzeczywistości lecz raczej koncepcyjną strukturą informacji. Często zawierają one elementy opisujące, których nie ma w rzeczywistości np. granice, strzałki oraz elementy symboliczne jak nazwy, liczebności i odległości.

Szkice z innych dziedzin niż kartografia, cechuje podobny charakter, to znaczy: składają się z kombinatorycznego zestawu stylizowanych elementów. Nigdy jednak nie są prezentacją rzeczywistości, lecz jej reprezentacją. Dlatego pomijają one, dodają, a nawet zniekształcają informację.

Używane potocznie zdanie „koń, jaki jest – każdy widzi” [Chmielowski] jest trywializowane, z perspektywy bezwzględności przekazu. Ludzie różnie potrafią odczytać ze szkiców to, co one zawierają. Dlatego na przykład często wracamy do dawnych zapisków zdumieni informacją, jaką one niosą.

Szkice dla diagramowania są strukturyzowane. Składają się one z elementów i segmentów, które dzięki kombinatorycznym zestawieniom mogą tworzyć wiele znaczeń. Elementy te są zwykle prostymi, schematycznymi figurami, takimi jak np. okrąg, odcinek, których znaczenie może być łatwo odczytywane z reprezentacji graficznej lub zestawienia. Struktura diagramów jest podobna do języków mówionych. Jednak w przeciwieństwie do języków, diagramy oddziałują poprzez cechy kształtowe i relacyjne bezpośrednio na pamięć, rozumienie i wnioskowanie.

Zmieniające się uprzedzenia w świecie nauki do reprezentacji diagramowej oraz prace nad wielo-modalnym rozumowaniem prowadzą do bardzo dynamicznego rozwoju trzech nowych obszarów badań określanych jako „Diagramatyka” (Diagramowanie) [SEofP 02].

Pierwszy nurt badań dotyczy filozofii rozumowania i nauki kognitywnej. Odkąd granice form lingwistycznych są rozpoznane dla tych, którzy zajmują się reprezentacją umysłu i rozumowaniem, wielu filozofów i naukowców kognitywnych z dużym entuzjazmem prowadzi badania nad wielo-modalnym rozumowaniem z wykorzystaniem form poza lingwistycznych.

Drugi obszar badań wykazuje, że nie ma żadnej wewnętrznej różnicy między systemami symbolicznymi i diagramowymi. Na podstawie przypadków użycia naukowcy dowodzą, że systemy diagramatyczne mogą być równie solidne i kompletne jak systemy symboliczne. Badania te obalają szeroko rozpowszechnione przekonanie, że diagramy są z natury mylące i nie mogą być wykorzystywane jako materiał dowodowy.

Trzeci obszar badań nad wielo-modalnym rozumowaniem jest rozwijany szczególnie przez naukowców z dziedzin związanych z komputeryzacją, których zainteresowania są zdecydowanie bardziej pragmatyczne. Próbuje oni wykorzystać diagramowanie dla reprezentacji wiedzy, projektowania systemów programowania graficznego, projektowania interfejsu graficznego dla użytkownika oraz podobnych.

Diagramatyka jest obszarem zainteresowań psychologii, antropologii, lingwistyki, filozofii, informatyki (ang. *computer-science*) i inżynierii neuronalnej (ang. *neuro-science*).

Material wizualny

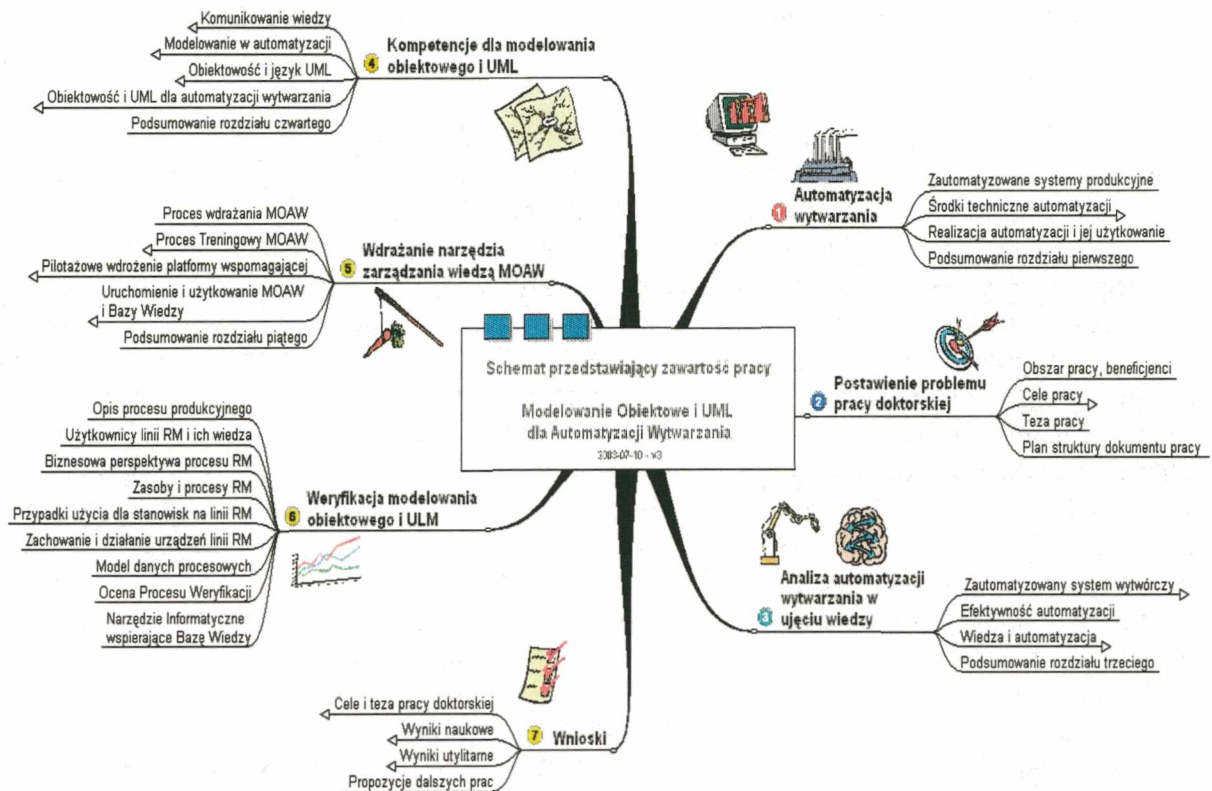
Poszukiwanie gruntownych podstaw dla wykorzystania materiałów wizualnych zwraca uwagę na elementy procesu percepcji, stosowane systemy symboli i konwencji, wpływ posiadanej wiedzy merytorycznej o dziedzinie problemu oraz strategię widzenia.

Procesy percepcji obejmują organizację naszego widzenia, która przebiega zwykle w sposób automatyczny i pre-symboliczny metodą uogólniania (ang. *bottom-up*). W pierwszej fazie następuje interpretacja tego, co widzimy. Rozróżnienie elementów jeden od drugiego następuje poprzez dyskryminację. Efekt „pop-up” Treismana mówi o wyszukiwaniu szeregowym dla wielowymiarowej zmienności cech oraz przeszukiwaniu równoległym, gdy cecha jest jednowymiarowa. Dyskryminacja może bazować na kolorze, kształcie, barwie, itp. Kolejny proces to organizacja cech, czyli łączenie i grupowanie, które wykorzystują powinowactwo, bliskość, kolor i połączenia. System symboli może obejmować tekst (listę) lub grafikę. Winn, Li & Schill wykazują różnice w interpretacji drzewa genealogicznego, gdy dane wejściowe podano w sposób tekstowy i graficzny. Celem znalezienia odpowiedzi na pytanie o powinowactwo konieczne jest przeszukiwanie i wnioskowanie a nie przypominanie. Badania prowadzono na trzech poziomach zaznajomienia, tj., gdy reguły powinowactwa nie były znane, ze znajomością reguł języka angielskiego, gdy dane zawierały nonsensy. Miara efektywności porównania był czas znajdowania odpowiedzi. W miarę wzrostu trudności zadania, czasy odpowiedzi się skracały, zaś praktyka badanego wpływała negatywnie na wyniki badań. Badania dotyczące wpływu posiadanej wiedzy kontekstowej dowodzą wpływu na to, co widzimy jak również, jaka jest nasza interpretacja wizualizacji oraz czego poszukujemy. Badania Winn & Sutherland'a [Winn, Sutherland 89] porównują zdolność przypominania lokalizacji i nazw elementów w przypadku mapy i diagramów symbolicznych. Wykazano w nich, że zdolność ta jest gorsza, gdy elementami szkicu są jednakowe symbole. W przypadku rysunków zdolność przypominania zależy w głównym stopniu od zastosowanej strategii pomocniczej. Strategia taka jest jednak trudna do znalezienia wówczas, gdy cechy obrazu są podobne.

Mapy i mapowanie pamięci (ang. *mind mapping*)

Badania nad skutecznością notatek, prowadzone przez T.Buzana w latach 60. wykazały, że najwyższy współczynnik zapamiętania osiąga się, gdy słuchacze notują jedynie słowa kluczowe. Równocześnie R.Haber argumentował, że współczynnik sukcesu przy rozpoznawaniu wzorców graficznych przez człowieka może wynosić 85% a nawet 95%. Powyższe obserwacje zaowocowały opracowaniem przez T.Buzana techniki diagramowej prezentacji koncepcji - zwanej Odwzorowywaniem Pamięci (ang. *mind mapping*) [Buzan 89,91]. Podstawą tej metody było wsparcie skojarzeniowych funkcji mózgu. Dlatego bazowała ona na tworzeniu notatek możliwie jak najkrótszych i prezentowanych w sposób interesujący i przyjemny dla oka ludzkiego – patrz rys. 4.1.

Technika opracowywania map umysłu polega na poszukiwaniu elementów i relacji otaczających główną ideę. Poprzez koncentrowanie się na centralnym elemencie tworzony jest obraz powiązań i rozszerzeń. Tworzenie mapy idei jest nieliniowym poszukiwaniem skojarzeń, które dla wzmocnienia wyrazu tworzonej mapy stosuje symbolikę i kolorystykę. Nie ma żadnych ograniczeń definiujących formę mapy, a wręcz zalecane jest poszukiwanie różnorodnych prezentacji, dzięki którym znacznie ułatwia się identyfikację i zapamiętywanie map.



Rys. 4.1 Przykład mapy pamięci [jr]

Mapowanie koncepcji jest narzędziem pomagającym i rozszerzającym wiele form kreatywnego myślenia i uczenia się. Mapy postrzegane są jako forma „burzy mózgów” (ang. *brainstorming*). Zachęcają one do generacji nowego materiału a w przeciwieństwie do „burzy mózgów” ograniczają pole poszukiwań i zmniejszają rolę przypadkowych pomysłów niezwiązanych z tematem. Forma generowanych wyników nie jest liniowa, lecz zawiera elementy hierarchii ważności, wzajemne zależności i grupowanie, równocześnie wykluczając bądź wskazując sprzeczności.

Jako najważniejsze cechy odwzorowywania mapy umysłu wymienia się:

- mocną definicję myśli głównej,
- możliwość nieuporządkowanego wprowadzania informacji,
- redukcję opisów koncepcji do zwięzłych etykiet,
- możliwość nadawania poziomów ważności,
- łatwość pokazywania powiązań i hierarchii,
- możliwość przedstawienia całej koncepcji na jednym widoku,
- łatwość późniejszego rozszerzania informacji,
- łatwość wnioskowania i odszukiwania informacji poprzez różne widoki,
- możliwość pokazywania skomplikowanych powiązań i relacji np.: pętli,
- łatwość identyfikacji sprzeczności i niejasności („białe dziury”).

Dzięki wielu zaletom podkreślanym przez entuzjastów diagramowania pamięci, metoda ta znalazła zastosowanie nie tylko przy sporządzaniu notatek [Buzan, Buzan 1993]. Z dużym sukcesem jest ona stosowana w celu:

- odkrywania nowych kreatywnych możliwości zadanego tematu,

- uściślenia wcześniejszych przypuszczeń i założeń,
- generowania idei dla konkretnych zadań,
- pobudzania bardziej spójnego kreatywnego myślenia,
- uchwycenia i rozwijania pomysłów,
- generowania nowych szablonów koncepcyjnego postępowania,
- planowania kreatywności.

Firma Mindjet [Mindjet 02], w raporcie z badań firm stosujących technikę map pamięci, przytacza ilościowe wskaźniki poprawy efektywności:

- redukcja liczby spotkań podczas wdrażania systemu IT o 66% - sieć sklepów bezcłowych w San Francisco,
- redukcja czasu oceny projektów o 50% - IBM,
- skondensowanie 30 stronicowego dokumentu na jednostronicowej mapie,
- zwiększenie uczestnictwa w naradach projektowych o 100% - HP, dotyczy projektów międzynarodowych.

Mapy pamięci wspierają skojarzeniową logikę umysłu a nie logikę liniowego wnioskowania i dlatego też są trudne do przetwarzania przez komputery. Programy wspierające tworzenie map umysłu pozwalają głównie na ich edycję, oferując bogate biblioteki symboli, ikon i elementów graficznych. Żaden z dostępnych programów nie wspiera zarządzania zmianami, ponieważ w tym celu wymagana byłaby interpretacja mapy. Przykładami takiego oprogramowania są MindManager [MindManager], VisualMind [VisualMind], Inspiration [Inspiration], VISIO [Visio], MindMapper [MindMapper], ConceptDraw [ConceptDraw], etc.

Technika odwzorowywania mapy (mapowania) pamięci jest efektywna wówczas, gdy problemy odnoszą się do przetwarzania koncepcji. Gdy poziom szczegółowości wzrasta i próbujemy odpowiedzieć na pytanie „JAK?”, to znacznie efektywniejsze są inne techniki specjalizowane diagramowania, ułatwiające przedstawienie struktury logicznej i działania systemu.

4.1.2 Model i modelowanie

Autorzy UML Grady Booch, James Rumbaugh i Ivar Jacobson, wprowadzając w świat modelowania, przytaczają niezwykle obrazowy przykład z budownictwa:

„(...) gdy chcesz zbudować psią budę, możesz rozpocząć od zgromadzenia desek, gwoździ, (...) . W kilka godzin, obywając się bez wcześniejszego planowania robót, jesteś w stanie postawić coś takiego. Twój pies będzie Ci wdzięczny - pod warunkiem, oczywiście, że buda będzie wystarczających rozmiarów, (...). Jeśli Ci się nie uda realizacja tego przedsięwzięcia, możesz wszystko zacząć od początku, lub zmienić psa na mniej wymagającego.

(...) gdy zechcesz zbudować dom dla rodziny, też możesz zacząć od zgromadzenia desek i gwoździ, (...). Jeśli nie zbudowałeś wcześniej wielu domów, lepiej byłoby zacząć od szczegółowego planowania, (...). Pamiętaj, że w razie niepowodzenia trudno będzie zmienić rodzinę.

(...) gdy chcesz zbudować wielopiętrowy biurowiec, głupotą byłoby rozpoczynać od zgromadzenia desek i gwoździ (...). W takim przypadku zazwyczaj wydaje się cudze pieniądze; inwestorzy chcą mieć wpływ na rozmiar, kształt i styl budynku. Często się zdarza, że zmieniają zdanie, - i to wtedy, gdy budowa idzie pełną parą. (...) koszt

niepowodzenia jest bardzo wysoki. (...) będziesz członkiem dużego zespołu projektującego i realizującego biurowiec." [UML 02]

Konkluzja autorów jest niezwykle wymowna:

„Zadziwia fakt, że wielu wytwórców oprogramowania pragnie wznosić drapacze chmur, ale przystępują do tego jak do zbijania psiej budy”

Model i modelowanie należą do tych pojęć, które z uwagi na nadużywanie, straciły jednoznaczność znaczeniową. Słownik Wyrazów Obcych [SWO web] podaje aż osiem definicji pojęcia „model”:

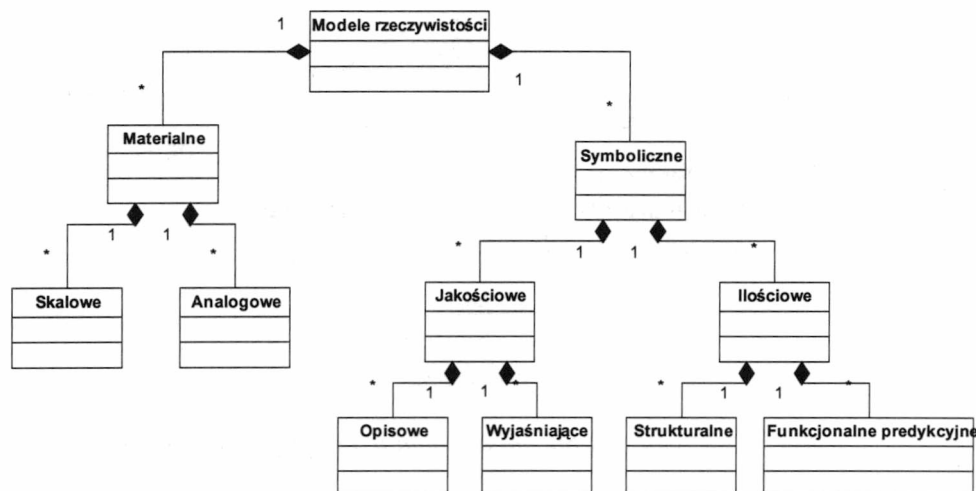
model (fr. modčle z wł. modello zdrobn. od modo z łc. modus 'miara')
 1. schemat, wzór, wg którego coś można wykonać. 2. wykonany w określonej skali (gł. pomniejszony) obiekt, urządzenie itp., służące m.in. do prezentacji wyglądu, zasady działania. 3. uproszczona struktura funkcjonowania jakiegoś układu elementów, np. działania urządzenia, gospodarki, przedsiębiorstwa. 4. męczyzna prezentujący ubiory w trakcie pokazu mody; prezenter. 5. szt. męczyzna pozujący artyście, gł. malarzowi, rzeźbiarzowi, fotografowi. 6. fiz. m. atomu - planetarny model budowy atomu opracowany w 1913 r. przez N. Bohra, w którym elektrony obiegają po różnych kołowych orbitach centralnie usytuowane jądro atomu. 7. infor. symulacja obiektu, procesu, zjawiska lub działania urządzenia wykonana za pomocą specjalistycznych programów komputerowych. 8. technol. m. odlewniczy - zespół przyrządów i przedmiotów układanych na kształt odlewu, stosowanych do wielokrotnego wyrobu określonych form odlewniczych.

Z powyższej, wieloznaczeniowej definicji modelu, terminy 1, 2, 3 i 7 odpowiadają poszukiwanemu znaczeniu, choć znacznie praktyczniejszą definicję przytacza Cempel [Cempel web].

„Model to uproszczone przedstawienie wybranego fragmentu rzeczywistości celem jego lepszego zrozumienia” [Billinger 98]

Autor ten zwraca uwagę na intencjonalność modeli, czyli preferencje do wyróżniania jednych cech, a zaniedbywania innych. Klasyfikacja modeli świata rzeczywistego została przedstawiona na rys. 4.2. Wyróżniono w niej następujące klasy i podklasy modeli: materialne (fizyczne realizacje), skalowe (w innej skali), analogowe (zasada działania), symboliczne (graficzne, matematyczne, komputerowe), jakościowe (opisowe), opisowe (cel), wyjaśniające (zasadę działania), ilościowe (definiujące właściwości), strukturalne (np.: FEM, BEM, etc.) i funkcjonalne predykcyjne (symptomowo-wskaźnikowe, zmienne stanu).

Modelowanie jako technika abstrakcji, jest jednym z niewielu sposobów (narzędzi) radzenia sobie z kompleksowością [Douglas 99]. Stanowi ono formę abstrakcyjności, umożliwiającą zrozumienie, skomunikowanie, rozwiązanie i udokumentowanie problemów. Model pozwala uchwycić jedynie istotne cechy systemu dla rozważanego kontekstu, ukrywając mniej istotne szczegóły. Ludzie konstruują modele, ponieważ są one dużo łatwiejsze i tańsze do budowy i testowania niż urządzenia i obiekty modelowane. Często modele są jedyną formą wizualizacji tego, czego nie widać.



Rys. 4.2 Diagram klas dla modeli rzeczywistości [jr na podst. Cempel web]

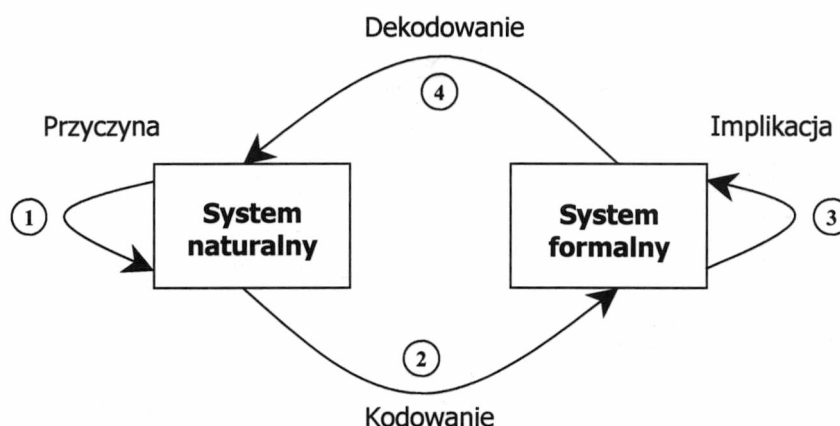
Dzięki modelowaniu osiągane są następujące cele:

- łatwiej wyobrażamy sobie system taki, jaki jest, lub taki, jaki ma być,
- specyfikujemy strukturę i zachowanie systemu,
- otrzymujemy szablony ułatwiające sterowanie działaniami w kolejnych etapach tworzenia systemu,
- dokumentujemy podjęte decyzje.

Modelowanie wymieniane jest jako główne narzędzie badań naukowych, szczególnie w odniesieniu do nauk eksperymentalno-teoretycznych. Wymieniane są tu następujące kroki:

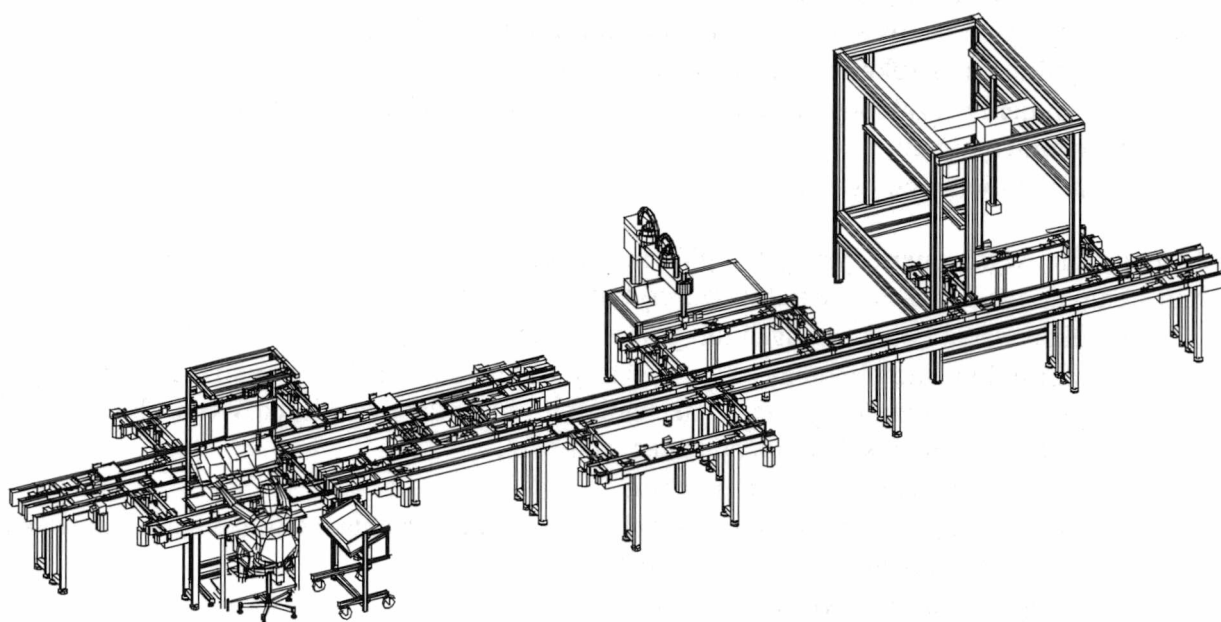
- obserwacja i opis zjawiska lub grupy zjawisk,
- sformułowanie hipotezy wyjaśniającej obserwację, która w naukach fizycznych przyjmuje często postać mechanizmu przyczynowo-skutkowego lub relacji matematycznej, którą można uznać za model,
- przeprowadzenie walidacji modelu,
- użycie zaproponowanego modelu do przewidzenia określonego zjawiska, podobnego zjawiska, obserwacji lub stanu,
- przeprowadzenie weryfikacji hipotezy.

Używanie naszych zmysłów do obserwacji świata dookoła nas i wyciąganie wniosków z informacji sensorycznej jest modelowaniem relacyjnym. Modelowanie relacyjne oznacza obserwowanie, poznawanie, próbowanie zrozumienia systemu naturalnego i zdarzenia, które wywołują obserwowane zmiany. To, co zachodzi w naszych umysłach, to dekodowanie systemu naturalnego poprzez tworzenie lub wybieranie systemu nazywanego systemem formalnym. System formalny może być „manipulowany” na wiele sposobów w celu odwzorowywania zmian systemu naturalnego. Zmiany takie, na systemie formalnym, nazywane są implikacjami. Ostatecznie, gdy uważamy, że mamy odpowiedni system formalny i implikacje wywołujące efekty odpowiadające skutkom zdarzeń w systemie naturalnym to, musimy zdekodować system formalny, w celu oceny efektów zdarzeń. Poniższy diagram (rys.4.3) przedstawia opisaną sytuację.



Rys. 4.3 Model przebiegu badań naukowych [Mikuleck web]

Jeśli wszystkie części diagramu są zharmonizowane, to uważa się, że diagram pasuje i mamy model. Model rzeczywistości jest wynikiem udanego wdrożenia metod naukowych, ale może być również osiągnięty w sposób mniej formalny.



Rys. 4.4 Model elastycznego gniazda montażowego w ITMA PWr (EGM-ITMA) [Reiner, Koch, 98]

Modelowanie jako podstawowe narzędzie poznawcze znalazło zastosowanie w każdej dziedzinie wiedzy. Zależnie jednak od specyfiki obszaru dotyczącego problemu stosowane są dedykowane metody modelowania, języki i narzędzia wspierające. Największą tradycję ma modelowanie skalowe w mechanice, gdzie modele dwu- i trójwymiarowe (2D, 3D) są podstawą projektowania i dokumentowania struktury wyrobów (patrz rys. 4.4). Narzędzia takie jak AutoCAD®, CATIA®, etc. należą do podstawowego „wposażenia kompetencyjnego” inżyniera mechanika.

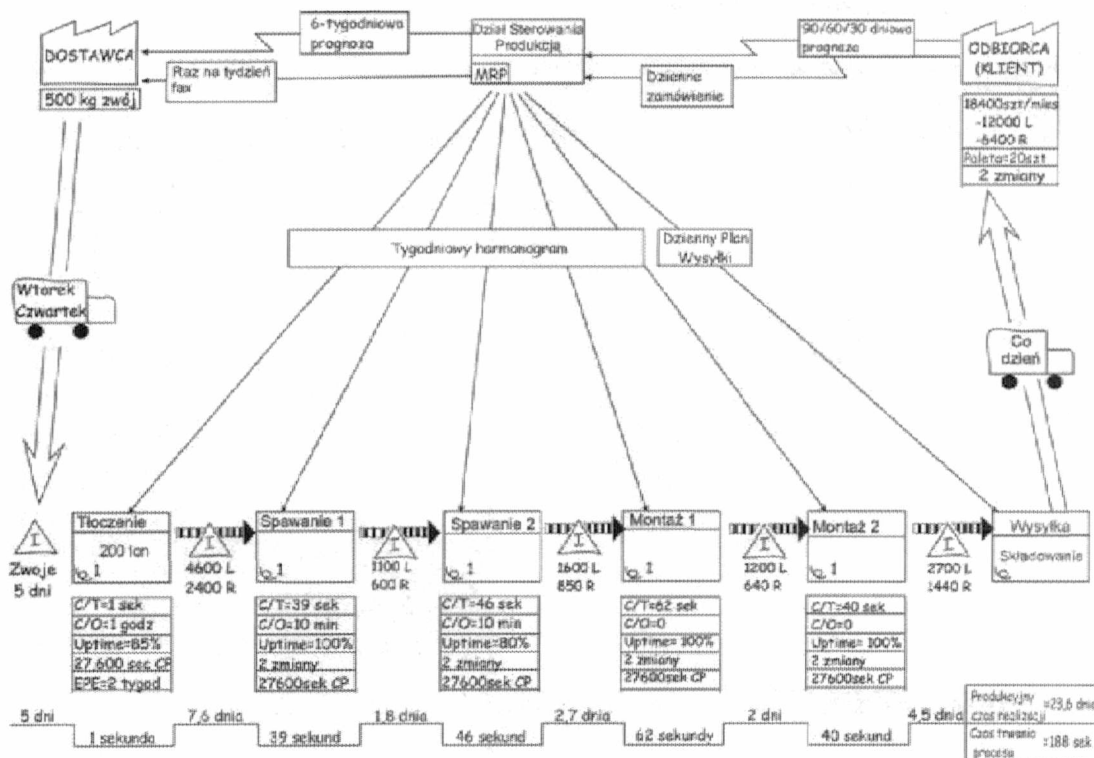
W elektrotechnice i elektronice struktura obwodów elektrycznych (schemat) zapisywane są za pomocą standaryzowanych symboli graficznych. Do najistotniejszych narzędzi wspierających

rysowanie schematów elektrycznych należą OrCAD®, AutoCAD® z biblioteką symboli elektrycznych.

Oprogramowanie rozwija własne metody i języki graficzne dla symbolicznego obrazowania algorytmów (patrz rozdział 1.2).

Modelowanie wykorzystywane jest zarówno do reprezentacji procesów ciągłych jak i dyskretnych. Diagramy takie są symbolicznym uchwyceniem stanu procesu, tj. operacji przetwarzania, oraz związanych z nimi zasobów i materiałów. Przykładem metod modelowania procesów są IDEFx [IDEF web], Diagram Przepływu Procesu, Mapa Strumienia Wartości VSM (ang. *Value Stream Map*) [Rother et al. 99] i inne.

Szczególnie dydaktycznym przykładem modelowania, którego celem jest zrozumienie, tj. uchwycenie rzeczywistości, jest odwzorowywanie strumienia wartości (mapowanie strumienia wartości). Przykład mapy strumienia wartości przedstawia rysunek 4.5. Metoda ta bazuje na wykorzystaniu kilku prostych symboli, zalecając tworzenie mapy za pomocą ołówka i papieru, bezpośrednio dla procesu produkcyjnego. Dzięki „oddaleniu komputera” opracowujący model VSM zdani są wyłącznie na własny intelekt – diagram ten musi być dla wszystkich uczestników grupy zrozumiały.

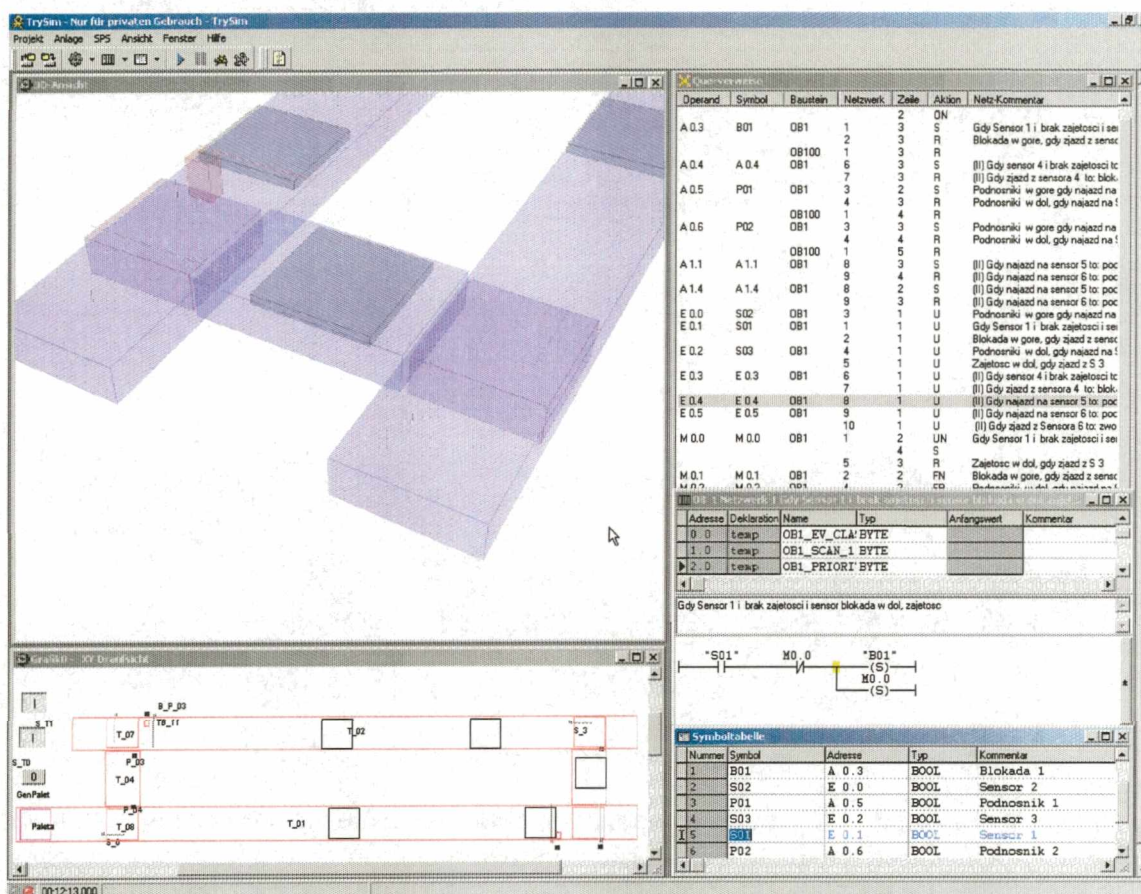


Rys. 4.5 Przykład mapy strumienia wartości VSM [Koch, T. 01]

Kiedy „rozumienie problemu” wykracza poza aspekty statyczne, to konieczny jest model dynamiczny. Szczególnie cennym narzędziem umożliwiającym kontrakcję lub ekspansję czasoprzestrzeni jest symulacja. Jej podstawą jest model symulacyjny oraz identyfikacja zmiennych zależnych. Najbardziej spektakularną formą ich prezentacji jest animacja – podobnie jak w grach komputerowych. Najlepsze praktyki symulacji (ang. *best practices*) ARENA [Arena 00] wskazują jednak, że animacja nie jest jedynie zbędną i kosztowną funkcjonalnością. Wskazują one na to, że animacja może stanowić nawet o sukcesie projektu. Sponsorzy, dla których ma dostarczyć ważnych wskaźników dla podjęcia decyzji, na jej podstawie budują zaufanie do jakości modelu. Jeśli to, co widać odbiega od wyobrażenia, to

wtedy wyniki liczbowe są kwestionowane. Ten sam mechanizm jest pomocny dla opracowywujących model w celu weryfikacji jego poprawności.

Jako przykłady modeli symulacyjnych, wykorzystujących animację 3D, z zakresu mechaniki wymienić należy modele kinematyczne, model rozmieszczenia przestrzennego środków produkcji (ang. *layout*), modele ergonomiczne pracowników, modele przebiegu obróbki CNC i modele przepływu materiału w procesach dyskretnych. W zakresie elektro-elektronicznym modele symulacyjne wykorzystywane są do projektowania obwodów drukowanych i układów wielkiej skali integracji. Graficzna weryfikacja algorytmów sterowania bazuje na animacji diagramów drabinkowych i automatów stanów. Interesującym przykładem symulatora sterownika PLC i wirtualnego obiektu jest oprogramowanie TrySim® (rys. 4.6). Obiekt rzeczywisty jest zamodelowany w przestrzeni 3D z wykorzystaniem elementów konstrukcyjnych i elementów wykonawczych. Elementy wykonawcze mogą być sterowane za pomocą sygnałów elektrycznych generowanych przez wirtualny sterownik PLC. Sterownik ten przetwarza program napisany w jednym z języków zgodnych z IEC61131. Dzięki wirtualnym czujnikom oraz elementom symulującym przepływ materiału, możliwe jest zamknięcie pętli sprzężenia zwrotnego. Poprzez takie podejście i wykorzystanie odpowiednich narzędzi, oprogramowanie sterujące może być testowane na wirtualnym obiekcie w warunkach zmiany stanów.



Jak wynika z powyższych rozważań, modelowanie jest trudnym zadaniem badawczym, często nawet określanym „sztuką”. Mimo to, modelowanie znajduje coraz szersze zastosowanie jako narzędzie inżynierskie.

W celu zapewnienia wysokiej jakości modeli konieczne jest podporządkowanie modelowania poniższym zasadom:

- cel modelowania, jakie modele tworzyć,
- poziom szczegółowości dla modelu,
- relacja z rzeczywistością,
- zakres modelu i jego widoki.

Opierając się na definicji modelowania sformułowanej przez Billingera, że celem modelowania jest „lepsze zrozumienie”, można postawić tezę, że modelowanie symboliczne jest zatem narzędziem zarządzania wiedzą, które obejmuje pełny cykl rozwoju wiedzy tj. jej kreowanie, komunikowanie i archiwizację.

Podczas kreowania powstaje model mentalny, który reprezentuje wymaganą perspektywę rzeczywistości. Podczas komunikowania wiedza jednostki przekazywana jest na zewnątrz (eksternalizacja). Ponieważ najprostszy przekaz modeli następuje w formie graficznej lub matematycznych równań, to dokumentacja jest niejako „produktem ubocznym” komunikacji interpersonalnej. Informacja taka może być przechowywana w formie klasycznej lub elektronicznej i służyć ponownemu generowaniu wiedzy (internalizacja).

4.2 Modelowanie w automatyzacji

4.2.1 Programowanie graficzne w automatyzacji

Co to jest program?

Niestety nawet nowoczesna Encyklopedia Internetowa [Wiem 01] dostarcza bardzo nieprecyzyjnej definicji:

„Oprogramowanie, software, program lub zbiór programów, umożliwiający korzystanie z komputera oraz ewentualnie ze sprzętu peryferyjnego (modemu, skanera, itp.) bądź pozwalający rozwiązywać konkretne problemy użytkownika.”

Oryginalne rozumienie pojęcia „software”, w książce „Doing Hard-Time”, przedstawia Bruce Douglas [Douglas 99]

„...oprogramowanie to model rzeczywistego lub koncepcyjnego świata.”

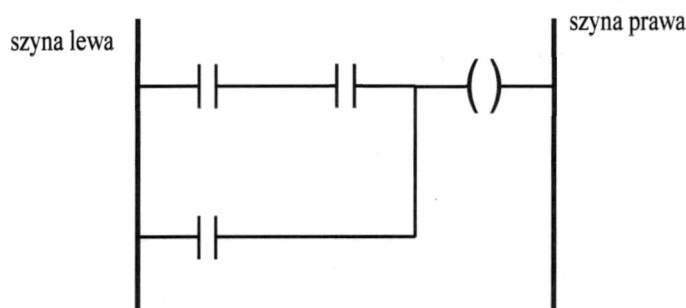
Dalej jednak czytamy, że model rzeczywistości jest jednak zupełnie odmienny od modelu programistycznego. Świat nie składa się z obiektów wysyłających sobie komunikaty. Jeśli jednak przyjmiemy taką abstrakcję, to uzyskany model będzie tworzył uruchamialną symulację, a przejście do systemu programistycznego będzie łatwe. Cook i Daniels [Cook, Daniels 94] podkreślają fundamentalny problem przewidywalności, którego pozbawiony jest świat rzeczywisty. Niezależnie od nas, świat nie jest przewidywalny, a oprogramowanie – tak, gdyż jego zachowanie determinowane jest poprzez kod. Dlatego też oprogramowanie nie może w pełni modelować świata rzeczywistego, a do rozwiązania problemu korzysta się z uproszczenia - częściowej przewidywalności. Kolejny problem to współbieżność towarzysząca otoczeniu. Ponieważ nasz tok myślenia jest jednowątkowy, to mamy trudności z analizą modeli współbieżnych.

Programowanie, czyli implementacja algorytmu, lub budowanie modelu symulacyjnego jest realizowane za pomocą języków programowania. Pierwsza generacja języków

programowania to programowanie w kodzie maszynowym, czyli poprzez kody instrukcji procesora. Druga generacja, to assembly, czyli wykorzystanie symbolicznych nazw instrukcji procesora. Trzecia generacja to języki imperatywne (proceduralne), aplikatywne (funkcjonalne), deklaratywne (regulowe), symboliczne i obiektowe. Języki czwartej generacji, to generatory aplikacji obejmujące fizyczne słowniki danych, formatery ekranu, generatory raportów, języki zapytań, specyfikatory dialogów, generatory kodu i języki wysokiego poziomu. Języki te określane są również jako CASE (ang. *Computer Aided Software Engineering*) – komputerowo wspomagana inżynieria oprogramowania.

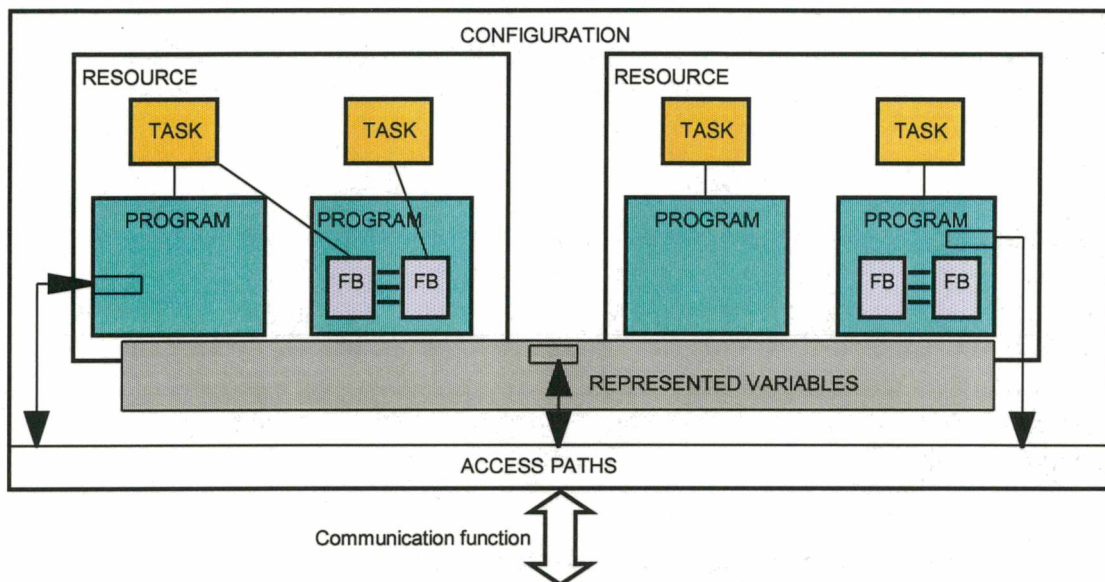
Programowanie graficzne, za pomocą symboli, ma szczególne znaczenie podczas implementacji oprogramowania dla sterowników swobodnie programowalnych PLC (patrz rozdział 1.2).

Najwcześniej stosowany w programowaniu sterowników PLC był język drabinkowy LD, oparty na schemacie elektrycznym logicznego układu przekaźnikowego. Symbole graficzne reprezentujące styki i cewki umieszcza się w obwodach w sposób podobny do szczebli, między szynami prądowymi - lewą wejściową a prawą wyjściową. Obwód jest definiowany jako zbiór wzajemnie połączonych elementów graficznych – patrzy rys. 4.7. Przetwarzanie programu PLC następuje w sposób sekwencyjny i cykliczny. Dodatkowo z obwodem może być skojarzona etykieta mająca postać nazwy lub liczby, co poprawia czytelność schematu. Ograniczenia języka LD oraz jego późna standaryzacja (dopiero w 1993r.) to główne przyczyny powstania wielu firmowych rozwiązań bazujących na idei LD [Kosmol et al. 00].



Rys. 4.7 Przykład programu PLC w języku LD

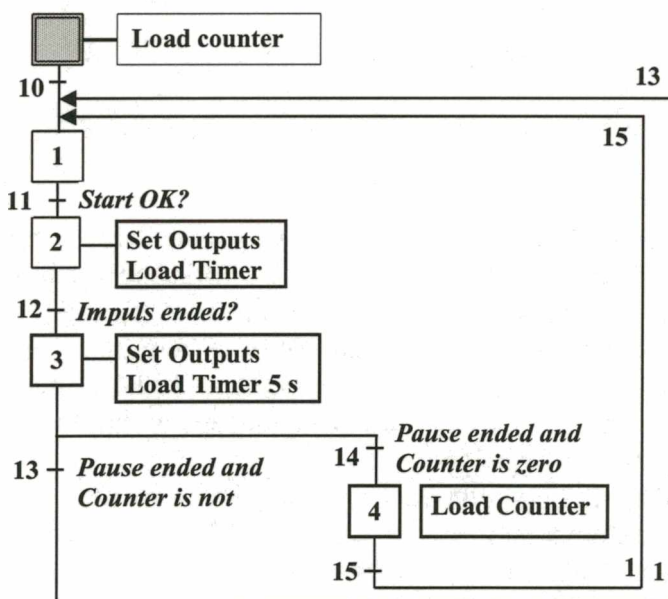
Natomiast język schematu bloków funkcyjnych FBD (ang. *Function Block Diagram*) opiera się na schemacie blokowym reprezentującym przepływ danych (rys. 4.8.). Poszczególne bloki funkcyjne stanowią tzw. „czarne skrzynki” (ang. *black box*) ukrywające sposób realizacji algorytmu. Dostęp do funkcjonalności bloku ograniczony jest przez ściśle zdefiniowany interfejs. W ten sposób chroniona jest funkcjonalność przed nieprawidłowym użyciem, a modyfikacje wymagają utworzenia nowego bloku, co poprawia stabilność układu. „Enkapsulacja” bloków funkcyjnych jest wykorzystywana również jako mechanizm ochrony wiedzy i doświadczenia (ang. *know-how*) producentów. Modelowanie za pomocą bloków funkcyjnych pozwala ustrukturyzować zadanie oraz pokazać wzajemne relacje między poszczególnymi składowymi.



Rys. 4.8 Bloki funkcyjne (FB) w modelu konfiguracyjnym IEC 61131-3 [PLCOpen]

Oprócz języka FBD, zdefiniowanego w normie IEC61131-3, stosowane są również firmowe „mutacje” takie jak CFC (ang. *Continuous Function Chart*) firmy SIEMENS [Siemens 96].

Język graficznego programowania sterowników PLC w postaci grafu sekwencji SFC (ang. *Continuous Function Chart*) bazuje na sieciach Petriego. Dzięki takiej podstawie możliwe jest stosowanie metod matematycznych do weryfikacji modelu w zakresie „zakleszczeń” (ang. *deadlock*), „zagłódzeń”. Składnia diagramu SFC obejmuje akcje i tranzycje (rys. 4.9). Akcje określają zadania wykonywane przy wejściu i wyjściu ze stanu oraz powtarzane cyklicznie. Przejście do następnego stanu następuje na podstawie oceny warunku opisującego tranzycję. Semantyka SFC pozwala również na zapis ciągów sekwencji współbieżnych oraz skoki. Do najbardziej rozpowszechnionych implementacji SFC należy IsaGraf (firmy JC International); S7-Graph (firmy SIEMENS).



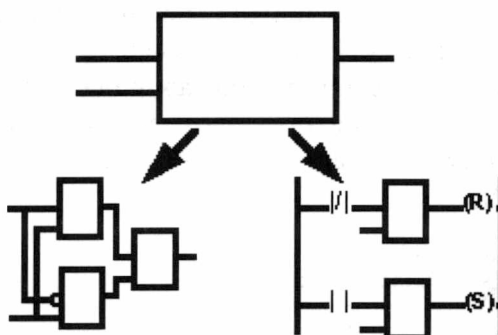
Rys. 4.9. Przykład grafu sekwencji SFC [Kosmol et al. 00]

Alternatywną propozycją, dla modelowania opartego na sieciach Petriego, jest metoda Grafpol opracowana przez Mikulczyńskiego i Samsonowicza [Mikulczyński, Samsonowicz 97]. Metoda ta nadaje się do graficzno - analitycznego modelowania matematycznego, zarówno algorytmów dyskretnych procesów produkcyjnych, jak również algorytmów ich sterowania.

W przypadku tworzenia oprogramowania dla sterowników PLC, stosowano głównie zasadę dekompozycji problemu dla wyodrębniania lokalnych zadań sterowniczych. Wraz ze wzrostem wymagań wobec zautomatyzowanych systemów produkcyjnych (rozdział 1.1) oraz wprowadzaniem rozproszonych struktur sterowania (rozdział 1.2), podejście powyższe do programowania, tj. zgodne z normą IEC61131 nie przestało być efektywne. Automatyzacja wytwarzania realizowana w strukturach rozproszonych, wymaga jednak globalnego spojrzenia na problem.

4.2.2 Bloki Funkcyjne w automatyzacji wytwarzania

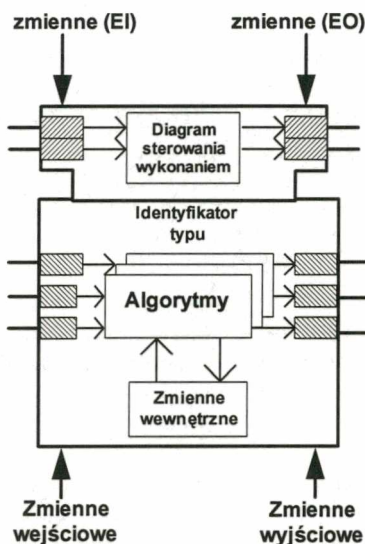
Wprowadzenie bloków funkcyjnych do programowania sterowników PLC/CNC, zdefiniowanych w IEC61131 było odpowiedzią na potrzebę strukturyzacji programu. Efektywna analiza coraz obszerniejszych programów wymaga zastosowania różnych poziomów szczegółowości, np.: analiza od ogółu do szczegółu (ang. *top-down*). W tym celu całościowy fragment algorytmu został wydzielony tworząc blok. Jego implementacja może być realizowana w dowolnym języku programowania, tekstowym lub graficznym. Taki blok funkcyjny może być także ponownie wykorzystany w innym fragmencie programu.



Rys. 4.10 Blok Funkcyjny (FB) i jego rozumienie zgodnie z IEC61131

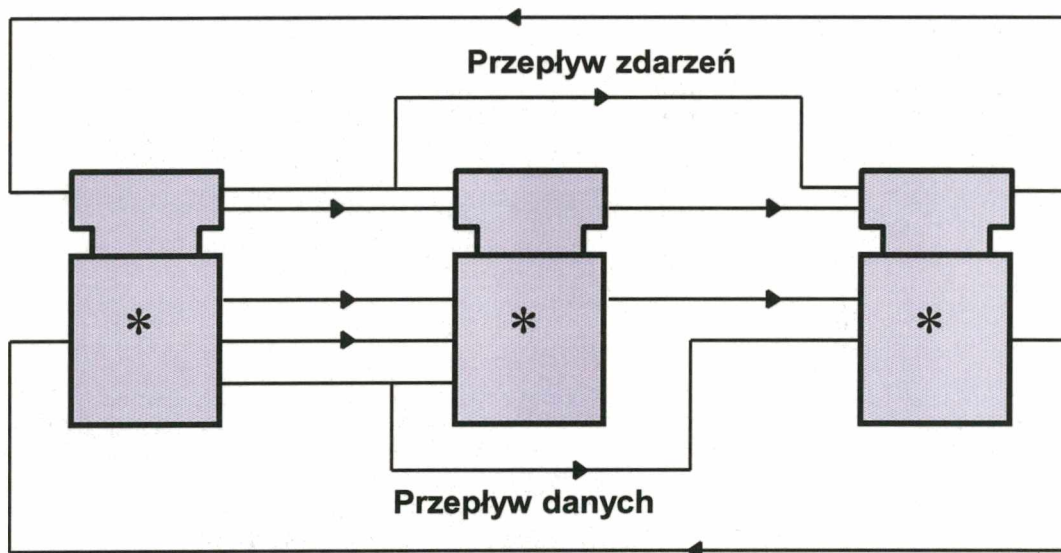
Jednak dopiero bloki funkcyjne zdefiniowane w późniejszej normie IEC61499 pozwalają sterować ich działaniem (rys. 4.10). Norma IEC61499 jest odpowiedzią na wymagania standardu dla bloków funkcyjnych w rozproszonych systemach pomiarowych procesów przemysłowych oraz dla systemów sterowania IPMCS (ang. *Industrial Process Measurement and Control Systems*). Zaproponowane rozwiązanie wykorzystuje komunikację *fieldbus* zgodną z IEC61158 oraz IEC61158 dla rozproszonych aplikacji, jak również standard IEC61131 definiujący bloki funkcyjne.

Bloki funkcyjne z IEC61131-3, wykorzystywane jako elementy programów w sterownikach scentralizowanych, są elementami rozproszonych aplikacji w zdecentralizowanych systemach sterowania IEC61804. Dla zastosowania bloków funkcyjnych w IPMCS przyjęto wymagania dotyczące planowania sterowania, komunikacji i interakcji, czasu rzeczywistego, wyboru alternatywnych algorytmów, działania zgodnego z automatem stanów oraz elastyczności.



Rys. 4.11 Podstawowy typ Bloku Funkcyjnego (FB) zgodnie z IEC61499

Norma IEC61499 obejmuje rozproszone aplikacje, interfejs zdarzeń i danych oraz zdarzeniowy automat stanów. Wspecyfikowano w nim komunikację między urządzeniami poprzez jedno lub wiele połączeń komunikacyjnych (rys.4.11). Urządzenia te mogą się ponadto komunikować ze sterowaną maszyną lub procesem. Aplikacja może być umieszczona w jednym lub wielu urządzeniach. Aplikacja składa się z jednego lub wielu instancji bloków funkcyjnych połączonych przez łączniki zdarzeń i danych. Dla zgodności z modelem IEC61131-3, wprowadzono pojęcie urządzeń wirtualnych występujących w obrębie urządzeń fizycznych. Urządzenie może być dzielone na zasoby (ang. *resources*), które zawierają bloki funkcyjne przetwarzania oraz bloki funkcyjne interfejsów usług dla zapewnienia obsługi portów wejść i wyjść, komunikacji sieciowej oraz realizacji usług systemu operacyjnego.

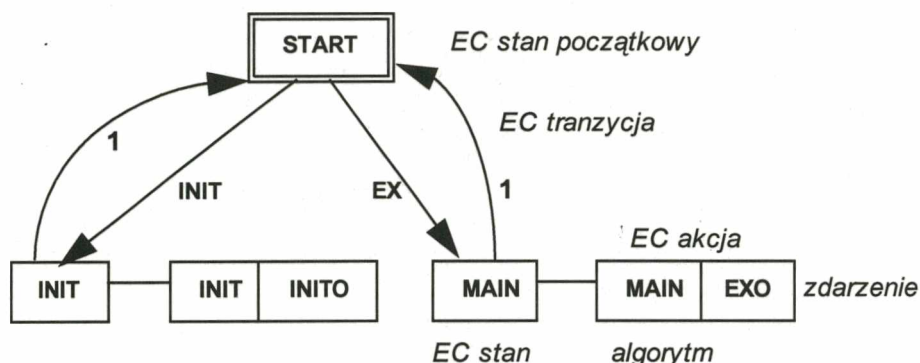


Rys. 4.12 Rozproszona aplikacja na bazie bloków funkcyjnych

Zdarzeniowy automat stanów (rys.4.13) jest podstawą deklaracji bazowego bloku funkcyjnego. W tym celu należy:

- zdefiniować zewnętrzne interfejsy,
- schemat sterowania wykonaniem algorytmu ECC (ang. *Execution Control Chart*) oraz

- algorytmy związane z ECC.

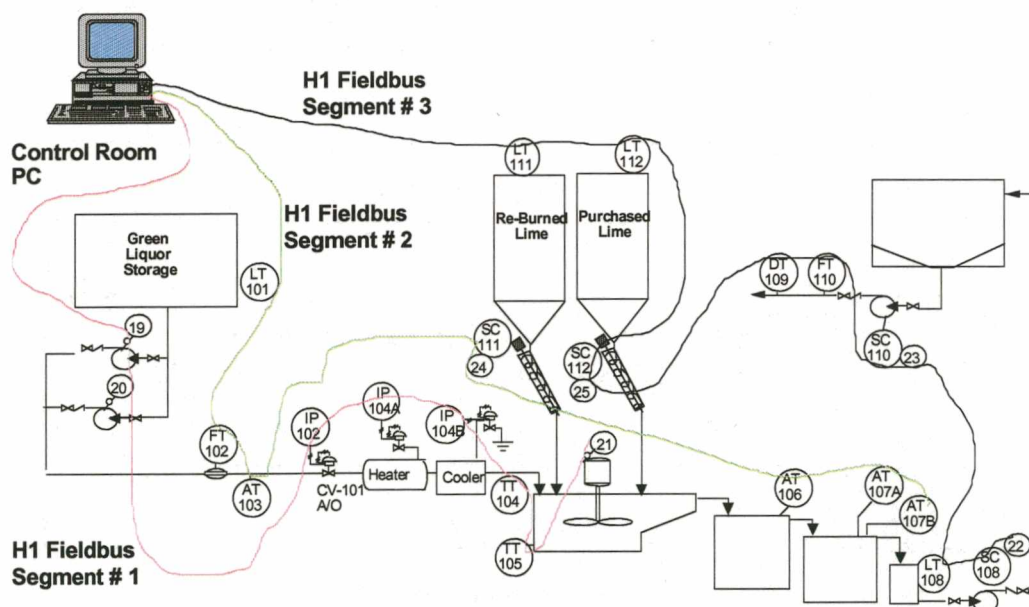


Rys. 4.13 Diagram sterowania przetwarzaniem ECC

Model ECC może być przedstawiony w postaci Sekwencyjnego Diagramu Stanów, którego tranzycje są zmiennymi binarnymi lub stanami wejść zdarzeń. Algorytm przetwarzania może być wyrażony w językach IEC61131-3 lub innym języku programowania.

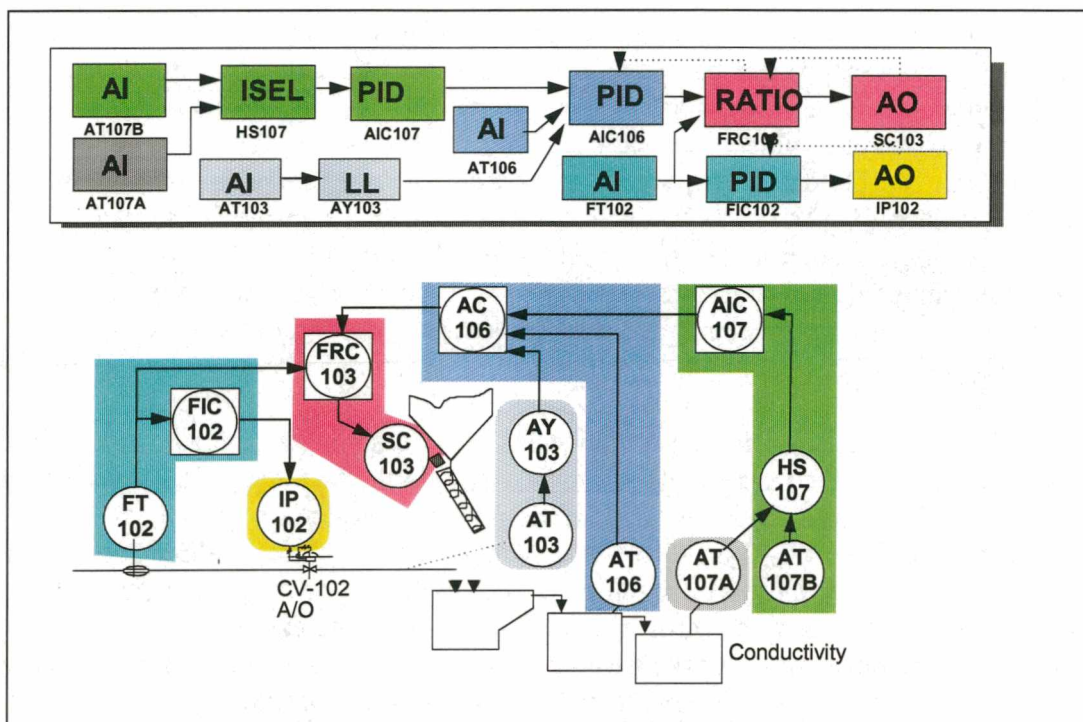
W celu opisu systemu sterowania procesami ciągłymi, standard IEC61804 zaleca dwa widoki, fizyczny i logiczny.

Widok fizyczny (ang. *Physical View*) identyfikuje czujniki i elementy wykonawcze w relacji z obiektami procesu technologicznego oraz urządzenia sterujące. Na diagramie pokazywane są również relacje fizyczne między obiektami. Każdy obiekt opatrzony jest także etykietą identyfikacyjną (rys. 4.14).



Rys. 4.14. Widok fizyczny instalacji rozproszonego IPMCS

Natomiast widok logiczny (ang. *Logical View*) (rys. 4.15), przedstawia strukturę sterowania w ujęciu bloków funkcyjnych, takich jak: analogowe wejścia (AI), regulatory (PID), analogowe wyjścia (AO), etc.



Rys. 4.15 Widok logiczny rozproszonego IPMCS

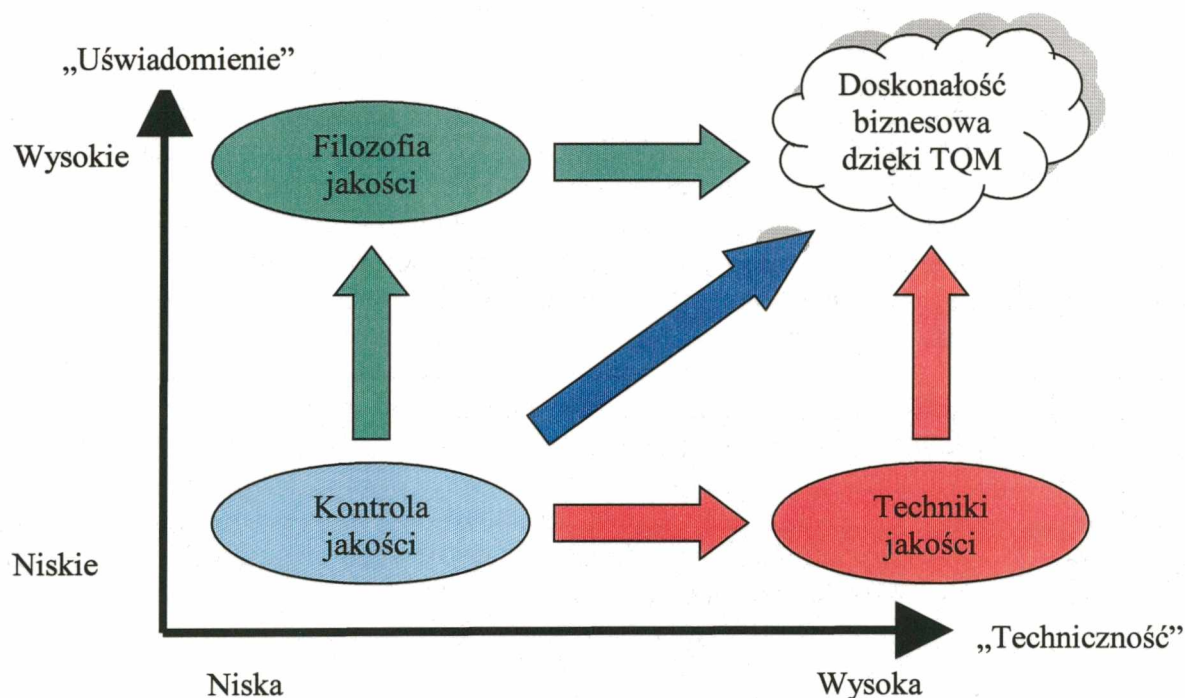
Jak wynika z powyższej charakterystyki bloków funkcyjnych, zdefiniowanych w normach IEC61131 i IEC61499 oraz widoków fizycznego i logicznego z normy IEC61804, rozwiązania te adresowane są dla dostawców rozwiązań automatyzacji. Przedstawione rozwiązania są wynikiem rozwoju stosowanych języków programowania i paradygmatu opracowywania układów sterowania. Paradygmat programowania ewoluował wprawdzie w kierunku graficznego, czerpiąc wzorce z paradygmatu obiektowo - zorientowanego, ale jego filozofia tkwi nadal w elektrotechnice. Bloki funkcyjne, których realizacja jest w pełni programistyczna, przedstawiane są jako wirtualne układy scalone. Układy te mają „nóżki”, tj. wejścia/wyjścia, które wymagają połączenia. Paradygmat elektrotechniczny jest łatwy do przyjęcia i stosowania przez doświadczonych programistów, zaś znacznie trudniejszy dla klienta zlecającego zadanie automatyzacji. Paradygmat ten jest efektywny w świecie rozwiązań, mimo że ogranicza współczesne możliwości wykorzystania oprogramowania, gdyż abstrakcja elektrotechniczna nie wspiera efektywnie biznesowej i analitycznej perspektywy systemu poddawanej automatyzacji.

4.2.3 Paradygmat automatyzacji

Paradygmat to sposób myślenia. Natomiast paradygmat programowania to ogólne podejście do programowania lub rozwiązywania problemów za pomocą języków programowania. Język programowania jest językiem formalnego zapisu algorytmu. Zapis ten pozwala na wykonanie tego algorytmu przez komputer, jest też czytelny dla człowieka. Ten sam algorytm można zaimplementować stosując różne paradygmaty oraz różne języki programowania. Ten sam język programowania może być wykorzystywany, choć z różną efektywnością, do implementowania w ramach różnych paradygmatów. Do najistotniejszych paradygmatów należy zaliczyć następujące: proceduralny, obiektowo-zorientowany, funkcyjny, logiczny, bazujący na regułach, zapytań do baz danych, wizualny, skryptowy, symulacji i arkusza kalkulacyjnego.

Paradygmat, czyli sposób myślenia, jest zatem pewną abstrakcją lub filozofią wykorzystania umiejętności (kompetencji) lub narzędzi. Abstrakcja jest podstawowym narzędziem radzenia sobie z kompleksowością. Używając jej identyfikujemy kluczowe aspekty składowych, przy pominięciu pozostałych mniej istotnych, dzięki czemu łatwiej koncentrujemy uwagę na rozwiązywanym problemie. Korzystanie z abstrakcji jest niezbędne dla rozwiązywania problemów występujących w kompleksowych systemach socjo-technicznych. Abstrakcja funkcjonuje w języku automatyzacji w postaci urządzeń wirtualnych (ang. *virtual devices*), bloków funkcyjnych (ang. *functions block*) oraz etykiet zmiennych (ang. *tag*). Pojęcia te pozwalają opisać wybrany element systemu automatyzacji bez wnikania w szczegóły implementacyjne.

Coraz wyraźniej wskazuje się, że współczesne rozwiązania techniczne nie są należycie wykorzystywane. Jako tego przyczynę identyfikuje się brak równowagi między świadomością a kompetencjami. Deming wskazuje, że doskonałość biznesowa wymaga zrównoważonego rozwoju techniki i filozofii – rys. 4.16. W przypadku automatyzacji należy zestawić z jednej strony tempo rozwoju mocy obliczeniowej komputerów, wzrost przepustowości sieci, przyrost objętości informacji, a z drugiej zdolności kognitywne człowieka i jego „uzbrojenie” metodyczne i organizacyjne. Poszukiwanie miar ilościowych dla takich zestawień, jak wykazano w Rozdziale 3, nie przynosi jednak oczekiwanych wyników.



Rys 4.16 Doskonałość w biznesie powstaje poprzez technikę i mentalność [Kamise web]

Programowanie (modelowanie) w automatyzacji opiera się ciągle na paradygmacie elektro-technicznym [Reiner, Koch, 00]. Wprawdzie koncepcja bloków funkcyjnych jest zbliżona do obiektów to, jednak ograniczenia wynikające z myślenia „elektrotechnicznego” nie pozwalają efektywnie implementować algorytmów rozproszonego sterowania dla obsługi zdarzeń asynchronicznych.

4.3 Obiektowość i język UML

4.3.1 Paradygmat obiektowy

Podejście obiektowo-zorientowane opiera się na strukturze i jest orientowane na problem, a nie na jego funkcje i konkretne rozwiązanie. Autorzy metodyki Syntropy, Cook i Daniels [Cook, Daniels 94] podają analogię paradygmatu obiektowego do mapy.

Jeśli chcesz znaleźć drogę ze swojego domu do biura, masz dwie możliwości: pytasz o drogę albo kupujesz mapę. Kierunki wyrażamy funkcjonalnie: idź na zachód 2km, skręć w prawo, po trzeciej przecznicy skręć w prawo za budynkiem poczty. Jeśli podana marszruta była poprawna i zostanie przez nas dokładnie wypełniona, natychmiast rozwiążemy nasz problem – dotrzemy do celu. Rozwiązanie takie nie jest jednak ogólne i nie da się go ponownie zastosować, a gdy stracimy punkt odniesienia, cel zostanie prawdopodobnie utracony.

Mapa jest modelem bazującym na elementach świata rzeczywistego. Jest ona bogata w informację kontekstową, nie tworzą jej punktowe znaczniki, lecz struktura. Doświadczenie pokazuje, że podążanie do celu, gdy znana jest mapa jest dużo łatwiejsze. Bogactwo modelu mapy pozwala zawrzeć w niej więcej informacji modelujących świat rzeczywisty. Mapa reprezentuje strukturę tj. elementy i relacje, dlatego nawigacja w takim środowisku jest oczywiście funkcjonalna.

Paradygmat obiektowy postrzega świat w sposób abstrakcyjny, nazywając otaczające elementy obiektami. Obiekty te mają cechy i umiejętności wewnętrznego działania i komunikacji z otoczeniem. Obiektowość jednak, nie koncentruje się na realizacji danego bytu, lecz poszukuje uogólnień, szablonów, wzorców, przedstawiając je jako klasy. Za pomocą relacji dziedziczenia, udostępnia ona mechanizm wspierający rozbudowę istniejących już systemów i ponowne wykorzystanie opracowanych wcześniej komponentów.

Pojęcie obiektowości kojarzone jest jednak ciągle z językami programowania obiektowego lub „nieużyteczną abstrakcją”. Programowanie obiektowe nie znajduje ciągle zastosowania przy programowaniu sterowników, ponieważ sposób ten zależny jest w znaczącym stopniu od producentów układów sterowania PLC. Brak języków obiektowych w normie IEC61131 nie zachęca producentów sterowników do wzbogacania środowiska programistycznego. Programowanie jest jednak językiem implementacji. Kompleksowe systemy sterowania wymagają projektów, a te języków specyfikacji rozwiązań na poziomie logicznym i koncepcyjnym, a sam poziom implementacji jest już wspierany w dostatecznym stopniu.

Filozoficzna misja obiektowości podkreśla uzyskanie jak najmniejszej luki pomiędzy myśleniem o rzeczywistości a myśleniem o danych i procesach na nich zachodzących. W modelu relacyjnym, model pojęciowy stara się odwzorowywać świat rzeczywisty, lecz jest ograniczony dostępną bazą implementacyjną. W rezultacie, schemat struktury gubi semantykę danych. Model obiektowy podtrzymuje te zgodności, przybliżając semantykę danych do świata rzeczywistego [Subieta 98].

Podstawowe zasady obiektowości obejmują: obiekty, hermetyzację, typy i klasy, dziedziczenie, komunikaty, polimorfizm.

Obiekt jest abstrakcyjnym pojęciem integrującym dane oraz zachowanie. Może on reprezentować elementy świata rzeczywistego, takie jak np. obrabiarka, robot, itp. lub abstrakcyjne formy koncepcyjne, takie jak np. zlecenie produkcyjne, plan obróbki. Każdy z takich elementów charakteryzuje wiele różnych właściwości. Są to: atrybuty (struktury

danych), zachowanie (operacje i metody), stan (pamięć), tożsamość i odpowiedzialność. Obiekt posiada swoją tożsamość, która jest niezależna od stanu atrybutów oraz lokalizacji, co wyróżnia go spośród innych obiektów. Obiekt ma określone zachowanie, tj. zestaw operacji które wolno do niego stosować. Implementacje operacji zwane są metodami.

Smith [Smith 99] personifikując obiekty, podkreśla cztery ich cechy:

- Prostota obiektów stanowi o ich pięknie i sile – mogą one zostać łatwo powielone dla rozszerzenia funkcjonalności, wymienione dla zwiększenia wydajności czy zmiany zachowania lub usunięte, gdy ich funkcjonalność nie jest już potrzebna – wszystko bez wpływu na całościową funkcjonalność systemu.
- Obiekty są tolerancyjne - nie zależą od wyglądu, kompleksowości, czy otoczenia. Np.: sterownik PLC może być traktowany jako obiekt i dzięki temu podejście to może być także stosowane dla systemów tradycyjnych, stopniowo je zastępując i uzupełniając.
- Obiekty są wiarygodne: wierzę, że obiekt poprzedzający i następny będą wykonywały swoje zadania. (Podejście tradycyjne stawia pytanie: czy my możemy otoczeniu rzeczywiście zaufać?) Przełamanie tej bariery jest istotnym czynnikiem urzeczywistnienia koncepcji obiektowej.
- Obiekty są szeroko rozpraszalne - nie muszą działać w obrębie tej samej platformy, a otaczające obiekty mogą pojawiać się od czasu do czasu i znikać (dynamicznie). Zachowują się one jednakowo zarówno wówczas, gdy działają samodzielnie jak gdy współpracują z innymi.

Podstawowym efektem abstrakcji jest wyraźny spadek skali problemu. Dla redukcji możliwych interakcji między obiektami oraz ukrycia ich wewnętrznej architektury wprowadza się hermetyzację.

hermetyzacja (ang. encapsulation) - zamknięcie pewnego zestawu bytów programistycznych w „kapsułę” o dobrze określonych granicach; oddzielenie abstrakcyjnej specyfikacji tej kapsuły (obiektu, klasy, modułu, etc.) od jej implementacji; ukrycie części informacji zawartej w tej kapsule dla operacji z zewnątrz obiektu. [Subieta 99]

Hermetyzacja jest realizowana poprzez specyfikację interfejsu do obiektu, który definiuje dostępne usługi i dane. Koncepcje ortodoksyjnej i ortogonalnej hermetyzacji w różny sposób traktują możliwość dostępu od wnętrza obiektu. Z jednej strony rozważają one elastyczność programistyczną, a z drugiej bezpieczeństwo.

dziedziczenie (ang. inheritance) - związek pomiędzy klasami obiektów określający przekazywanie cech (definicji atrybutów, metod, itd.) z nadklas do jej podklas. ... jest podstawowym mechanizmem wspierającym ponowne użycie. [Subieta 99]

Przykładowo obiekt klasy „tokarka” może dziedziczyć właściwości (definicje atrybutów i metody) określone w ramach klasy nadrzędnej „obrabiaarka”.

polimorfizm (ang. polymorphism) - termin używany w dwóch nieco różnych znaczeniach: (1) w terminologii obiektowej: możliwość istnienia wielu metod o tej samej nazwie, powiązana z możliwością wyboru konkretnej metody podczas wykonywania; (2) w terminologii teorii typów i języków polimorficznych: (...) umożliwienie

definiowania funkcji lub procedur, których argumenty i wynik mogą posiadać jednocześnie wiele typów... [Subieta 99]

Metodyki obiektowo – zorientowane

Henderson-Shell specyfikuje wymagania odnośnie dojrzałej metodyki dla projektu obiektowo-zorientowanego:

- pełny cykl życia procesu dla zastosowań biznesowych i technologicznych,
- kompletny zbiór koncepcji i modeli spójnych wewnętrznie,
- szczegółowy opis wszystkich wytworów (ang. *deliverables*),
- dającą się użyć notację,
- wsparcie narzędzi do diagramowania od obcych producentów,
- sprawdzony i przetestowany zestaw technik,
- zestaw odpowiednich metryk,
- standardy i strategie testowania,
- identyfikacje zadań organizacyjnych, np.: analiza, programowanie,
- zalecenia dotyczące kierowania projektem i zapewnienia jakości,
- wskazania dotyczące zarządzania bibliotekami i ponowne użycie.

Największy portal internetowy dotyczący obiektowości Cetus Links – Object Orientation [Cetus web] zawiera blisko 20 tysięcy odnośników do informacji w Internecie. W rozdziale poświęconym Metodykom Obiektowym prezentuje on aż 51 różnych pozycji:

BON, Booch, BOOM, Catalysis, CBD/e, Coad/Yourdon, COMMA, CRC, Convergent Engineering, Demeter, DOORS, DOOS, EPA, EROOS, Fusion, Goofee, HOOD, IDEA, ION, KISS, MERODE, MOSES, MWOOD, Object COMX, Objecteering, Objectory, OEP, Octopus, OMT, OOAD/OOIE, OOA/RD, OOBE, OOCL, OOHDM, OOram, OOSC, OOSD, OOSE, OOSP, Open, OSA, PAUD, ROAD, ROPES, RUP, Scrum, Skill-Driven Design, SDL, Shlaer & Mellor, Softstar, SOMA, SOMT, Syntropy, XP.

Większość z powyższych propozycji stanowią firmowe rozwiązania, które nie znalazły szerszego zastosowania głównie ze względu na ograniczony zakres stosowalności i brak dokumentów standaryzacji. Z tego względu brak również narzędzi (programów) wspomagających ich zastosowanie.

Paradygmat obiektowy znajduje ciągle nowe interpretacje i prowadzona jest nieustanna polemika nad jego ograniczeniami. Jako podstawowe z nich wymienia się:

- Klasy/obiekty implementowane w jednym języku nie mogą współdziałać z innymi implementowanymi w innych językach;
- Niektóre języki programowania wymagają nawet zgodności wersji kompilatora;
- Kompozycja obiektów zachodzi na poziomie języka;
- Brak wsparcia kompozycji – brak narzędzi wizualnego/interaktywnego łączenia obiektów [Pree 97].

Powyższa krótka charakterystyka obiektowości wskazuje, że jest to pewien sposób postrzegania rzeczywistości. Efektem takiego postrzegania jest powstawanie modeli mentalnych. Dla ich wyrażenia, skomunikowania z innymi i udokumentowania potrzebne są języki. Paradygmat obiektowy jest modelowaniem abstrakcyjnym. Dlatego też jego

wyrażanie jest najefektywniejsze poprzez szkice lub diagramy. Językiem wyrażania obiektowości, zyskującym rosnącą akceptację i zastosowanie jest Zunifikowany Język Modelowania (UML).

4.3.2 Standard UML

Zunifikowany język modelowania UML (ang. *Unified Modeling Language*) [UML 99] jest narzędziem do specyfikacji, wizualizacji, konstrukcji i dokumentacji elementów systemów programistycznych, systemów biznesowych jak również innych nie-programistycznych systemów. Jego opracowanie zostało zainspirowane jako odpowiedź na zapytanie ofertowe Object Management Group na standard metodyki obiektowej. Język UML został opracowany jako porozumienie trzech przedstawicieli najistotniejszych metodyk obiektowych: OMT, OOSE i Booch oraz wielu firm programistycznych m.in.: Digital, Hewlett-Packard, I-Logix, ICON Computing, MCI, Microsoft, Oracle, Systemhouse, Texas Instruments i Unisys, i najważniejszych twórców technik modelowania: Davida Harela, Petera Coad i Jima Odell. Język UML jest znacznie bardziej kompletny, niż pozostałe metodyki obiektowe. Dzięki temu umożliwia on modelowanie kompleksowych systemów. Może on być również używany do modelowania systemów czasu rzeczywistego oraz *embedded*. Język UML nie jest jednak metodyką.

UML dla prezentacji modeli definiuje następujące diagramy:

- diagram przypadków użycia (ang. *use case*),
- diagram klas (ang. *class*),
- diagram obiektów (ang. *objects*),
- diagramy zachowania (ang. *behavior*):
 - diagram zmiany stanów (ang. *statechart*),
 - diagram aktywności (ang. *activity*),
- diagramy interakcji (ang. *interaction*):
 - diagram sekwencji (ang. *sequence*),
 - diagram współpracy (ang. *collaboration*),
- diagramy implementacji (ang. *implementation*):
 - diagram komponentów (ang. *component*),
 - diagram rozmieszczenia (ang. *deployment*).

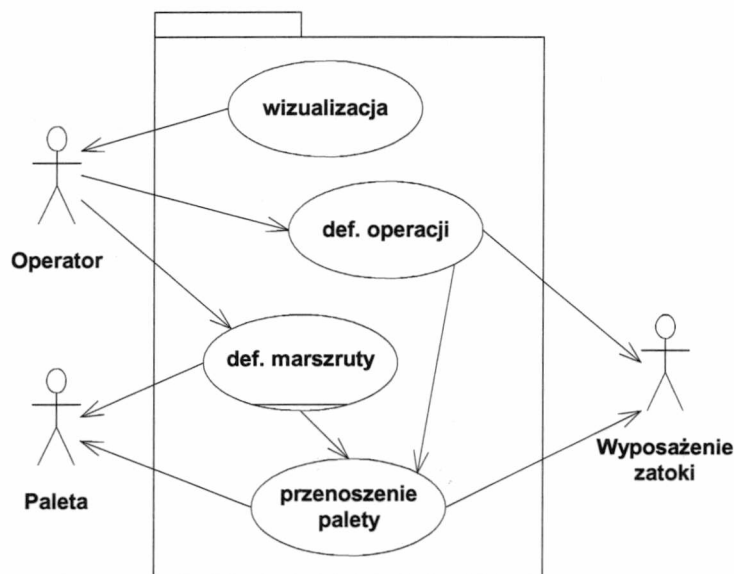
Język specyfikacji UML stał się w krótkim czasie przemysłowym standardem de facto. Aktualnie dostępna jest jego specyfikacja 1.5, a prowadzone obecnie intensywne prace mają doprowadzić do jego międzynarodowej standaryzacji.

Przypadki użycia i diagram przypadków użycia (use-case)

Przypadki użycia są opisem potencjalnych sytuacji, które powinny być wspierane przez system, bez odkrywania jego struktury wewnętrznej. Bardziej precyzyjnie możemy nazwać je funkcjami zwracającymi obserwowalne wartości dla aktorów (obiektów poza kontekstem). Np.: dla zatoki elastycznego gniazda montażowego ITMA mogłyby to być:

- wizualizacja,
- definicja operacji,
- definicja marszruty,
- przenoszenie palety.

Podstawowy celem przypadków użycia i scenariuszy przypadków użycia jest definicja i zrozumienie wymagań aplikacji i użytkownika [USM web].



Rys. 4.17 Przykład diagramu przypadków użycia dla zatoki EGM-ITMA [jr]

Przypadki użycia są dobrą techniką specyfikacji wymagań, jednak ich szersze zrozumienie wymaga określenia kontekstu ich występowania.

Diagram przypadków użycia jest bardzo prosty, składa się z aktorów (osób, urządzeń, organizacji lub zewnętrznych systemów, które są w interakcji z systemem modelowanym) oraz przypadków użycia, które są rozpatrywane (rys. 4.17). Aktorzy prezentowani są jako „zapałkowe ludki”, przypadki użycia jako owale, a sam system jako prostokąt. Odcinki ze strzałkami pokazują, który z aktorów bierze udział, w którym przypadku oraz dominujący kierunek komunikacji.

Diagram przypadków użycia jest zatem raczej dekompozycją funkcjonalną systemu niż dekompozycją obiektową. Przypadki użycia nie są klasami i dlatego też takie ich utożsamianie prowadzi do nieprecyzyjnego zobrazowania funkcjonalności systemu oraz niedokładnego modelu obiektowego.

Oprócz relacji między aktorami szczególnie ważne są związki między przypadkami użycia, wśród których zdefiniowano zawieranie, uogólnienie i rozszerzenie.

- <<extended>> oznacza, że jeden z przypadków użycia jest rozszerzoną formą drugiego,
- <<uses>> oznacza, że jeden z przypadków użycia wykorzystuje usługi oferowane przez inny.

Do relacji zawierania (ang. *uses*) dochodzi wtedy, gdy kilka przypadków użycia wykorzystuje sekwencje podobnych kroków. Dzięki zawieraniu nie ma potrzeby powtarzania jednakowych elementów, a nawet ich kopiowania, lecz tylko wskazanie na zdefiniowany wcześniej element.

Diagram przypadków użycia został użyty we wcześniejszym rozdziale do zamodelowania celów pracy doktorskiej (rys. 2.1) oraz zamodelowania przypadków użycia w Cyklu Życia Automatyzacji (rys. 3.11).

Diagram klas

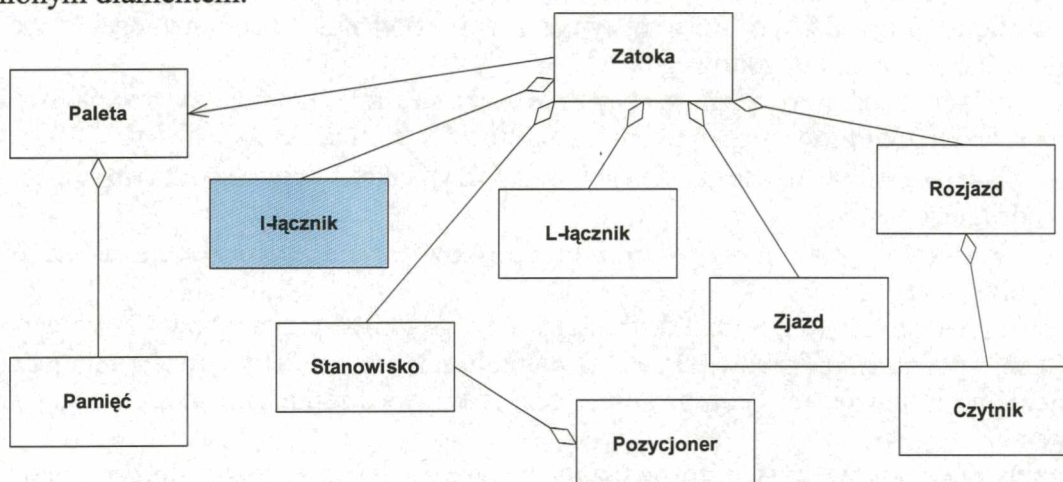
Diagram Modelu Obiektowego jest najważniejszą prezentacją modelowania obiektowego. Pokazuje on architekturę systemu poprzez klasy i ich instance oraz statyczne relacje między nimi. Jego znaczenie jest niezwykle ważne, gdyż implikuje on strukturę i zachowanie systemu. Relacje pokazują drogi, zgodnie z którymi mogą komunikować się klasy. W celu bardziej precyzyjnego określenia relacji między klasami lub obiektami wyróżnia się cztery ich typy:

- asocjacja,
- agregacja,
- kompozycja.

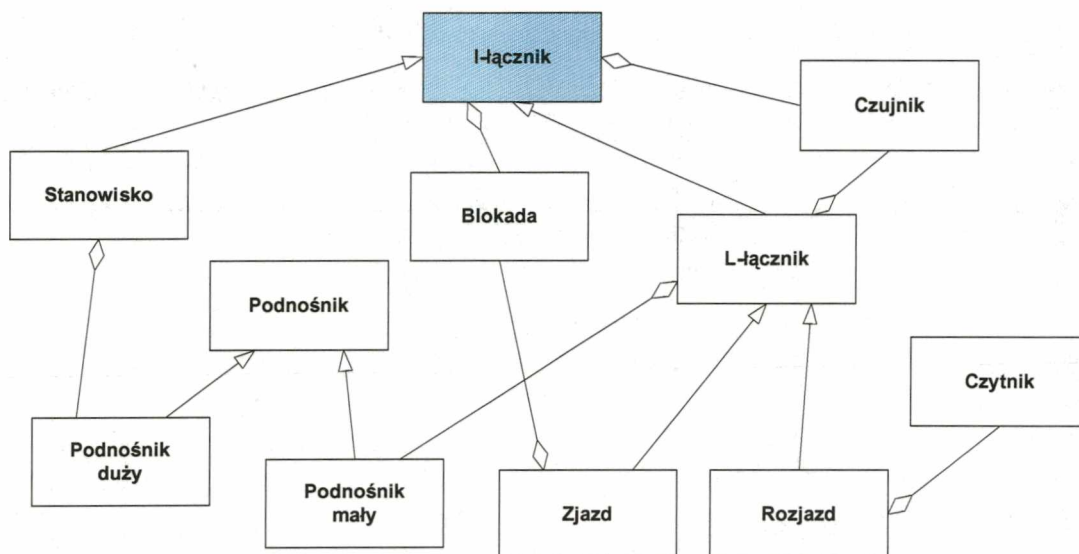
Asocjacja stanowi najprostszy sposób relacji między klasami. Jest nawiązywana, gdy jeden obiekt używa usług innego obiektu, których sam nie posiada. Obrazowana jest ona jako odcinek łączący, który może być opatrzony etykietą nazywającą. Interfejsy każdej z klas wymagają jednak nazwy, określanej jako rola. Będą one wykorzystywane przy adresowaniu kanałów komunikacyjnych. W przypadku relacji dwukierunkowej nawigacja (przechodzenie) jest możliwa w obu kierunkach. Dla precyzyjniejszego zobrazowania systemu możemy wprowadzić asocjację kierunkową zaznaczając strzałką kierunek nawigacji. Ważnym elementem asocjacji jest oznaczenie mnogości występowania. Możemy ją podać zgodnie z szeregiem: 0..1; 1; n; 1..n; *.

Agregacja jest stosowana, wówczas, gdy jeden obiekt fizycznie lub koncepcyjnie zawiera inny. W ten sposób pokazywana jest zależność typu całość-część. Graficznie na diagramie obrazowana jest ona jako odcinek zakończony pustym diamentem po stronie klasy całościowej. Pozostałe właściwości agregacji, takie jak nazwa roli, kierunkowość i mnogość występowania są identyczne jak w przypadku asocjacji.

Kompozycja jest silniejszą formą agregacji. Wymusza ona dodatkowo kontrolę nad czasem życia obiektów stanowiących jej części składowe. Odpowiedzialna jest ona zatem za tworzenie obiektów i ich usuwanie. Graficznie przedstawia się ją jako agregację z wypełnionym diamentem.



Rys. 4.18a Przykład obiektowego diagramu klas dla zatoki EGM-ITMA [jr]



Rys. 4.18b Przykład obiektowego diagramu klas dla I-łącznika EGM-ITMA [jr]

Jeśli jedna klasa jest specjalizacją innej, to relację między nimi nazywamy generalizacją lub dziedziczeniem. Oznacza to, że potomek przejmie atrybuty i operacje klasy nadrzędnej, pod warunkiem, że nie były one zdefiniowane jako prywatne. W klasie potomka można dedefiniować atrybuty i operacje oraz zmodyfikować przejęte. Zależność dziedziczenia graficznie przedstawiana jest jako odcinek zakończony trójkątem, którego wierzchołek wskazuje klasę nadrzędną. Generalizacja należy do niezwykle silnych mechanizmów podejścia obiektowego. Dzięki niej możliwa jest specyfikacja obiektów poprzez wskazanie różnic, a nie każdorazowe tworzenie ich od nowa. W klasycznym podejściu strukturalnym rozszerzanie lub specjalizacja funkcji wiąże się z modyfikacją kodu źródłowego.

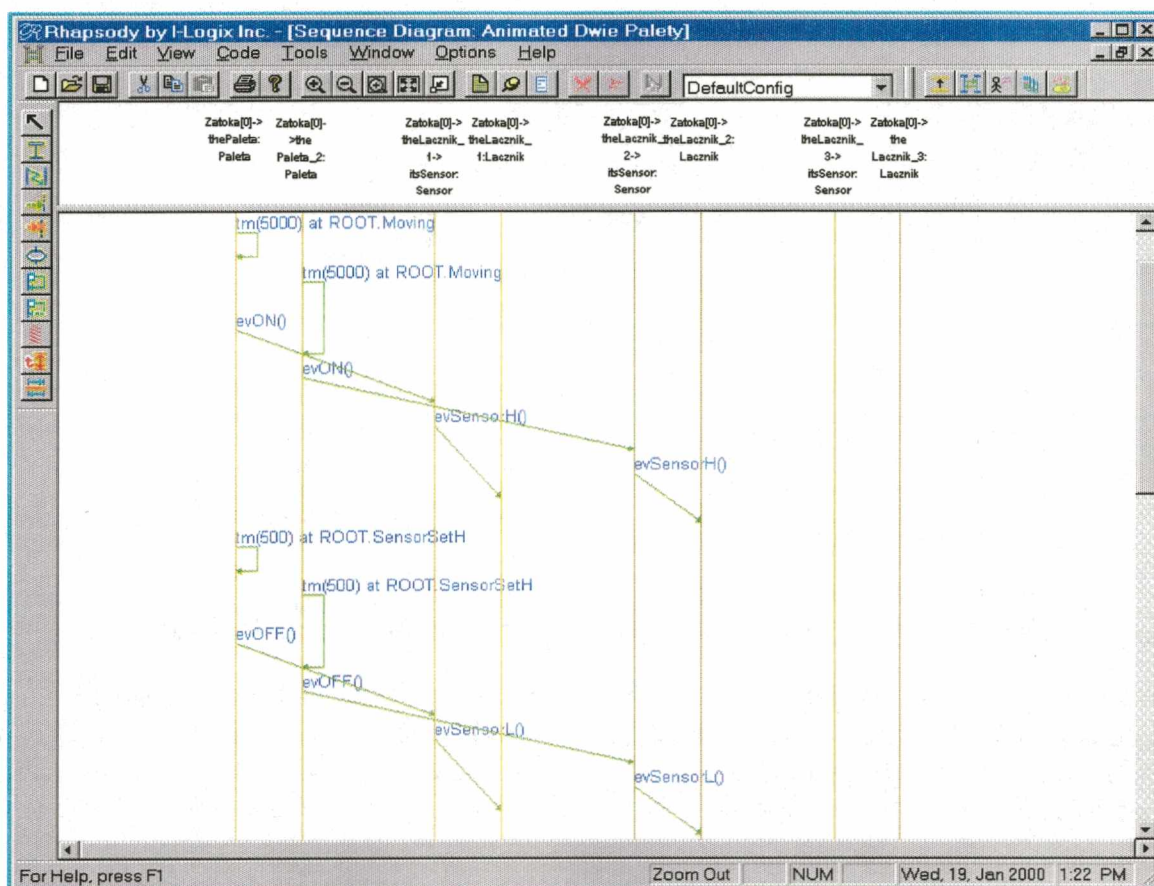
- Dla graficznego zobrazowania specjalnych zależności między elementami diagramów stosowany jest odcinek linii przerywanej, ze strzałką skierowaną, wraz z dodatkową etykietą. Język UML definiuje sześć zależności „zaprzyjaźnienia”

Diagram klas został użyty we wcześniejszym rozdziale do zamodelowania rodziny produktów automatyzacji – patrz rys. 3.18.

Na rys 4.18a i 4.18b przedstawiono diagramy klas dla elementów elastycznego gniazda montażowego w ITMA.

Diagram interakcji

Podobnie jak w większości koncepcji języków obiektowo-zorientowanych, przypadki użycia również wykazują dwudzielną: typ – instancja. Scenariusz jest konkretną realizacją przypadku użycia, który krok po kroku pokazuje wymianę komunikatów między aktorami a składowymi systemami (rys. 4.19). Diagramy sekwencji są używane do uchwycenia najistotniejszych interakcji z systemem i interakcji wewnętrznych, a więc dynamiki modelu. Dla zróżnicowania poziomu szczegółowości używane są scenariusze przypadków użycia oraz scenariusze obiektowe. Jedyna różnica między nimi polega na rodzaju instancji występujących w interakcjach. W pierwszym przypadku występują aktorzy i składowe systemy, natomiast drugi z nich pokazuje interakcje na poziomie konkretnych obiektów.



Rys. 4.19 Przykład diagramu interakcji dla EGM-ITMA [jr]

Diagramy sekwencji są szczególnie przydatne w dwóch fazach projektu:

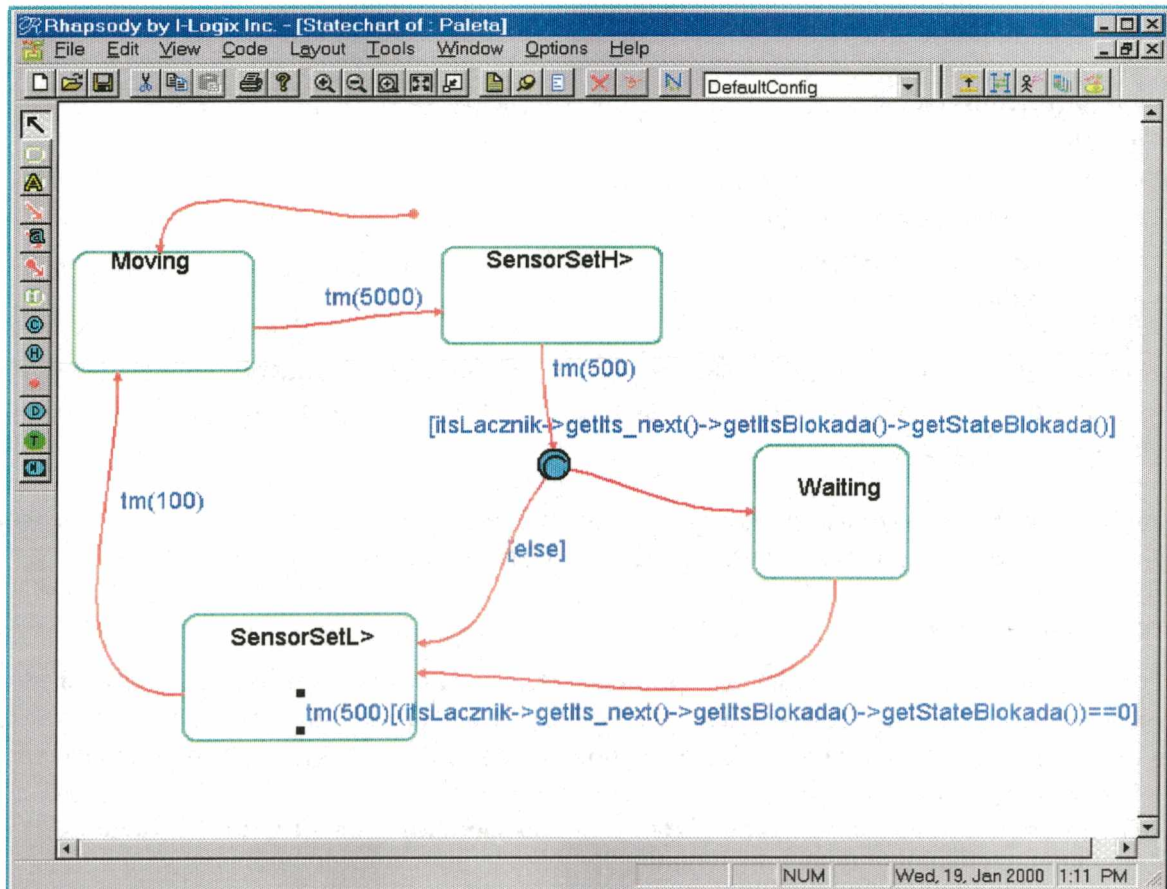
- analizy, gdy należy szczegółowo uchwycić wymagania systemu oraz
- testowania i integracji, gdy analiza interakcji między obiektami zapewnia zgodność z projektem.

Diagramy sekwencji budowane są z obiektów i komunikatów. Pionowa linia reprezentuje obiekt, a nie klasę. Możliwe jest opcjonalne dodanie nazwy klasy dla zwiększenia czytelności. Obiekt klasy wysyłający komunikat musi posiadać jakąś asocjację z obiektem odbierającym, a rola obiektu odbiornika musi mieć taki sam typ jak nazwa odbieranego obiektu. Forma grotu strzałki symbolizująca komunikat określa jego typ.

Diagram interakcji zastosowano do zamodelowania scenariusza komunikacji klient-dostawca podczas automatyzacji w rozdziale 3 – patrz rys. 3.23.

Diagram stanów

Diagram stanów określa zachowanie reakcyjnych obiektów poprzez specyfikację ich działania pod wpływem zdarzeń lub operacji (rys.4.20). Działanie następuje pod wpływem przejść między stanami, jak również wykonania akcji. Diagram stanów bazuje na automacie stanów, dodając hierarchiczne zagnieżdżenia i regiony ortogonalne oraz rozszerzając notację akcji.



Rys. 4.20 Przykład Diagramu Stanów EGM-ITMA (Rhapsody) [jr]

Obiekt, którego zachowanie jest opisane poprzez diagram stanów nazywany jest reakcyjnym. Stan jest „uchwyceniem obiektu”, w którym akceptowany jest zbiór zdarzeń, realizowane są pewne akcje i aktywności, a ponadto obiekt może przejść do określonego zbioru stanów po zaakceptowaniu zdarzeń. Zmianie stanu towarzyszą następujące działania:

- akceptacja zdarzenia wejściowego,
- przejęcie tranzycji wyjściowej jako następstwo zdarzenia wejściowego,
- wykonanie akcji wejścia do stanu,
- wykonanie akcji wyjścia ze stanu,
- wykonanie.

Diagram czynności

Diagram czynności, zwany również diagramem aktywności (ang. *activity diagram*) [UML 99], [Rhapsody 00] specyfikuje przepływ danych lub działań (ang. *workflow*) i jest preferowany wówczas, gdy zachowanie nie jest determinowane poprzez zdarzenia. Diagram aktywności jest szczególnie przydatny do definicji algorytmów. Za jego pomocą mogą być specyfikowane klasy, przypadki użycia lub operacje. Podstawowa różnica między nim, a diagramem stanów polega na definicji węzłów decyzyjnych, stanów aktywności oraz przebiegu przetwarzania. Węzły decyzyjne są to rozgałęzienia przebiegu sterowane warunkami. Stany akcji reprezentują wywołania funkcji, które nie muszą być zlokalizowane w tym samym obiekcie. Diagramy aktywności nie stosują stanów typu „i” (ang. „and”), lecz współbieżność przedstawiają za pomocą rozgałęzień. W ten sposób równoległe, zależne przetwarzanie, modelowane za ich pomocą, jest znacznie czytelniejsze. Dla poprawy czytelności diagramów można stosować tory (ang. *swimlines*), które grupują stany czynności.

Semantyka diagramów czynności obejmuje również przepływ obiektów oraz zobrazowanie stanu wartości atrybutów obiektów. Służą one do badania logiki:

- złożonych operacji,
- złożonych reguł biznesowych,
- pojedynczych przypadków użycia,
- wielu przypadków użycia,
- procesów biznesowych,
- procesów software'owych.

Składnia diagramów czynności obejmuje stany i operacje, tranzycje, punkty decyzyjne, rozgałęzienia i połączenia. Z ich użyciem tworzone są podstawowe konstrukcje do modelowania dynamiki.

Diagram aktywności został wykorzystany we wcześniejszym rozdziale do zamodelowania interakcji między klientem a dostawcą podczas procesu automatyzacji – rys. 3.19.

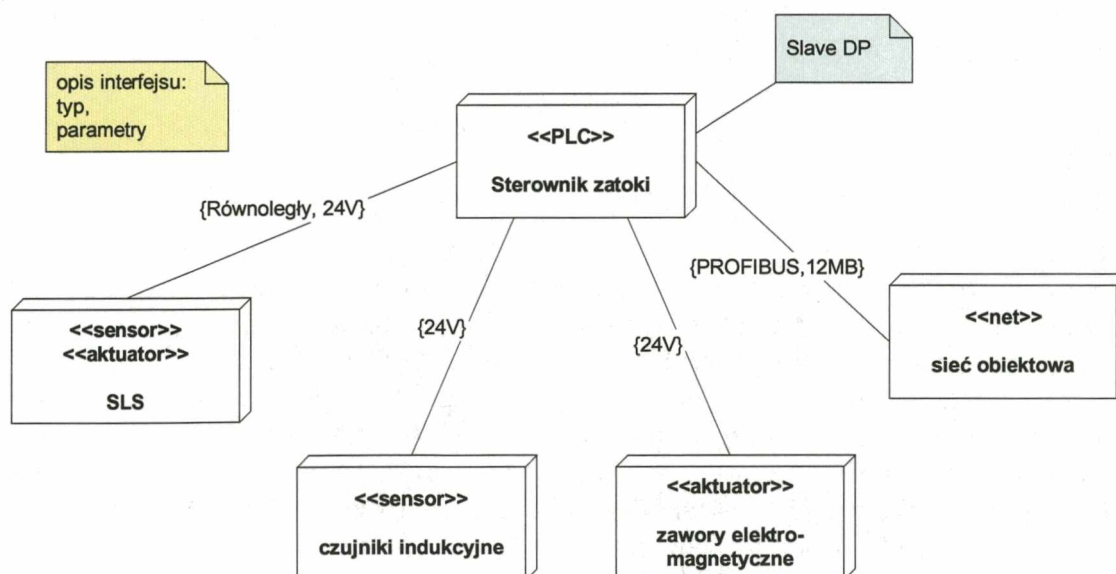
Reprezentacja fizyczna

„Specjalność” systemów *embedded* polega na konieczności uwzględnienia różnorodności zastosowanego sprzętu. Tradycyjne metody modelowania oprogramowania nie zapewniały sposobu integracji oprogramowania ze sprzętem, ani też nie wyjaśniały zależności między logicznym oprogramowaniem a fizyczną realizacją.

UML udostępnia dwa główne typy diagramów dla modelowania struktury fizycznej: diagram rozmieszczenia (ang. *deployment*) i diagram komponentów (ang. *component*).

Diagram rozmieszczenia jest uproszczoną formą architektury sprzętu elektronicznego, w którym będzie przetwarzane oprogramowanie. Elementami tego diagramu są węzły (urządzenia fizyczne) oraz połączenia między nimi (rys. 4.21).

UML używa diagramu komponentów do pokazania specjalnych rozwiązań projektowych dla systemów czasu-rzeczywistego. Komponenty te mogą stanowić statyczne organizacje oprogramowania, takie jak: pliki źródłowe, moduły testujące, konfiguracje projektowe, albo też dynamiczne jak np.: biblioteki wykonywalne i podsystemy. Dla przykładu podsystem może zawierać obiekty komunikacyjne, obiekty interfejsu użytkownika lub obiekty modułów I/O.



Rys 4.21 Przykład diagramu rozmieszczenia dla zatoki EGM-ITMA [jr]

Elementy wspólne

Do najistotniejszych elementów wspólnych UML należą notatki, pakiety, wymuszenia i stereotypy.

Notatki tekstowe są elementem bez semantycznego wpływu. Używane są one na wszystkich diagramach UML i zaznaczane jako prostokątne ramki z zawiniętym w dół górnym prawym narożnikiem.

Pakiety w UML prezentowane są jako foldery. Reprezentują one logiczne grupowanie elementów, które wirtualnie mogą być dowolne (klasy, obiekty). Popularne użycie pakietów dotyczy reprezentacji domen. Domeny stanowią niezależne elementy modelowania z własnym nazewnictwem i modelem obiektowym, jak np.: „interfejs użytkownika”, „urządzenie we/wy”. Cechą takiego podejścia jest wyizolowanie niezależnych obszarów, które mogą być analizowane autonomicznie. Pakiety mogą mieć interfejsy, podobnie jak klasy i mogą zawierać specjalizacje. Interfejs pakietu jest zdefiniowany poprzez zestaw usług, klas i obiektów widzialnych z zewnątrz pakietu. Pakiet może być specjalizowany i zawierać podklasy. W takim przypadku interfejs pozostaje niezmieniony, a zmienia się implementacja każdej specjalizacji.

Wymuszenia stanowią dodatkowe warunki nałożone na elementy modelu. Wymuszenia czasowe mogą specyfikować na diagramie sekwencji czas między komunikatami. Na diagramie są one umieszczane w nawiasach klamrowych „{ }” w tekście notatki.

Stereotyp jest nowym elementem modelowania wywodzącym się z istniejącej meta-klasy w meta-modelu UML. Meta-model UML jest modelem UML wyrażonym w UML. Stereotypy zapewniają ważny mechanizm rozszerzania języka modelowania dla lepszego adresowania potrzeb. Każdy element modelowania w UML jest reprezentowany jako meta-klasa. Stereotyp meta-klasa jest ostatecznie zaczerpnięty z istniejącej meta-klasy UML.

4.4 Obiektowość i UML dla automatyzacji wytwarzania

Zastosowanie obiektowości dla automatyzacji wytwarzania podejmowane jest głównie w ramach projektów badawczych. Obiektowość w C++ czy Java, wprowadzana jest jako alternatywny, dla IEC 61131, STEP7, etc., język programowania. Taka innowacja wymaga opracowania własnej platformy sterownika tzn. sprzętu i oprogramowania systemowego. Dostępność komputerów typu *embedded* oraz oprogramowania RTLinux (LinuRT), Windows CE i Windows Embedded znacznie redukuje koszty takiej implementacji. Jednak z uwagi na elitarność takich rozwiązań, tzn. potrzebę wysokiej sprawności programistów obiektowych, propozycje takie nie znajdują szerokiego zastosowania w praktyce. Drugie zastosowanie obiektowości, choć w bardzo zubożonej formie, realizowane jest jako modularność. Przykładem tego jest platforma OSACA (patrz rys. 3.18) lub projekt MOWIMA. Projekt ten realizowany w ISW Stuttgart, pt.: „Modelowanie i Ponowne Wykorzystanie Obiektowo-Zorientowanego Oprogramowania Maszyn” [MoWiMa 97]) proponuje opracowywanie oprogramowania na bazie komponentów definiowanych w katalogach.

Dla modelowania kompleksowych systemów wytwórczych proponuje się zastosowanie modelowania obiektowego na poziomie strategicznym i logicznym.

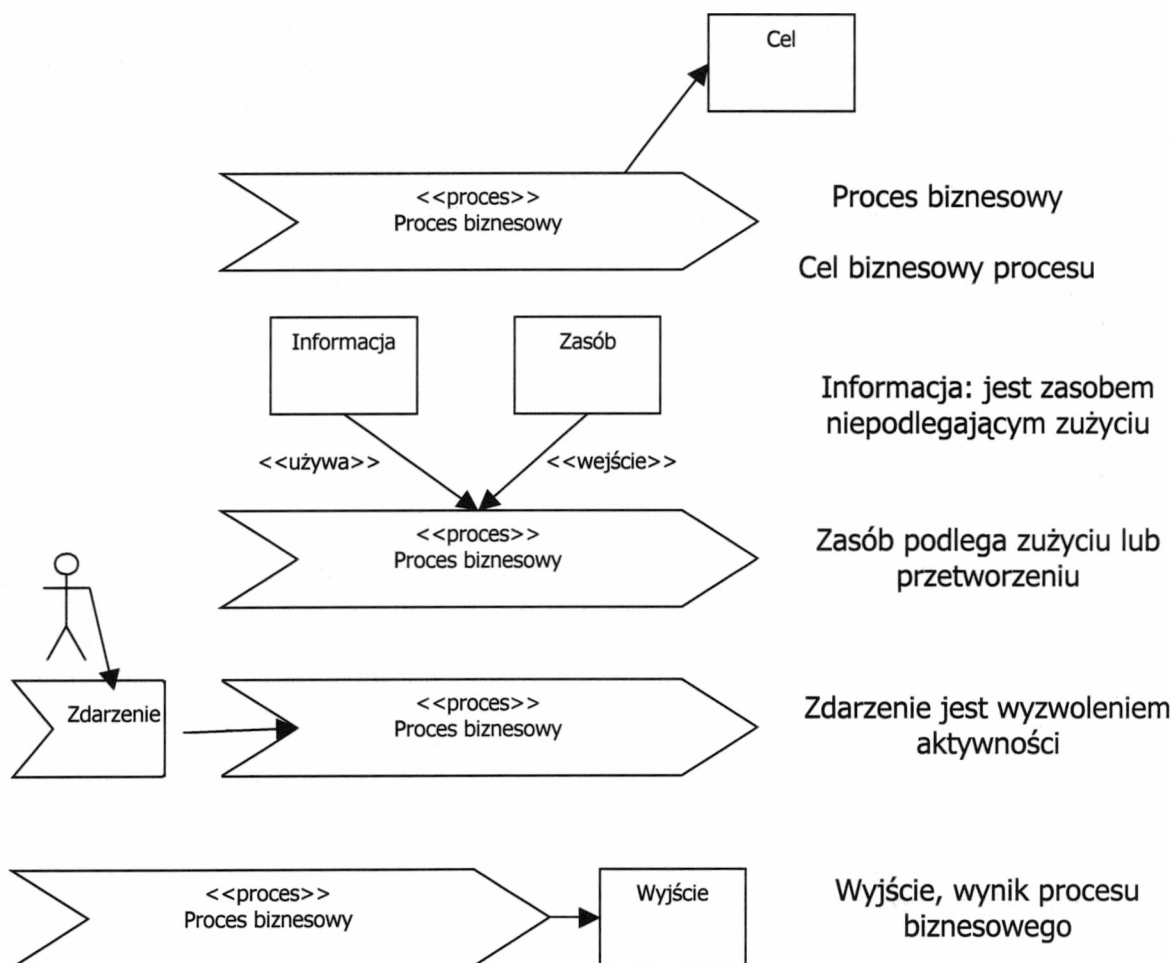
Oryginalność tej propozycji polega na „nauczeniu” obiektowości, modelowania obiektowego i języka UML użytkownika systemów automatyzacji. Dzięki takiemu podejściu klient będzie lepiej przygotowany do formułowania swoich potrzeb oraz weryfikacji otrzymanego rozwiązania (patrz rys.3.19).

Kompetencje z zakresu Modelowania Obiektowego dla Automatyzacji Wytwarzania (MOAW) zgrupowano w czterech kategoriach:

- modelowanie biznesowe procesów automatyzacji,
- modelowanie logicznej struktury procesów,
- modelowanie zakresu i funkcjonalności systemów,
- modelowanie dynamiki systemu i jego obiektów.

4.4.1 Modelowanie biznesowe procesów automatyzacji

Głównym przedmiotem konkurencji między przedsiębiorstwami są ich wyniki finansowe. Dlatego też w gospodarce wolnorynkowej, stymulowanej globalizacją, zadowolenie klientów czy pracowników musi ostatecznie zostać przeformułowane w wymierne wskaźniki ekonomiczne. Stąd wynika perspektywa udziałowców i sponsorów, która upraszcza przedmiot biznesu, nie wnikając w świat „imponujących” rozwiązań technicznych. „Świat automatyzacji”, który nie zawsze akceptuje fakt, że to on jest na usługach produkcji - a nie odwrotnie, chętnie zupełnie odcina się od biznesowego spojrzenia na problem produkcji. Jak wykazano w rozdziale 3, informatyzacja i automatyzacja postrzegane są coraz częściej jako inwestycje, przed których podjęciem musi być określona ich rentowność.



Rys.4.22 Składnia języka UML dla modelowania procesów biznesowych [jr na podst. Sparks 00]

„Światom automatyzacji i biura” wiele uwagi poświęcono w publikacjach z zakresu komunikacji przemysłowej. Komunikacja, jako narzędzie integracji pytała, dlaczego jeden wspólny standard komunikacyjny nie może połączyć obszarów biura i produkcji, przecież wiele danych i informacji jest wspólnych? Początkowo były to problemy czysto techniczne,

np.: czas rzeczywisty, zakłócenia w środowisku przemysłowym, itp. Przełom wieku XX/XXI możemy uznać za kres problemów technicznych, gdyż komunikacyjne sieci biurowe, a nawet Internet, mogą zagwarantować dostateczne wymagania obszaru produkcyjnego [Skura 2002]. Problemy istnieją jednak nadal, ale ich rozwiązań poszukuje się nie w technice, lecz organizacji a nawet filozofii.

Jakie jest biznesowe spojrzenie na automatyzację wytwarzania i na procesy biznesowe?

Jak sponsorzy rozumieją jej potrzebę i uzasadnienie? Na podstawie analizy dokumentacji technicznej zautomatyzowanych rozwiązań należy stwierdzić, że problem właściwie nie istnieje. Filozofia i strategia automatyzacji „przychodzą” bowiem wraz z produktem, a później podczas eksploatacji są dopiero „odgadywane”. Stąd wynika brak spójnego podejścia do sposobu automatyzowania i w konsekwencji poszukiwanie rozwiązań pod kątem najniższych kosztów.

Procesy biznesowe są kolekcją aktywności zaprojektowaną dla uzyskania charakterystycznego wyniku dla klienta (użytkownika) lub rynku. Podczas ich definiowania zwraca się uwagę na to, jak przebiega praca, bez charakterystycznej dla produktu analizy, co jest robione. Podczas analizy biznesowej skupia się uwagę na celach nadrzędnych, szansach i zagrożeniach, słabych i mocnych stronach (analiza SWOT⁵) oraz interfejsach z otoczeniem, tj. bezpośrednio klientem lub innymi działami firmy.

W modelowaniu procesów biznesowych zwraca się uwagę na następujące elementy:

- cel i sens procesu,
- charakterystykę wejść,
- charakterystykę wyjść,
- wymagane zasoby,
- wykonywane czynności i ich kolejność oraz
- zdarzenia sterujące procesem.

Żaden z diagramów UML nie wspiera modelowania procesów biznesowych. Jednak dzięki stereotypom możliwe jest łatwe rozszerzenie składni języka o wymagane elementy przez użytkownika. Dla modelowania procesu biznesowego i wspierającego go procesu automatyzacji skorzystano z rozszerzenia opracowanego przez Erikssona i Penkera [Eriksson, Penker 00] (rys. 4.22).

Dla graficznego zamodelowania powyższych elementów, zdefiniowano następujące elementy graficzne:

Proces wytwórczy opisany w sposób biznesowy porządkuje „postrzeganie” procesu przez użytkowników wewnętrznych, tj. na codzień zajmujących się procesem, ułatwia również wprowadzenie do tematu, dla użytkowników zewnętrznych, którzy będą uczestniczyć w jego modernizacji. Ponadto, pozwala wskazać rolę automatyzacji, jej produkty, interfejsy i otoczenie, dla osób nie związanych bezpośrednio z rozwiązaniami technicznymi lecz finansowymi, tj. np. sponsorów.

Dobrze zdefiniowane założenia strategiczne pozwolą następnie rozwijać spójną koncepcję realizacji automatyzacji wytwarzania. W przeciwnym przypadku dochodzi do „zlepku” różnych niespójnych, choć indywidualnie bardzo dobrych, rozwiązań. Znajdowane optimum rozwiązania powinno bazować na kryteriach optymalizacji globalnych a nie lokalnych.

4.4.2 Modelowanie logicznej struktury procesów

Model biznesowy w perspektywie inżynierskiej jest „mało konkretny”. Inżynier postrzega systemy techniczne jako obiekty i relacje między nimi, poszukując odpowiedzi „co” i „jak

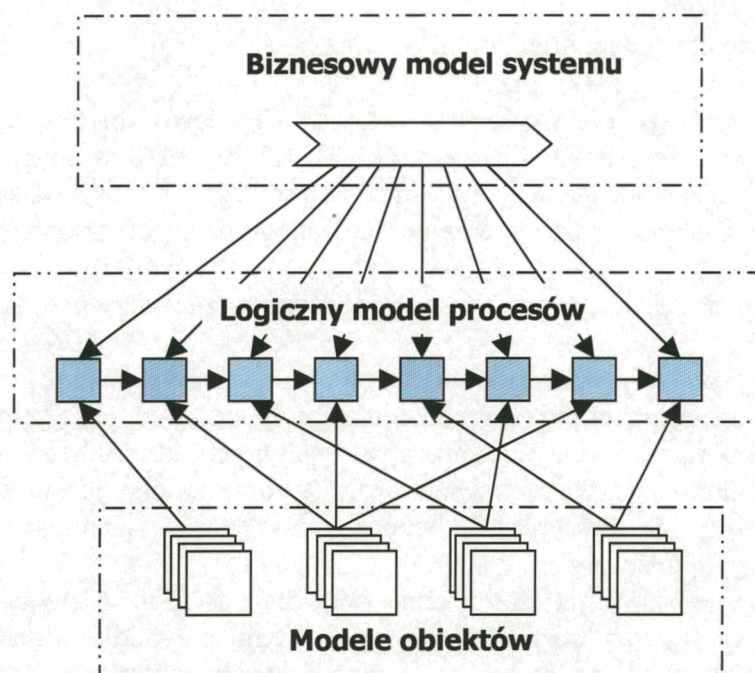
⁵ analiza stron mocnych, słabych, szans i zagrożeń (ang. Strengths, Weaknesses, Opportunities, Threats)

działa”. Model logiczny procesu, czyli operacje technologiczne i relacje między nimi, już przy średniej wielkości instalacjach technologicznych, wyraźnie się rozbudowuje. Przedstawianie modelu na większym formacie rysunku lub w wyższej rozdzielczości na monitorze jest tzw. „rozwiązaniem siłowym”. Nie wspiera ono predyspozycji postrzegania przez człowieka, które wskazują na optymalną liczbę 7-12 istotnych obiektów na diagramie. Stąd też wynika modelowanie kontekstowe wspierane przykładowo przez modelowanie IDEF [IDEF web], które przedstawia model na różnych poziomach szczegółowości, zgodnie z podejściem od ogółu do szczegółu (ang. *top-down*). W ten sposób problem zostaje uproszczony, ale nie zredukowany. To znaczy, liczba elementów do analizy pozostaje niezmienna.

Redukcję kompleksowości modelu wspiera podejście obiektowe. Jak scharakteryzowano wyżej, zgodnie z paradygmatem obiektowym pierwszoplanowa jest klasyfikacja, tj. poszukiwanie podobieństw i uogólnień, a nie koncentrowanie się na rozwiązaniu.

Zgodnie z takim podejściem, celem pierwszoplanowym jest charakterystyka posiadanych zasobów. Klasyfikacja nie jest jednak zadaniem czysto formalnym, a identyfikacja odpowiednich nazw klas i ich zakresu znaczeniowego nie jest jednoznaczna. Metody wspierające identyfikację klas bazują na pracy grupowej i doświadczeniu. Punktem wyjścia do klasyfikacji jest jednak model biznesowy oraz logiczna struktura procesu technologicznego (rys.4.23).

Do utrwalenia opracowywanego modelu klas proponuje się zastosowanie wcześniej omówionego obiektowego diagramu klas języka UML. Diagramy klas, klasyfikując różne komponenty lub abstrakcje (np.: błędy montażowe) stanowią słownik pojęć. Dzięki klasyfikacji skala problemu ulega redukcji.



Rys 4.23 Relacje między modelem biznesowym, logicznym i obiektowym [jr]

Syntaktyka i semantyka modelu logicznego w ujęciu procesowym, powinna wspierać również możliwość modelowania relacji między czynnościami. W wielu przypadkach diagramy czynności są obiektowo-zorientowanym odpowiednikiem diagramów przepływu lub diagramów przepływu danych DFD (ang. *Data Flow Diagram*). Dzięki możliwości modelowania współbieżności oraz równocześnie przepływu danych, diagramy te są efektywnym narzędziem modelowania. Diagramy aktywności można przyrównać do sieci

Petrieego, których pojemność każdego z miejsc (ang. *places*) jest ograniczona do jednego znacznika (ang. *tokena*).

4.4.3 Modelowanie zakresu i funkcjonalności systemów

Dla lepszego zrozumienia wymagań systemu, niezależnie od zastosowanej techniki jego budowy, obiektowej czy tradycyjnej, często stosowane są także opisy typowych interakcji użytkownika z systemem. Zwykle jednak scenariusze takie traktowane są nieformalnie, tj. nie są dokumentowane.

Przypadki użycia (rys. 4.17 jak również rys. 2.1) są tekstowymi dokumentami, które opisują „CO?” system powinien robić, z wysoko-poziomowej perspektywy użytkownika, określając w ten sposób zakres projektu oraz nadając mu strukturę. Przypadki użycia nie są zatem modelem funkcjonalnej dekompozycji. Ich zadaniem nie jest również uchwycenie wszystkich wymagań dla systemu, bądź próba odpowiedzi „JAK?” system lub jego składnik ma działać. Przypadki użycia nie uwzględniają również cech użytkowników, które nie mają bezpośredniego związku z systemem. Dodatkowo należy podkreślić, że przypadki użycia nie są techniką wywodzącą się z podejścia obiektowo-zorientowanego.

Przypadek użycia nie potrafi samodzielnie inicjować działania. Każde działanie w systemie inicjowane jest przez aktora – funkcję użytkownika zewnętrznego. Funkcja użytkownika, to rola jaką pełnią osoby wobec systemu. Tak więc, ta sama fizyczna osoba może reprezentować wielu użytkowników. Użytkownikiem systemu typowo jest człowiek, ale może być nim również inny system, komputer czy nawet bardziej abstrakcyjne formy jak, np. zegar.

Każdy aktor może inicjować wiele funkcji systemu, przy czym oddziaływania aktora z systemem oznaczane są przez relacje.

Uogólnienie wskazuje, że dany przypadek użycia jest podobny do innego, ale jest on obszerniejszy. Relacja uogólnienia jest odpowiednikiem obiektowego dziedziczenia i jest ona używana do uchwycenia scenariuszy alternatywnych.

Relacja rozszerzenia (ang. *extends*) jest podobna do relacji uogólnienia, jest jednak ona bardziej formalna. Rozszerzenie przypadku użycia może wzbogacić go o dodatkowe zachowania, ale w takiej sytuacji, podstawowy przypadek użycia musi określić pewne dodatkowe punkty rozszerzenia (ang. *extensions points*), zaś rozszerzający przypadek użycia może dodawać rozszerzenia tylko w tych punktach. Zgodnie ze standardem UML ver. 1.4 [UML 1.4], relacja rozszerzenia powinna jako atrybut specyfikować warunek (ang. *conditions*), który musi zachodzić, aby rozszerzenie następowało.

Ponieważ przypadki użycia definiują interakcje między użytkownikiem a systemem, to często dla ich znalezienia pomocne jest spojrzenie na problem z dalszej perspektywy. Obserwacja, jakie przypadki użycia występują między przedsiębiorstwem a klientem, nazywane biznesowymi przypadkami użycia, pomaga zaproponować odpowiednie realizacje systemowych przypadków użycia [Jacobson, et al. 95].

Do najistotniejszych cech przypadków użycia należy zaliczyć następujące:

- definiują zakres systemu oraz wizualizują wielkość i zakres projektu;
- są podobne do specyfikacji wymagań, która jednak w przeciwieństwie do przypadków użycia, często jest niezbyt precyzyjna, ogólnikowa a nawet zagmatwana. Jakość przypadków użycia dzięki zwęższej strukturze, czyni je bardziej skoncentrowanymi na problemie,
- „suma” przypadków użycia określa cały system. To, co nie jest wyspecyfikowane, nie należy do systemu,
- dzięki prostocie stają się one narzędziem komunikacji projektanta z klientem,
- mogą one prowadzić proces rozwoju projektu dla całej grupy projektowej,
- zapewniają metodykę planowania prac merytorycznych, znajdowania priorytetów i ułatwiają szacowanie czasu trwania projektu,

- stanowią podstawę do opracowania testów systemu,
- są podstawą instrukcji obsługi systemu.

4.4.4 Modelowanie dynamiki systemu i jego obiektów

Dynamika systemu została wstępnie nakreślona podczas modelowania struktury procesu, gdyż to struktura jest nośnikiem działania. Model procesu jednak tylko ogólnie określa interakcje między komponentami i nie wnika w mechanizmy ich zachowania.

Konwencjonalne podejście do sterowania bazuje na działaniu cyklicznym (rozdział 1.2). Otoczenie generuje jednak zdarzenia w sposób niedeterministyczny. Tak więc rozwiązania zdarzeniowe (ang. *event driven*), takie jak automaty stanów lepiej modelują rzeczywisty proces technologiczny. Jednak ich akceptacja i zastosowanie są niewielkie. Użytkownicy nie potrafią i nie mają odpowiedniego wsparcia narzędziowego dla analizy i zagwarantowania ich wydajności i niezawodności.

Do modelowania komunikacji między obiektami dedykowany jest diagram interakcji języka UML (rys. 4.19 i rys. 3.24). Diagram ten jest również użyteczny do specyfikowania wymagań systemu szczególnie interakcji użytkowników z systemem. Diagram interakcji jest scenariuszem specyfikującym komunikaty wysyłane między partnerami komunikacji.

Do modelowania zachowania poszczególnych obiektów, język UML oferuje natomiast diagram stanów (rys.4.20). Modelowanie za pomocą diagramu stanów może być prowadzone na różnym poziomie szczegółowości. Najwyższym poziomem jest zachowanie maszyny lub urządzenia wytwórczego. Model taki uzupełnia instrukcje przy stanowiskach i stanowi interfejs między implementacją a widocznym działaniem.

4.5 Podsumowanie rozdziału czwartego

W rozdziale 4 zbadano wybrane techniki i narzędzia wspierając komunikację wiedzy dla automatyzacji wytwarzania.

Na wstępie przeanalizowano szkice i rysunki oraz mapy pamięci dla komunikowania koncepcji. Wskazano na brak standaryzacji ich syntaktyki i semantyki ograniczający zakres potencjalnych jej użytkowników oraz uniemożliwiający wsparcie komputerowe.

W kolejnym etapie przebadano zagadnienie modelowanie, tj. najważniejsze narzędzie abstrakcji, dla radzenia sobie z kompleksowością. Wskazano na znaczenie modelowania dla poznawania rzeczywistości, przedstawienia projektów i dokumentowania. Wykazano, że modelowanie jest nie tylko podstawowym narzędziem nauki, ale w coraz większym stopniu stosowane jest przez inżynierię.

Badanie modelowania w automatyzacji wskazało, że test ono stosowane głównie jako programowanie graficzne. Z uwagi na swoje podstawy elektrotechniczne, nie jest ono efektywnym narzędziem komunikacji między automatykami i nie-automatykami. Ponieważ programowanie opisuje rozwiązanie na „niskim poziomie”, czyli przez pryzmat implementacji, słabo wspiera poziomu strategiczny i logiczny problemu. Jak wykazano w rozdziale 3, właśnie poziomy koncepcji i logiczny są podstawą komunikacji w grupach interdyscyplinarnych oraz komunikacji zewnętrznej klient - dostawca.

W rozdziale 4 zaprezentowano paradygmat obiektowy i wybrane elementy języka UML, których zastosowanie przedstawiono w kontekście problematyki automatyzacji wytwarzania. Do prezentacji wykorzystano diagramy z opracowanej dokumentacji obiektowej dla

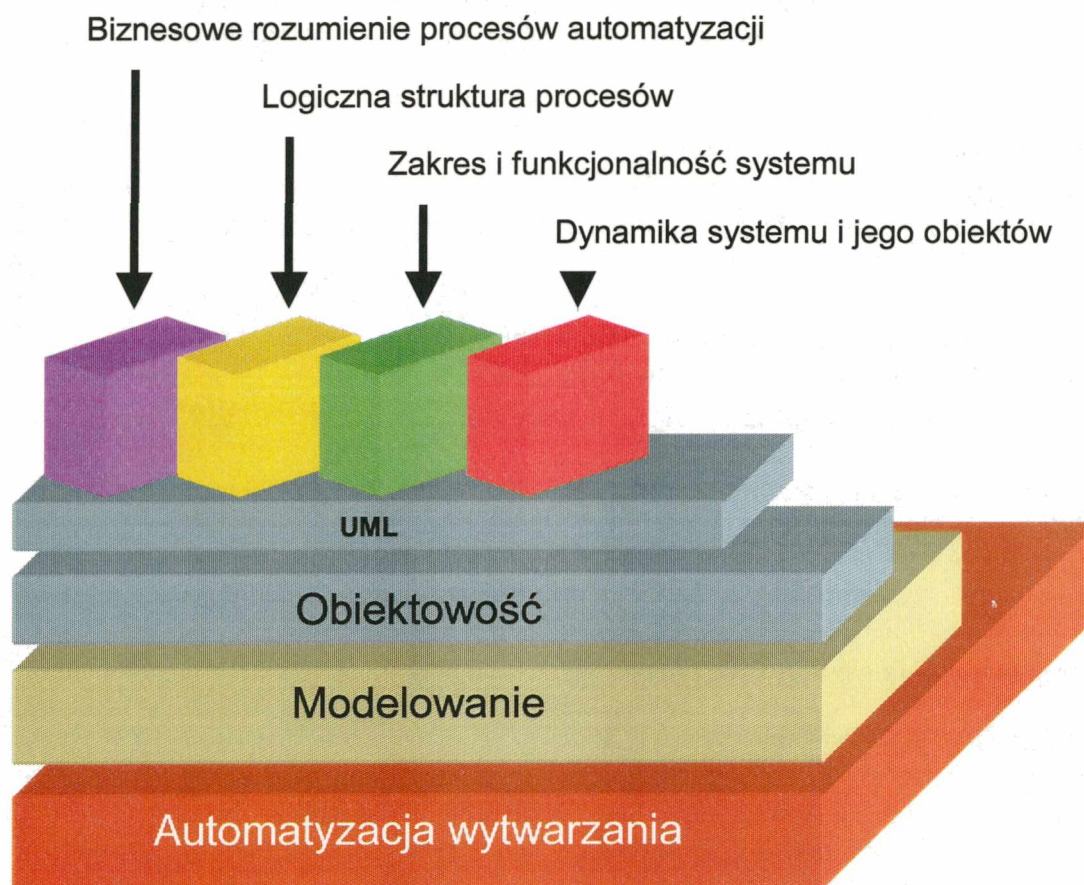
rozproszonego sterowania w elastycznym gnieździe montażowym EGM w ITMA [Koch, et al. 98],[Reiner, et al. 00].

Określono przy tym cztery obszary kompetencyjne w zakresie:

- biznesowego rozumienia procesów automatyzacji,
- logicznej struktury procesów,
- funkcjonalności systemu,
- dynamiki systemu i jego obiektów.

Powyższy zakres kompetencji, elementy języka UML oraz podstawy obiektowości i modelowania, osadzone na problematyce automatyzacji wytwarzania, zostały nazwane MOAW (Modelowanie Obiektowe dla Automatyzacji Wytwarzania) (rys. 4.24).

Umiejętności takie, w rozumieniu „zarządzania wiedzą”, nazywane są narzędziem zarządzania wiedzą.



Rys 4.24 Architektura narzędzia zarządzania wiedzą MOAW [jr]

Efektywność narzędzi zarządzania wiedzą zależy w głównej mierze od ich wdrożenia i użytkowania, a szczególnie krytycznym ogniwem tego procesu jest człowiek i kultura organizacyjna. Dlatego też w kolejnym rozdziale, zostanie opracowana szkieletowa metodyka wdrożenia MOAW oraz zostanie zwrócona uwaga na wybrane elementy decydujące o sukcesie wdrożenia.

5 Wdrażanie narzędzia zarządzania wiedzą MOAW

Dla zbadania słuszności postawionej tezy, w rozdziale 5 skoncentrowano się na drugiej części postulatu:

„Poprawę efektywności automatyzacji wytwarzania można osiągnąć dzięki odpowiednim kompetencjom w zakresie modelowania obiektowego i języka UML oraz poprzez odpowiednie zarządzanie nimi.”

Dla Rozdziału 5 postawiono następujące problemy badawcze:

5.1 Proces wdrażania

- *Jak powinien przebiegać proces wdrażania narzędzia MOAW?*
- *Na jakie elementy należy zwrócić uwagę podczas efektywnego wdrożenia?*

5.2 Proces treningowy

- *Jak zmotywować uczestników zespołu?*
- *Jakich kompetencji wymaga zespół?*
- *Jakie elementy mogą wspierać lub utrudniać komunikację w zespole?*

5.3 Proces pilotażowy

- *Jakie wymagania sformułować dla oprogramowania wspomagającego?*
- *Jakich korzyści należy oczekiwać z wprowadzenia Bazy Wiedzy?*
- *Czy możliwe jest sformułowanie obiektywnych miar sukcesu dla MOAW?*

5.4 Proces uruchomienia i użytkowania

- *Jakie zagrożenia mogą wystąpić podczas uruchomienia i użytkowania MOAW i Bazy Wiedzy?*

5.1 Proces wdrażania nowego narzędzia zarządzania wiedzą

Wdrażanie metodyki (narzędzi zarządzania wiedzą) w organizacji jest niezwykle trudnym zadaniem, gdyż skierowane jest na zmiany organizacyjne, a te ze swej istoty są związane z dużym ryzykiem. W przedsiębiorstwach jest znanych wiele dobrych metodyk i narzędzi wspierających prace inżynierskie, podejmowanie decyzji etc., jednak ich użytkowanie jest zwykle ograniczone. Przykładem tego może być niezwykle silne narzędzie wspomagania decyzji – analiza FMEA procesu (ang. *failure mode effects analysis*), które choć jest niezbędne dla uzyskania lub utrzymania certyfikatu jakości ISO, nie jest jednak stosowane w codziennej praktyce [Wiśniak 02].

Jako przyczynę niepowodzeń projektów organizacyjnych wskazuje się uwarunkowania:

- obiektywne – niezależne od składu osobowego zespołów,
- subiektywne – bezpośrednio zależne od czynnika ludzkiego.

Wobec rozległości i różnorodności zagrożeń, dla sukcesu we wdrażaniu metodyki Modelowania Obiektowego dla Automatyzacji Wytwarzania (MOAW), konieczne jest wypracowanie odpowiedniej strategii wdrażania i późniejszego jej utrzymania.

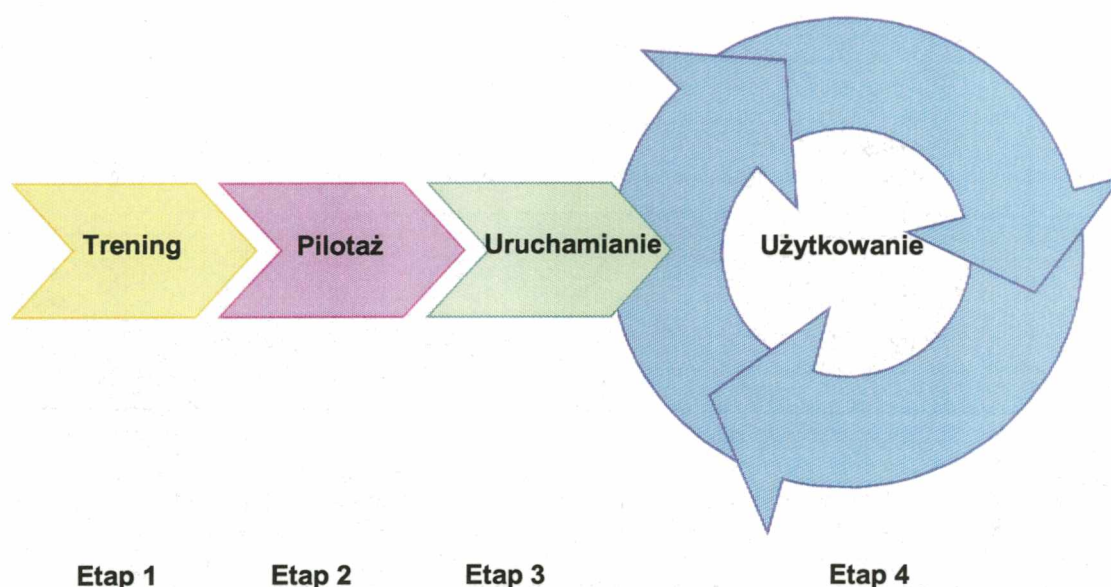
Istnieją dwie podstawowe strategie wprowadzania dużych projektów organizacyjnych:

- wdrożenie kompleksowe – obejmujące cały zakres pakietu,
- wdrożenie etapowe – wprowadzające poszczególne pakiety zmian w rozwinięciu czasowym.

Wdrożenie kompleksowe wymaga dojrzałości produktu jak również odpowiedniego przygotowania organizacji. Dojrzałość metodyki może być zapewniona wyłącznie poprzez jej weryfikację w poprzednich wdrożeniach. Przygotowanie organizacji musi być poprzedzone szczegółową analizą. W wyniku wdrożenia kompleksowego, należy oczekiwać spójnej metodyki oraz krótkiego czasu realizacji. Planowanie projektu wdrażania może być precyzyjne, zarówno w zakresie kosztów jak i terminów. Koszty takiego przebiegu realizacji są jednak zwykle wysokie, gdyż wymagają ogromnego nakładu pracy również po stronie zleceniodawcy, a niejednokrotnie wstrzymania procesu biznesowego. Zleceniodawca musi być również przygotowany na ogromne przeciążenie w początkowej fazie pracy, wynikające z weryfikacji zaproponowanych rozwiązań. Dlatego też wdrażanie kompleksowe można przyrównać do *reinżynieringu* przedsiębiorstwa, zwanego niekiedy „terapią wstrząsową”.

Natomiast **wdrożenie etapowe**, jest znacznie łagodniejsze, zarówno dla zleceniodawcy jak i wykonawcy. Dopuszcza ono wprowadzanie rozwiązań nowatorskich oraz daje czas organizacji na uczenie się i rozwój. Metodę etapową można określić jako „bliższą człowiekowi”, gdyż daje ona czas na uzyskanie jego akceptacji oraz skorzystanie z jego wiedzy merytorycznej. Ponieważ produkt wdrożenia powstaje jako wynik projektu, to znacznie trudniejsze jest tutaj precyzyjne oszacowanie terminów i kosztów. Dodatkowo szczególną uwagę należy zwrócić na zarządzanie zmianami, które stają się procesem towarzyszącym. Bez dobrze przygotowanej metodyki zarządzania zmianami i platformy wspomagającej, kolejne moduły funkcjonalne systemu mogą stać się niespójne.

Ponieważ proponowana w niniejszej pracy doktorskiej metodyka „Modelowania Obiektowo dla Automatyzacji Wytwarzania” MOAW jest rozwiązaniem nowatorskim wymagającym zmiany kultury pracy, to zaleca się wdrożenie etapowe jak przedstawiono na rys. 5.1.



Rys. 5.1 Cykl życia Modelowania obiektowego i UML dla automatyzacji wytwarzania [jr]

W planie wdrażania narzędzia MOAW wyróżniono 4 główne etapy:

Etap 1: Proces TRENINGOWY

Cel:

- przygotować zespół do efektywnej pracy,
- zapoznać uczestników z metodyką,
- przeprowadzić praktyczny przykład,
- uzyskać akceptację zespołu.

Etap 2: Proces PILOTAŻOWY

Cel:

- przygotować i udostępnić odpowiednie narzędzia wspomagające,
- przeprowadzić praktyczny przykład,
- uzyskać wyniki do oceny przedsięwzięcia i estymacji przyszłościowych wyników.

Etap 3 Proces URUCHAMIANIA SYTEMU (skalowania)

Cel:

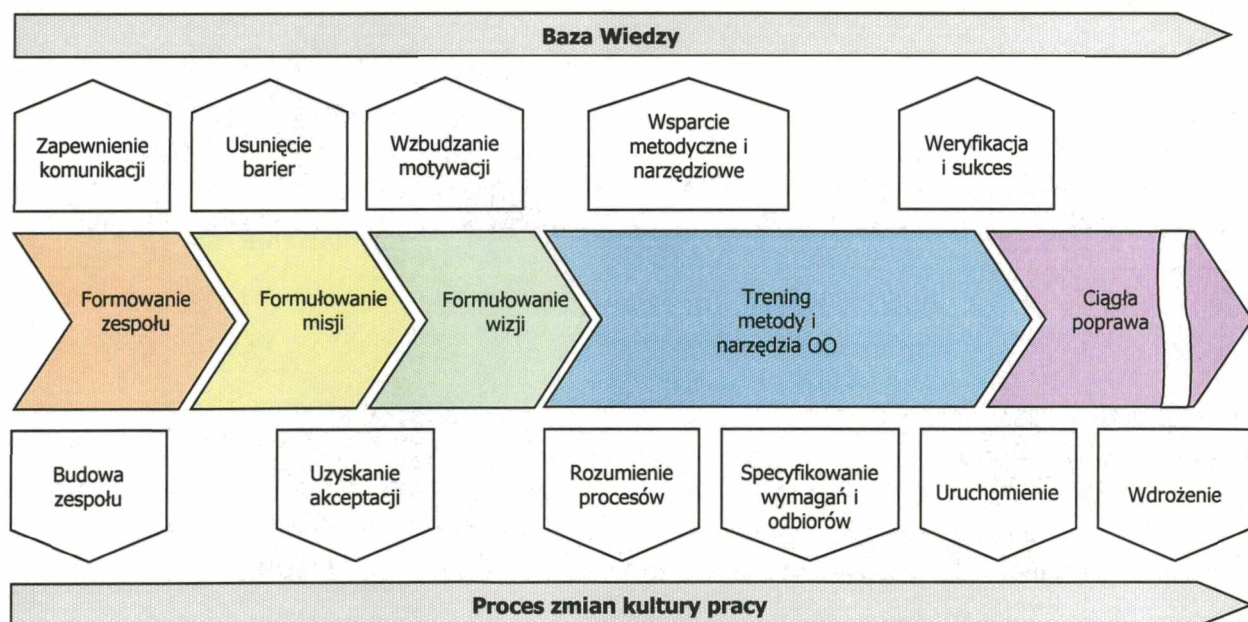
- napęlnić system danymi zgodnie z metodyką (inżynieria odwrotna),
- rozszerzyć zespół użytkowników.

Etap 4 Proces UŻYTKOWANIA narzędzia wiedzy

Cel:

- pielęgnować narzędzie,
- rozwijać narzędzie.

Podczas procesu wdrażania można wyróżnić kilka perspektyw jego oddziaływania. Na rys. 5.2 przedstawiono proces wdrożenia MOAW oraz jego oddziaływanie na tworzenie Bazy Wiedzy oraz proces zmian kultury pracy. Baza Wiedzy jest abstrakcyjnym pojęciem dla scharakteryzowania organizacji wspierającej tworzenie, komunikowanie i użytkowanie wiedzy oraz środki techniczne wspierające powyższe działania. W rozważanym przykładzie Baza Wiedzy będzie stanowiła wiedzę organizacyjną o wdrożonej i planowanej automatyzacji wytwarzania, metody i języki komunikacji wiedzy między specjalistami oraz narzędzia informatyczne wspierające proces archiwizowania informacji.



Rys. 5.2 Wyniki procesu wdrożenia MOAW [jr na bazie HP]

Wprowadzanie innowacyjnych metod pracy grupowej będzie wymagało zmiany kultury pracy. Zmiany takie wymagają odpowiedniego przygotowania pracowników i organizacji. Ich tempo wdrażania hamowane jest przez inercję zależną od aktualnej kultury pracy.

W kolejnych podrozdziałach przedstawione zostaną etapy wdrażania MOAW ze szczególnym zwróceniem uwagi na wybrane elementy procesu wdrażania.

5.2 Proces Treningowy MOAW

Ponieważ proces treningowy traktowany jest jako projekt, to dla zapewnienia jego dużej efektywności konieczne jest podporządkowanie go wymaganiom zarządzania projektem [PMBOK 00].

Nadrzędnym celem procesu treningowego jest zaznajomienie jego uczestników z metodyką obiektowej inżynierii odwrotnej, zapewnienie ich kompetencji i uzyskanie akceptacji dla prezentowanych rozwiązań. Aby trening taki odniósł oczekiwany skutek, tj. aby uczestnicy byli przygotowani do zastosowania metodyki w praktycznym użytkowaniu systemów automatyzacji wytwarzania, zwraca się uwagę na:

- motywację uczestników,
- efektywną pracę grupową.

5.2.1 Motywacja uczestników

Procesy motywacyjne to zespół procesów psychicznych i fizjologicznych określający stan ludzkich zachowań i ich zmian [pp web].

Ogólną charakterystykę procesu motywacyjnego określają następujące zjawiska:

- wzbudzanie energii,
- ukierunkowanie wysiłku na określony cel,
- selektywność uwagi w stosunku do odpowiednich bodźców – zwiększenie wrażliwości wobec bodźców istotnych, zmniejszenie w odniesieniu do nieistotnych,
- zorganizowanie pojedynczych reakcji w zintegrowany wzorzec bądź sekwencję,
- konsekwentne kontynuowanie ukształtowanej czynności, dopóki warunki, które je zapoczątkowały nie ulegną zmianie.

Motywacja uczestników, tj. wewnętrzna chęć zapoznania się z proponowanym rozwiązaniem MOAW i późniejsze jego użytkowanie jest podstawowym wymaganiem wysokiej efektywności procesu szkoleniowego.

Proces motywacyjny zależy od dwóch podstawowych założeń:

- wynik czynności musi zostać przez jednostkę oceniony jako potrzebny i pożądaný (użyteczność wyniku),
- dana osoba musi być przekonana, że pożądaný wynik można w danych warunkach osiągnąć z prawdopodobieństwem wyższym od zera (subiektywne prawdopodobieństwo osiągnięcia wyniku).

Wartość natężenia procesu motywacyjnego, czyli wielkość motywacji do osiągnięcia danego celu jest modyfikowana przez stopień użyteczności tego wyniku oraz spostrzeganego przez jednostkę prawdopodobieństwa osiągnięcia tego wyniku.

Zgodnie z powyższymi założeniami psychologii zdecydowano, aby proces motywacyjny dla uczestników treningu MOAW podkreślał użyteczność wyniku i możliwość jego osiągnięcia. Takie postawienie priorytetów dla reprezentantów przemysłu jest warunkiem koniecznym.

Czy uzasadnianie podjęcia tematu powinno bazować na wynikach przeprowadzonej wcześniej analizy heurystycznej procesu produkcyjnego, która wykazała problemy? Wyniki takie są najbardziej przekonujące dla podjęcia tematu, ale wyzwalone napięcie motywacyjne określane jest przez psychologię jako „ujemne”. Wynika z tego, że emocje typu „muszę” wyzwolą mniejsze działanie i będą nastawione na szybką realizację celów wynikających z analizy, tj. głównie usunięcie problemów, a nie na systemowe rozwiązywanie problemów, będące inwestycją długofalową.

Czy są dostępne również argumenty mogące wyzwolić napięcie motywacyjne w formie dodatniej? To znaczy, czy istnieje potrzeba lepszego niż dotychczas zaspokajania potrzeb? Taką formą mogłyby być przykłady referencyjne, najlepiej sprawdzone, tj. z innych przedsiębiorstw. Jednak, gdy proponowane rozwiązania wymagają dużej inwestycji badawczej, tj. są one zdefiniowane na poziomie wizji, zaś odbiorca nastawiony jest głównie na wymierne ekonomiczne wyniki, to wówczas argumenty na poziomie wizji nie odnoszą pożądanego znaczenia.

Drugim elementem procesu motywacyjnego jest prawdopodobieństwo osiągnięcia pożądanego wyniku. Jeśli człowiek jest świadomy uzyskania oczekiwanego wyniku wykonywanych czynności, to wówczas ten wynik nazywany jest celem. Oznacza to, że definicja celów powinna poprzedzać każde działanie. Cele te muszą jednak zapewniać możliwość weryfikacji zarówno jakościowej jak i ilościowej.

Cel treningu określony jako „przekazanie wiedzy na temat narzędzia MOAW”, choć realny i dający się zweryfikować, np. poprzez testy, nie odpowiada jednak pierwszemu założeniu procesu motywacyjnego – użyteczność wyniku. Jeśli owa użyteczność została zdefiniowana dla przyszłej i aktualnej poprawy stanu procesu, to osiągnięciem wyniku będzie zastosowanie treści treningu oraz uzyskanie efektów metody. Dlatego podczas treningu zwrócono szczególną uwagę na to, aby przeanalizować taki przykład, który pochodzi z rzeczywistego obiektu, tj. taki, który uczestnicy szkolenia już znają.

Jako ważne pytanie dla użyteczności wyniku zidentyfikowano problem „skalowalności rozwiązania”. Czy przedstawiony i rozwiązywany przykład jest reprezentatywny i da się przenieść na inne, docelowo wszystkie, elementy systemu? Jak rozszerzanie problemu wpłynie na kompleksowość? W odpowiedzi na powyższe pytania przedstawiono w poniższym podrozdziale koncepcję narzędzia wspierającego, zarówno merytoryczną stronę zagadnienia, jak i organizacyjną. Narzędzie to w wersji rozwiniętej, wraz z metodyką jego użytkowania nazwano Bazą Wiedzy.

Prawdopodobieństwo osiągnięcia wyniku obejmuje również podstawowe mierzalne ograniczenia: czas i koszt. Jaki jest horyzont czasowy szkolenia a następnie wprowadzania rozwiązania i jaki będzie szacunkowy koszt wdrożenia rozwiązań komputerowych?

Ocenę prawdopodobieństwa osiągnięcia pożądanego wyniku modyfikuje również poprzednie doświadczenie. Jeśli zmiany wprowadzone poprzednio, a także zastosowane metody i narzędzia nie przyniosły sukcesu, to nastawienie do kolejnych działań nie będzie pozytywne. Przykładowo certyfikat jakości ISO w wielu firmach nie jest narzędziem codziennej pracy nad procesem, a „zaszczytem” wyłącznie formalnym. Zbyt ortodoksyjny formalizm i technokratyzm bez rozumienia filozofii w przypadku takich paradygmatów jak: podejście obiektowo-zorientowane, Lean Manufacturing, KAIZEN, 6Sigma, etc. przesłania prawdziwy ich cel tzn. poprawę stanu.

Jednak w sytuacji, gdy użyteczność wyniku zostanie oceniona jako wyjątkowo duża, to należy spodziewać się tendencji do zawyżania, co nazywane jest myśleniem życzeniowym. Praktycznym przejawem takiego stanu będzie chęć zastosowania metody MOAW jako „*panaceum*” na wszystkie obecne i przyszłe problemy.

Motywacja

- pokazać użyteczność wyniku:
 - dla problemów stale pojawiających się lub
 - problemów ciągle nierozwiązanych;
- przekonać o możliwości wdrożenia.

Psychologia motywacji zwraca również uwagę na redukowanie napięcia motywacyjnego przez określony obiekt, stan rzeczy, osobę, sytuację lub warunki. Wówczas obiekt ten ma określoną wartość gratyfikacyjną. Wartość ta może być wielkością obiektywną, ogólnie uznaną lub taką, którą można w mniejszym lub większym stopniu przewidzieć i określić, tj. antycypować wartość gratyfikacyjną.

Ponieważ podstawą uzyskania miarodajnych wyników metodyki jest praca grupowa, to proces motywacyjny musi być rozwijany dla całego zespołu.

Brak zaangażowania przełożonych, którzy przyzwalają wprowadzić na proces treningowy traktując go jednak bardziej jako eksperyment, a nie inwestycję biznesową, zredukuje napięcie motywacyjne uczestników. Prowadzony zaś proces treningowy będzie się przedłużał, powodując stopniowy wzrost frustracji i niezadowolenia wśród przyszłych użytkowników metodyki.

5.2.2 Praca grupowa – komunikacja i kompetencje

Efektywność treningu, rozumienie prezentowanych treści, a więc i strategia dydaktyczna oraz jej realizacja będą decydowały o podtrzymaniu zainicjowanej motywacji. Celem procesu dydaktycznego jest przekazanie wiedzy przez trenerów. Jak to zostało już pokazane w rozdziale 4.1.1, proces ten może być efektywnie wspierany przez szkice i języki graficzne.

Rozwiązania prezentowane podczas szkolenia muszą być dostosowane do poziomu dojrzałości uczestników, który powinien wzrastać. Najbardziej „wrażliwym” elementem kontaktu trener-uczestnik jest jednak komunikacja. Jak wykazały obserwacje, podczas komunikacji z przedstawicielami przemysłu, wykorzystywane słownictwo – symbole, mogą „zerwać” relację trener-uczestnik. Uczestnik musi zostać przygotowany do rozumienia, a później stosowania pewnych abstrakcji pojęciowych takich jak np.: model, modelowanie, paradygmat, integracja, obiekt, klasa, relacja, wiedza, baza wiedzy, itp. W przeciwnym razie trener „nada” uczestnik zaś „nie odbiera” informacji, lub komunikacja generuje zupełnie niezamierzony, a wręcz nieprzewidywalny model pojęciowy. Dlatego też strategia komunikacji wymaga „sprzężenia zwrotnego”.

Mnogość zagadnień związanych z realizacją rozległych zadań zwraca uwagę na pracę grupową. Zastosowanie grup w organizacji uznawane jest za najwyższy priorytet dla liderów biznesowych. [Tannenbaum 96]. Dzięki podejściu grupowemu zwraca się uwagę na lepsze identyfikowanie rzeczywistych problemów, budowanie konsensusu, balansowanie obciążeń i zapewnianie ciągłości. [Ettinger 01]. Zewnętrzni konsultanci potrafią „świeżym okiem” spojrzeć na problem, zaś dzięki reprezentantom wewnętrznym z różnych dziedzin, prowadzona analiza lepiej „wnika” w istotę problemu. Proponowane rozwiązania lepiej są dostosowane do warunków, ograniczeń i oczekiwań członków zespołu, mimo, że nie są one w pełni sformalizowane. Zespół powinien na zasadzie konsensusu wewnętrznego, jak również w uzgodnieniu z wykonawcą, ustalać cele i zakres prowadzonych prac. Realizacja musi uwzględniać aktualne obciążenie członków zespołu, wykorzystując delegowanie zadań. Interdyscyplinarność zespołu i różne doświadczenia powinny szczególnie zaowocować podczas zadań wymagających kreatywności. Zaś wymiana doświadczeń, wspólne analizowanie problemów i poszukiwanie rozwiązań są naturalnym sposobem zarządzania wiedzą poprzez jej wymianę między członkami grupy.

W celu zwiększenia efektywności grupy zaleca się uwzględnienie predyspozycji grupowych, lub przynajmniej preferencji ról grupowych podczas selekcji. Tannenbaum i inni zwraca uwagę na trzy aspekty wpływające na wydajność grupy: identyfikacja poziomu umiejętności, wpływ na niejednorodność oraz identyfikacja predyspozycji grupowych. Dla określenia wymaganego profilu uczestników grupy odpowiedzialnej za użytkowanie, utrzymanie i rozwój narzędzia wiedzy MOAW, wygodnie jest posługiwać się rolami, jakie odgrywają poszczególni użytkownicy względem systemu. Role te określają zakresy kompetencji, czyli zakres posiadanych uprawnień, pełnomocnictw, odpowiedzialności oraz zakres posiadanej wiedzy. Kompetencje merytoryczne, doświadczenie oraz kompetencje społeczne określają profil kompetencyjny. Kompetencje określone są na podstawie centrów oceny (ang. *assessment centres*), wywiadów, testów 180°/360°/540°, itp.

Prace Belbina (1981) zidentyfikowały osiem następujących ról, które przyjmują ludzie pracujący w zespole:

<i>Plant</i>	Wizjoner
<i>Resource Investigator</i>	Poszukiwacz zasobów
<i>Shaper</i>	Wykonawca
<i>Completer Finisher</i>	Wykończający
<i>Monitor Evaluator,</i>	Rewizor
<i>Teamworker</i>	Członek zespołu
<i>Company Worker</i>	Pracownik firmy
<i>Chairperson</i>	Koordynator

Zgodnie z teorią Belbina [Belbin 96] każdym zespołem odnoszącym sukcesy powinien występować zrównoważony podział ról. Zaś na podstawie teorii typów Jungiana opracowano narzędzia pozwalające ocenić indywidualnie preferowaną rolę. Podobnie do instrumentów Belbina dostępne są wyniki prac Margerisona i McCann [Burton 94], identyfikujące następujące role:

<i>Reporter Adviser</i>	Reporter Doradca
<i>Creator Innovator</i>	Twórca Innowator
<i>Explorer Promoter</i>	Badacz Rzecznik
<i>Assessor Developer</i>	Rzeczoznawca Architekt
<i>Thruster Organiser</i>	Lokomotywa Organizator
<i>Concluder Producer</i>	Wnioskodawca Producent
<i>Controller Inspector</i>	Kontroler Inspektor
<i>Upholder Maintainer</i>	Serwisant

Do określenia powyższych ról dostępne są testy obejmujące 60 zapytań z wymuszonymi odpowiedziami z listy. Preferowane i spokrewnione role, wynikające z odpowiedzi badanych są grupowane w cztery klasy odnoszące się do następujących zapytań:

- Jakie są preferencje do relacji z innymi?
- Jak wykorzystywana jest informacja?
- Jak podejmowana jest decyzja?
- Jakie są predyspozycje do samoorganizacji i organizacji innych?

Proces rozwoju zespołu dzieli się zwykle na cztery następujące etapy [Tłuchowska 02]:

- **Formowanie zespołu** – gdy członkowie zespołu definiują swoje miejsce i pełnione role. Jednocześnie planowany jest przebieg pracy i analizowany problem. Podczas tej

fazy szczególnie przydatne są techniki i narzędzia stymulujące kreatywność i wymianę wiedzy.

- **Docieranie zespołu** – gdy wymagania i wykonalność zadań projektowych są kwestionowane a zasady i metody pracy poddane są osądowi. Weryfikowane są kompetencje i akceptacja. Powstaje wiele alternatywnych propozycji rozwiązań, które wymagają udokumentowania i zarządzania.
- **Wytwarzanie norm** – faza podczas której dochodzi do konsensusu i ustalenia zakresów zadań poszczególnych członków zespołów i zadań projektowych. Faza ta powinna być wspierana metodami i narzędziami analizy, podejmowania decyzji i komunikacji.
- **Wykonanie** – to ostatnia faza, najbardziej produktywna, gdy wdraża się procedury i reguły ustalone w fazach poprzednich. Jest to faza właściwego zarządzania projektem i procesem produkcyjnym. Praktyczna ocena dynamiki zespołu wskazuje na okres 4-8 tygodni potrzebnych na zawiązanie i zgranie zespołu. [Ettinger 01]

Czy zespół zestawiony na podstawie kompetencji, również społecznych, odniesie sukces? W literaturze psychologii stosunków międzyludzkich, np.: [Welenc 02], znajdujemy porównanie „grupy” i jej przeciwieństwa - „roju”. Rój jest nisko lub w ogóle niezhierarchizowany, nie ma podziału funkcji, każda jednostka działa dla siebie. Siła roju tkwi w ilości, a siła grupy wynika z organizacji i podziału kompetencji. Kultura organizacyjna, tj. normy i wartości organizacyjne, zależą głównie od sposobu zarządzania. Ten zaś zależy od strategii firmy, jej wizji i założeń politycznych. Stąd podkreślanie „myślenia strategicznego” dla grupy oraz potrzeba wyodrębnienia lidera, który łatwo przejmuje kontrolę nad motywacją. Dobrze zatem, gdy lider pełni rolę kierownika projektu, moderatora lub trenera (ang. *coach*) [Popadiuk 02].

5.2.3 Przebieg treningu

Postawienie problemu

W pierwszym kroku przebiegu treningowego ważne jest wspólne zdefiniowanie sukcesu. Oznacza to, że ustalone zostają cele oraz jakościowe i ilościowe miary dla weryfikacji ich osiągnięcia. Uczestnicy powinni zatem poznać zakres tematyczny, czasowy oraz własny wkład, wymagany dla osiągnięcia sukcesu. Wprawdzie „pedagogika odkrywczą”, która rozwija temat od problemu, wspierając go niezbędną teorią, potwierdza lepsze rozumienie i pamiętanie treści, ale nie radzi sobie z utrzymaniem motywacji. Podobna metoda „*On-The-Job Training*” OJT (nauka przez pracę) buduje motywację na potrzeby rozwiązania problemu.

Dla treningu metodycznego, obejmującego konieczność zmiany sposobu myślenia wybrano metodę klasyczną – projektową.

Uczestnicy szkolenia mają przygotować się do przeanalizowania linii technologicznej oraz zaproponowania zmian w celu poprawy jej działania. Ponieważ problem jest rozległy, gdyż obejmuje ponad 40 stanowisk ręcznych i automatycznych, a procesowi produkcyjnemu towarzyszą zmiany polegające na modyfikacji i dostawianiu nowych stanowisk, to potrzebne jest „uporządkowane” podejście do problemu. Uporządkowanie takie powinno polegać na zastosowaniu wybranych procedur, technik, i metodologii, w celu ujednolicenia wytworzonej dokumentacji. Ponieważ dokumentacja ta powinna być łatwa do późniejszego uaktualniania, w drugim etapie projektu zostanie zwrócona uwaga na komputerowe narzędzia wspomagające.

Sposoby komunikacji

Już podczas pierwszych dyskusji okazuje się, że interdyscyplinarność uczestników stanowi barierę komunikacyjną. Różna wiedza specjalistyczna, tj. definicje pojęć, rozwiązywane problemy a nawet żargon powodują, że dyskutanci nie rozumieją się. Można zaobserwować nawet technikę obrony polegającą na „opuszczaniu” fragmentów wypowiedzi. Typowym przykładem jest postrzeganie świata poprzez pryzmat „bitów i bajtów”, co nie może być zrozumiałe dla technologa. Podobnie jak zbyt abstrakcyjne patrzenie na problem i jego rozwiązanie prezentowane przez naukowców.

Rozwiązaniem może być wprowadzenie piramidy poziomów abstrakcji problemu. Wyróżniając trzy poziomy: poziom realizacji, poziom logiczny i poziom koncepcji ograniczamy zakres rozważań.

Ważne jest, aby pokazać, że poziom realizacji, obejmujący problemy codziennej pracy, jest inny dla każdego z uczestników. W przypadku linii montażowej, dla automatyka będą to „bity i bajty”, czujniki i elementy wykonawcze, dla technologa operacje montażowe, a dla inżyniera zapewnienia jakości błędy montażowe. Nie ma zatem możliwości ale i potrzeby, żeby takie lokalne problemy usiłować rozwiązywać wspólnie.

Wszystkie te lokalne problemy podporządkowane są jednak pewnym konkretnym przebiegom, algorytmom i zachowaniom. To poziom logiczny procesu definiuje struktury i zachowania, których nie widać. Dlatego też do opisu elementów tego poziomu potrzebne są pewne narzędzia abstrakcji, takie jak np.: wcześniej przedstawione modelowanie obiektowe (MOAW). Poziom ten pokazuje wyraźne związki między różnymi widokami zadaniowymi systemu produkcyjnego i wymaga wzajemnego zrozumienia i konsensusu.

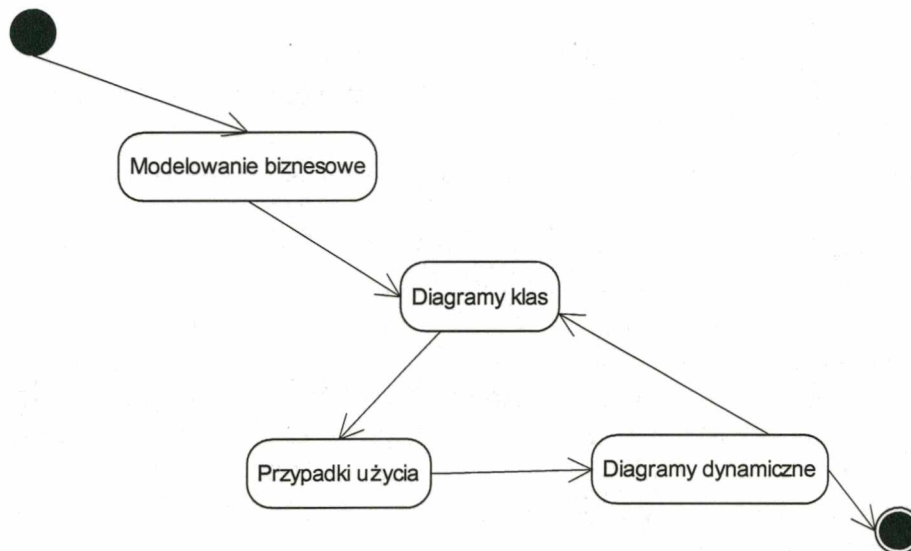
Czy typowa interdyscyplinarna grupa projektowa potrafi pracować efektywnie na poziomie logicznym? Jak wykazano wcześniej zależy to od umiejętności ograniczania zakresu problematyki oraz umiejętności komunikowania się w zakresie pojęć abstrakcyjnych.

Czego dotyczy poziom najwyższy, nazwany poziomem koncepcji? Czy nowoczesne, zaawansowane rozwiązania techniczne potrzebują jeszcze czegoś bardziej abstrakcyjnego jak algorytmy działania, czy modele danych?

Koncepcje CIM'u lat 70 upadły nie z powodu ograniczeń mocy obliczeniowej ówczesnych komputerów. Fabryki bezzałogowe nie powstają. Jeśli zatem człowiek pozostaje „elementem” systemu wytwórczego i ma być „zintegrowany” z procesem nadal produkcyjnym i informatycznym, to konieczne jest zwrócenie uwagi również na aspekty psychologiczne, socjologiczne a nawet filozoficzne i etyczne. Te zaś stawiają pryncypialne pytanie „człowiek jako przedłużenie ramienia robota, czy robot jako przedłużenie ramienia człowieka?” [Eversheim, et al. 02]. Poszukiwanie odpowiedzi na powyższe pytanie znacznie wykracza poza zakres tej pracy. Ważne jest natomiast, że algorytmy i struktury danych są wirtualną realizacją koncepcji, te zaś dla uzyskania dużej efektywności, muszą być zaakceptowane i udokumentowane, gdyż stanowią podstawę działań operacyjnych. Z poziomu algorytmów nie widać strategii systemu, a krytyczną sytuacją jest, gdy po pewnym okresie użytkowania i modyfikacji pojawia się brak spójności między produktami automatyzacji, co zostało szczegółowo omówione w rozdziale 3.

W związku powyższym, obszarem współpracy interdyscyplinarnych grup odpowiedzialnych za utrzymanie i rozwój systemów produkcyjnych są poziomy logiczny i koncepcyjny. Dla efektywnego „poruszania” się w nich niezbędne są jednak odpowiednie techniki komunikacji. W powyższej pracy zaproponowano zastosowanie modelowania obiektowego i UML dla automatyzacji wytwarzania, co przedstawiono w rozdziałach 3 i 4.

Schemat wprowadzenia kompetencji w zakresie modelowania obiektowego i UML tzn. MOAW przedstawiono na diagramie rys. 5.3.

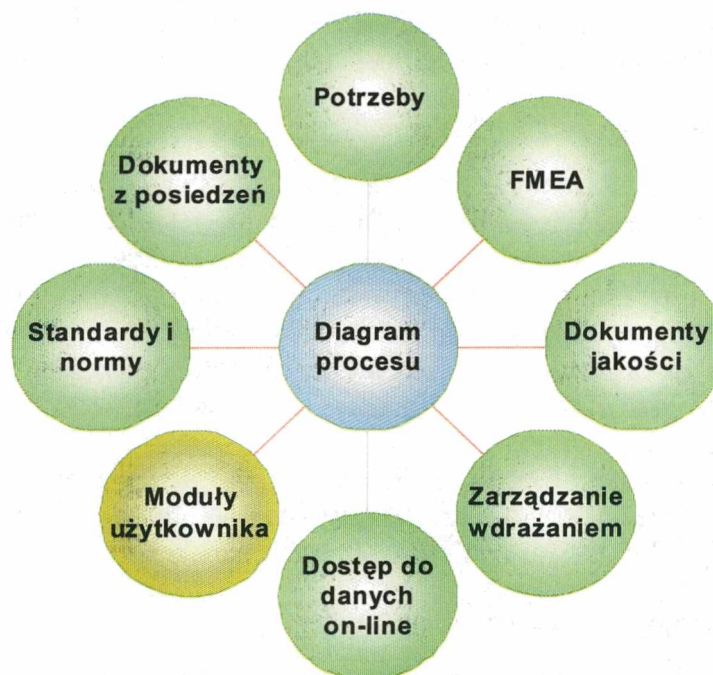


Rys. 5.3 Diagram stanów dla wprowadzania kompetencji MOAW [jr]

5.3 Pilotażowe wdrożenie platformy wspomagającej

5.3.1 Platforma wspomagająca

Praktyczne wykorzystanie modelowania obiektowego i UML dla automatyzacji procesu produkcyjnego wymaga wsparcia programistycznego. W przeciwnym razie aktualizacja diagramów i zapewnienie ich spójności, przy użytkowaniu przez wieloosobowy zespół okaże się barierą. Bez oprogramowania wspierającego oraz zaangażowania całego zespołu odpowiedzialnego za utrzymanie i rozwój systemu produkcyjnego, grupa wykorzystująca MOAW „utonie” w dokumentacji, która i tak będzie miała luki i błędy. Włożony trud i zaangażowany czas nie przyniesie spodziewanych wyników.



Rys. 5.4 Modułowa struktura platformy Bazy Wiedzy [jr]

Oprogramowanie takie powinno wspierać cały cykl życia systemu produkcyjnego:

- ułatwiać zarządzanie informacjami o stanie linii,
- pomagać przy podejmowaniu decyzji o wyborze koniecznych zmian,
- ułatwiać śledzenie zmian na linii oraz
- wspierać odbiory nowych maszyn i urządzeń oraz ich modyfikacje.

Dzięki codziennej pracy i archiwizowaniu działań oraz ich wyników platforma ta stanie się podstawą dla Procesu Ciągłej Poprawy. System tak szerokiego wsparcia informatycznego dla utrzymania systemu produkcyjnego i jego rozwoju oraz wsparcia pracy grupowej został nazwany Bazą Wiedzy.

Na rys. 5.4 przedstawiono modułową strukturę platformy Bazy Wiedzy. Jako jądro tego systemu wyróżniony został diagram (model) procesu. To on jest rozszerzany przez specyficzne komponenty takie jak: analiza FMEA, dokumenty jakościowe, dokumenty projektowe, dane produkcyjne, biblioteki i wzorce, raporty ze spotkań zespołów roboczych oraz specyfikacje wymagań.

5.3.2 Spodziewane rezultaty z wdrożenia Bazy Wiedzy

Dokumentacja techniczna stanu systemu wytwórczego

Dokumentacja na temat stanu systemu wytwórczego obejmuje bardzo wiele różnorodnych informacji, które nie są standaryzowane ani pod względem formy, zawartości, ani też formatu przechowywania. Ponieważ serwery plików (ang. *file servers*) umożliwiają jedynie zarządzanie plikami, poprzez ich katalogowanie i określanie prawa dostępu, to związek między informacjami zawartymi w tych plikach, a szczególnie poruszanie się po strukturze katalogów, ma niewiele wspólnego z rzeczywistym obiektem – systemem wytwórczym. Dlatego też jądrem systemu zarządzania dokumentacją o systemie musi być jego struktura. Dodatkowo niezbędne jest zdefiniowanie odpowiednich widoków z informacjami dla określonych użytkowników (technologia, jakość, planowanie, serwis, etc.) oraz formularzy do wprowadzania informacji, np.: o zmianach. Baza Wiedzy zapewnia wielodostęp do danych, kierując się ich zawartością merytoryczną, a nie technologią informatyczną.

Redukcja nadmiarowości informacji

Problem przeciążenia informacyjnego oraz tzw. „szum informacyjny” są widoczne w codziennym życiu, a szczególnie w pracy z sieciowymi systemami informacyjnymi. I tak np. „wyszukiwarka” internetowa www.google.com, dla hasła „baza wiedzy” znajduje 66000 adresów stron. Odfiltrowanie informacji i dotarcie do poszukiwanej staje się istotnym problemem. Podobnym problemem jest ocena jakości informacji, co powoduje, że narzędzia informatyczne do wyszukiwania informacji nie wykorzystują takiego kryterium. Wielodostęp, decydujący o potęgze rozwiązań sieciowych, jest równocześnie wyzwaniem dla konsystencji i unikania redundancji.

Niestety baza danych nie jest rozwiązaniem powyższych problemów, a jedynie platformą do implementacji algorytmów. Stąd potrzeba struktury i organizacji informacji o systemie produkcyjnym i jego odwzorowania w systemie informatycznym.

Automatyczne generowanie dokumentów jakości

Dokumenty jakości VDA, ISO, QS są zdefiniowanymi widokami wybranych informacji o stanie systemu. Często niestety, dokumenty te nie są akceptowane i wykorzystywane przez „nie-jakościowców”. Każdy z użytkowników tworzy więc własne, tracąc czas i wprowadzając

własne standardy. Brak mechanizmów synchronizacji dokumentów sprawia, że z czasem prezentują one odmienny, niespójny stan tej samej linii wytwórczej. Dzięki Bazie Wiedzy informacja przechowywana jest centralnie, zaś dokumenty jakości stanowią jedynie widoki dostępne do czytania i zapisu informacji o systemie.

Poprawa równomierności wiedzy o systemie produkcyjnym wśród pracowników

Wiedza o rozwiązaniach technicznych zastosowanych w linii nie jest równomierna zarówno wśród kierownictwa jak i pracowników. Pracownicy linii bardzo zręcznie wykonują typowe sekwencje obsługi. Zaś specjaliści znają problemy jedynie ze swoich dziedzin. Ponieważ linia produkcyjna to integracja wielu różnych rozwiązań i wymagań, to problemy powstają najczęściej na „styku” – interfejsach. Im lepsze rozumienie przez specjalistów otoczenia swojej dziedziny, tym lepsza jest komunikacja i rozwiązywanie problemów. Udostępnienie pełnej dokumentacji systemu sprzyja procesowi ciągłego samokształcenia nie tylko wśród kadry kierowniczej.

Konsultacja błędów i rozwiązań w grupie

Dzięki udostępnieniu pełnej informacji o systemie, wcześniejsze zauważenie zagrożeń i błędów jest łatwiejsze. Grupa kierująca linią ma możliwość przedyskutowania zauważonego problemu przez „pryzmaty” specjalistów z różnych dziedzin. W ten sposób ważne problemy interdyscyplinarne poddawane są pod dyskusję.

Lepsze identyfikowanie skutków i przyczyn błędów (drzewo błędów),

Dzięki grupowemu dbaniu o proces technologiczny, jak również o jego dokumentację, zapewniane są pierwsze trzy kroki do analizy procesu FMEA. Analiza FMEA dla procesu staje się narzędziem codziennej pracy. Dzięki niej informacje o nowych błędach rozszerzają informacje już zgromadzone. Drzewa błędów, więc łańcuchy skutkowo-przyczynowe pozwalają dotrzeć do rzeczywistej przyczyna problemów. W ten sposób leczy się przyczyny a nie objawy.

Lepsza ocena ryzyka błędów

Dzięki informacji archiwalnej mamy miarę porównawczą na temat występowania, dotkliwości i wykrywalności błędów występujących podczas użytkowania linii montażowej. Ponieważ miary te są porównywane z wartościami archiwalnymi oraz konsultowane w grupie interdyscyplinarnej, to ich jakość jest lepsza. W przeciwnym wypadku, gdy ocena współczynników powierzona jest pojedynczej osobie, bez możliwości porównania i danymi archiwalnymi, to uzyskane wyniki nie są obiektywne i stąd też wynika ich akceptacja niska przez samego autora FMEA.

Poprawa efektywności doboru rozwiązań

Jakie zmiany w systemie są naprawdę istotne? Jaka jest relacja kosztów do efektu zmniejszenia błędów dla planowanej modernizacji?

Powyższe pytania muszą znaleźć odpowiedź przy wszystkich inwestycjach na linii produkcyjnej. Jednak uzyskiwane obiektywnej odpowiedzi wymaga dostępu do danych. Istotną miarą uzyskiwaną dzięki analizie FMEA jest współczynnik LPR (Liczba Potencjalnego Ryzyka). Współczynnik LPR ułatwia podejmowanie decyzji o tym, jakie problemy są obiektywnie istotne, a potrzeba ich rozwiązania nie jest tylko wynikiem emocji. Ustanowienie miar ważności dla szeregu współczynników związanych z zarejestrowanymi błędami ułatwia analiza Pareto lub metoda ABC.

Wsparcie dla Procesu Ciągłego Doskonalenia

Każda firma akceptuje zazwyczaj wymóg a nawet potrzebę ciągłego doskonalenia. Brak systematyczności uniemożliwia jednak wprowadzenie mechanizmu inkrementalnej poprawy. Dzięki archiwizowaniu i porządkowaniu informacji możliwe jest śledzenie zmian. Ponadto „dobre” zmiany mogą być propagowane na pozostałe wystąpienia w zautomatyzowanym systemie wytwórczym.

Wdrożenie pracowników w proces (filozofię) zarządzania wiedzą

Osoby odpowiedzialne za procesy, produkty czy ludzi zwykle zdają sobie sprawę ze znaczenia wiedzy. Coraz częściej słyszymy o problemie zarządzania wiedzą, coraz częściej występują także obawy wynikające z zastąpienia wiedzy ludzkiej przez inteligentne komputery. Bardzo podobnie gdy przed laty, gdy mechanizacja i automatyzacja była upatrywana jako źródło zła. Umiejętnie budowane systemy wytwórcze, w których maszyny wspierają pracę ludzką, nie stanowią zagrożenia dla człowieka i motywują go do lepszej wydajności i jakości pracy. Proces wdrażania systemów informatycznych, wspierających pracę ludzką, jest jednak zależny od odpowiedniego przygotowania. Wprowadzenie takich systemów wymaga zmiany kultury pracy, mentalności, a to, jak pokazuje praktyka jest najtrudniejsze. Pracownicy, którzy przekonają się do zalet informatycznych systemów wiedzy i będą je rozumieli, to równocześnie będą je akceptowali i współpracowali przy ich rozwoju.

Wzrost motywacji

Elementem motywacji pracownika jest odpowiednie dopasowanie poziomu wymagań do ludzkich możliwości. Gdy kompleksowość problemów przytłacza i następuje utrata panowania nad problemami, to zostaje podważony fundamentalny wymóg bezpieczeństwa człowieka i techniki. Dlatego system wspierający pracę ludzką zyska jego akceptację. Ważne jest jednak, aby nie był to system techniczny, lecz organizacyjno-społeczny wsparty techniką. Rola człowieka sprowadzona do przedmiotowej nie będzie wyzwalalała jego zaangażowania i kreatywności.

Poprawa efektywności pracowników i innowacja

Gdy Baza Wiedzy „uzbroi” pracowników w informacje i narzędzia, to należy oczekiwać wzrostu efektywności pracy. Szersze spojrzenie na system, interdyscyplinarność i synergia powinny wyzwolić także większą kreatywność pracowników w zgłaszaniu i wprowadzaniu zmian i innowacji.

Międzynarodowość

Internacjonalizm może przynosić bardzo ewidentne korzyści. Niekiedy, jak w firmach z kapitałem zagranicznym, jest on konieczny. Dlatego też jego wspieranie i rozwój nie mogą ograniczyć się wyłącznie do nauki języków obcych. Oprogramowanie Bazy Wiedzy wspiera międzynarodowość, tj.: zarówno obsługa programów jak i treści dokumentów mogą być prezentowane w językach obcych. Specjalny moduł programowy wspiera synchronizację między wersjami językowymi dokumentów, pokazując nowe wystąpienia bez tłumaczeń oraz wprowadzone poprawki, które nie zostały zatwierdzone językowo.

Praca grupowa - mniejsze uzależnienie od pojedynczego pracownika

Ponieważ podstawą Bazy Wiedzy jest praca zespołowa, to działanie i rozwój systemu nie jest podporządkowany jednostce, lecz zespołowi. Zespół ma jednak określoną odpowiedzialność, która nie może być rozmywana. Ważne jest jednak, aby informacja zachowywała prawa

intelektualne, w przeciwnym razie wiedza anonimowa nie jest wsparta autorytetem. Dla efektywności grupa potrzebuje koordynatora lub moderatora.

Fantom zwiększający bezpieczeństwo systemu

Dzięki Bazie Wiedzy linia produkcyjna zostaje odwzorowana wirtualnie. W ten sposób wszystkie zmiany, przed wdrożeniem do rzeczywistego systemu, mogą być weryfikowane i konsultowane przez różne widoki wymagań. Unikamy w ten sposób wdrożeń pochopnych, źle zintegrowanych z całością, a nawet zakłócających działanie innych systemów. Rolę weryfikacji zmian spełnia bowiem Baza Wiedzy a nie „żywy organizm”.

5.3.3 Narzędzia wspierające MOAW

Pojęcie „Baza Wiedzy” nie jest jeszcze obecnie jednoznacznie zdefiniowane. Dlatego też jest ono często nadużywane w celach marketingowych dla podniesienia wartości produktów. Baza Wiedzy nie istnieje jako produkt komercyjny. Na podstawie produktów software’owych pełniących rolę platformy, gromadzone są i przetwarzane informacje. Ponieważ informacje te są współdzielone, to konieczne są procedury ich wprowadzania i użytkowania, jak również procedury zapewniające wewnętrzną konsystencję. Podstawą bazy wiedzy są przestrzenie wielowymiarowe tworzone z danych i relacji między nimi. Do nawigacji i użytkowania systemu Bazy Wiedzy niezbędne są modele lub widoki. Jeśli każdy z działów firmy wprowadza własny opis systemu produkcyjnego, to nie możemy mówić wówczas o Bazie Wiedzy. Każdy z działów ma prawo i powinien rozszerzać istniejącą strukturę procesów technologicznych i informacyjnych. Pojęcie Bazy Wiedzy, prócz wymogu abstrakcji, wzbudza również obawy społeczne. „Czy po wprowadzeniu swojej wiedzy do systemu komputerowego, bezpieczeństwo i rola pracownika nie będą zagrożone?” Dlatego właśnie Baza Wiedzy jest platformą przede wszystkim organizacyjno-społeczną a nie techniczną. Jest to jednocześnie procesem, którego celem nie jest uzyskanie produktu, lecz zapewnienie ciągłego dostępu do usług i informacji.

Modelowanie

Diagramy procesu mogą być szkicowane na papierze (tablicy), jednak wtedy ich przetwarzanie, tj. modyfikacja, uaktualnianie i archiwowanie są utrudnione, szczególnie wówczas, gdy diagram obejmuje rozległy proces. W takim przypadku zaleca się stosowanie oprogramowania graficznego lub modelowania wspomagającego diagramowanie.

Ponieważ opracowane diagramy wymagają uaktualniania przez wszystkie role interdyscyplinarnego zespołu odpowiadającego za system produkcyjny, to oprogramowanie wspomagające powinno oferować funkcjonalność wielodostępu.

Tab.5.1 Zestawienie typów oprogramowania wspierającego modelowanie

Typ oprogramowania	Wady zalety	Przykład rozwiązania komercyjnego
Wspierające szkicowanie	- brak biblioteki elementów	MS Word, Star Office
Modelowanie (biblioteka elementów)	+ biblioteka elementów, + możliwość pracy z płaszczyznami (Layers)	VISIO, DrawConcept, Objectteering
Modelowanie i symulacja	+ to, co typ Modelowanie + możliwość symulacji i analizy np: wąskich gardeł, utylizacji, itp - drogie	iGrafix, ProModeller Rhapsody, Rational Rose

Wymogu współdzielenia nie spełnia oprogramowanie zestawione w tabeli 5.1 Dlatego też należy poszukiwać systemów do modelowania procesu opartych na bazie danych i umożliwiających wielodostęp.

5.4 Uruchomienie i użytkowanie MOAW i Bazy Wiedzy

5.4.1 Akceptacja czy jakość?

Czym jest sukces dla podmiotu gospodarczego? Jeszcze niedawno sukcesem było „wyprodukować”, później „sprzedać” a teraz to coraz najczęściej czytamy, że „pomnożyć pieniądze” – a więc osiągnąć zysk. Maksymalizacja przychodu, minimalizacja kosztów i zwiększenie szybkości obrotu stały się głównymi miarami sukcesu biznesowego.

Dla projektu jako przedsięwzięcia technicznego jako kryteria sukcesu przyjmuje się: czas realizacji, koszty i zakres przyjętych zadań. Obiektywne spełnienie powyższych kryteriów nie jest jednak równoznaczne z sukcesem.

A może miarą sukcesu w przedsięwzięciach technicznych jest poziom subiektywnej akceptacji jego uczestników [Cierpisz 02]? Współczesna psychologia zakłada, że nie istnieje żadna obiektywna prawda. Jeśli tak, to założenie, że wszystko jest subiektywne będzie skuteczniejszą miarą sukcesu. Zatem postrzeganie potrzeb i poziom akceptacji stają się podstawą także postrzegania projektu. Jeśli odniesiemy to do realiów projektowych, do różnych, często niespójnych wymagań i oczekiwań, zmieniających się w czasie – to żaden sztywny model postępowania nie daje gwarancji sukcesu. Oczywiście takie ryzyko i elastyczność nie są do zaakceptowania z perspektywy biznesowej a szczególnie ekonomicznej, stąd poszukiwanie formalnych ustaleń jak procedury ISO, QS.

Czy metoda obiektowo-zorientowana dostatecznie silnie akcentuje istotę akceptacji merytorycznej przez odbiorcę? Moim zdaniem – generalnie NIE!

Metodyka obiektowa funkcjonuje w świecie techniki i technologii jako dwie asocjacje: programowanie obiektowe i abstrakcja. Programowanie obiektowe, w świecie dużych systemów informatycznych zyskało już szeroką akceptację i dlatego coraz więcej projektów wykorzystuje języki programowania obiektowego. Jako najistotniejsze wady programowania obiektowego, dla zastosowania w systemach sterowania (wbudowanych), do niedawna, wymieniane była potrzeba dużych zasobów pamięciowych i mocy procesora. Przy aktualnym stanie techniki komputerowej - argumenty te upadają. Rozwiązania na poziomie programowania nie decydują jednak o akceptacji merytorycznej przez odbiorcę. Mogą one wprawdzie być brane pod uwagę, ale ich waga jest niewielka.

Czy abstrakcja związana z podejściem obiektowo-zorientowanym może być wykorzystana do poprawy akceptacji merytorycznej przez odbiorcę? Tak, pod warunkiem, że wykaże ona swoją praktyczną użyteczność. Dlatego zaproponowany proces treningowy metody „modelowania obiektowego dla automatyzacji wytwarzania” początkowo uczy przydatności abstrakcji, rozwijając jej zastosowanie, ale ciągle pozostając w relacji z rzeczywistością. W ten sposób zaspokajana jest w pierwszej kolejności potrzeba akceptacji merytorycznej, a dopiero później potrzeby osobiste.

Projekt prowadzony w pragmatycznym ujęciu sukcesu, tj. nienastawiony pierwszoplanowo na spełnienie obiektywnych miar, lecz skoncentrowany na ludzkich emocjach, „nie wytrzyma oceny” pryncypialnej – tj. formalnej. Oceniający z zewnątrz, który nie uczestniczył w projekcie, musiałby przecież zastosować miarę sukcesu uczestników, którą są przecież subiektywne emocje.

„F.C. Schiller, oceniając poglądy W. Jamesa (jak również dorobek filozofii pragmatycznej), uznał, że pragmatyzm jest:

1) humanizmem, bo indywidualnego człowieka czyni miarą rzeczy w odróżnieniu od scholastyizmu, który poddaje go „obiektywnej”, nadludzkiej mierze.

(...)

8) poglądem typowo brytyjskim, nie zaś kontynuacją filozoficznej myśli germańskiej.” [Wiem 01]

Czy przeszkodą w pojmowaniu sukcesu może być międzynarodowość zespołu jak mógłby na to wskazywać pkt. 8 powyższej definicji? Jeśli pojmowanie sukcesu nie jest spójne – to tak. Należy zatem postawić pytanie, czy przykładowo Europa, USA i Japonia tak samo zdefiniują sukces? W myśl dyskusji przeprowadzonej w rozdziale 3, na temat rozumienia jakości i doskonałości oraz przykładowo różnic kulturowych między Niemcami i Polakami wskazywanych przez Wojciechowskiego [Wojciechowski], sukces jest różnie definiowany.

Jak mierzyć sukces, jeśli jest nim akceptacja?

Cierpisz [Cierpisz 02] jako subiektywną miarę poziomu akceptacji proponuje „wspólnota celów”, tj. unikanie przez uczestników projektu działań, które zagrażają osiągnięciu celów przez innych uczestników.

Brak jednak miary obiektywnej dla akceptacji jakości, doskonałości, co rozpatrzono w rozdziale 3.

Czy projekt prowadzony w kryteriach sukcesu pragmatycznego przyniesie sukces biznesowy przedsiębiorstwa? Czy nie jest „drogą zabawą” dla zainteresowanych, która będzie przeciągać się, produkując subiektywnie „dobre produkty”? Czy prowadzenie takiego projektu jest etyczne?

Odpowiedź wymaga również spojrzenia z drugiej strony, na projekt prowadzony zgodnie z kryteriami pryncypialnymi. Projekt, którego wartość nowatorska jest bardzo duża, a zatem i ryzyko związane z opracowaniem produktu i jego wdrożeniem. Czy w przypadku gdy podczas realizacji projektu okazuje się, że początkowe założenia nie są prawdziwe, a więc wytworzony produkt nie spełni rzeczywistych oczekiwań odbiorcy, a więc i nie uzyska jego akceptacji, czy dążyć konsekwentnie do celu?

To poziom odpowiedzialności prowadzącego projekt i rzeczywista definicja sukcesu zdecydują o jego przebiegu. Rozpad wspólnoty celów, brak akceptacji, utrzymujące się przyzwolenie jednostronne prowadzą w efekcie do konfrontacji. Taka sytuacja powoduje, że strony starają się wykorzystać instytucje prawa, a więc umowę projektu, sprawozdania i formalne ustalenia.

5.4.2 Skalowanie systemu

Po zakończeniu treningu, wyborze i zainstalowaniu platformy wspomagającej modelowanie obiektowe i przeniesieniu przykładów diagramów opracowanych podczas treningu, można przystąpić do właściwego budowania Bazy Wiedzy. System informatyczny musi zostać napełniony aktualnymi danymi. Zgodnie z przyjętą metodyką wdrożenia MOAW proponuje się stopniowe „napełnianie” systemu informatycznego danymi. Konieczne jest wstępne opracowanie szkieletowego modelu organizacji danych oraz podstawowych widoków, zaś uzupełnianie danymi odbywać się będzie w sposób przyrostowy (inkrementacyjny). Podstawą budowania Bazy Wiedzy jest inżynieria odwrotna, (ang. *reengineering*), której celem jest wytworzenie spójnej wiedzy zespołowej oraz podniesienie poziomu zawartości dokumentacji.

5.4.3 Wymagania i zagrożenia dla wdrożenia Bazy Wiedzy

Sukces lub ryzyko wprowadzenia i użytkowania Bazy Wiedzy w firmie zależą będą w głównej mierze od poniższych czynników.

Akceptacja i zaangażowanie kierownictwa

Każda zmiana innowacyjna, jaką jest np.: wprowadzenie Bazy Wiedzy o systemie produkcyjnym, powoduje oddziaływanie na szereg elementów, takim jak np: przyzwyczajenia ludzi, sprawdzone metody pracy itp. Bez zrozumienia, chęci i determinacji przełożonych, zmiany systemowe będą się przeciągać. Gdy efekt/zakres/czas nie przekroczą „masy krytycznej”, to „zamrą”!

Inwestycje, ale i rozbudzone nadzieje zaangażowanych zostaną wówczas utracone. Zmianom takich nie mogą przewodzić podmioty zewnętrzne. Konsultanci zewnętrzni mogą natomiast, często dużo skuteczniej wspierać merytorycznie i mobilizować podczas budowania i wdrażania systemu.

Jednoznaczne stanowisko musi być najpierw zajęte na poziomie decyzyjnym i strategicznym firmy, a dopiero później wolno podejmować stosowne działania operacyjne.

Zaangażowanie przez użytkowników podczas opracowania i wdrożenia systemu

Jak wspomniano wcześniej, Baza Wiedzy o linii produkcyjnej nie jest produktem komercyjnym i dlatego też jej opracowanie, wdrożenie i rozwój muszą być prowadzone bezpośrednio przez zainteresowanych. Odpowiednie zaplanowanie przedsięwzięcia, specyfikacja wymagań i planu wdrażania oraz oszacowanie kosztów są niezbędnymi działaniami poprzedzającymi realizację wspieraną profesjonalnym zarządzaniem projektem. Z uwagi na pionierstwo tego rozwiązania i brak przykładów, poprzez projekty pilotażowe należy zdobywać doświadczenie i podejmować decyzje o dalszym rozwoju.

Akceptacja użytkowników systemu

Bez akceptacji przyszłych użytkowników, system nie spełni oczekiwanych założeń. Wymagania poszczególnych grup użytkowników są jednak bardzo różne. Realizacja systemu będzie zatem kompromisem zebranych wymagań, które będą się zmieniały w czasie. Dużą rolę dla zrozumienia systemu odgrywają Centra Oceny (ang. *assessments centre*), podczas których grupy uczestników poprzez dyskusję starają się lepiej zrozumieć problem.

Włączenie całej grupy specjalistów zajmujących się linią

Ponieważ za linię produkcyjną odpowiadają różni specjaliści, to nie jest możliwe użytkowanie i rozwijanie systemu bazy wiedzy wyłącznie przez wybrane osoby. Bez kompletnej grupy specjalistów, nie jest możliwe uzyskanie kompletności i spójności informacji o w systemie. Zakres zespołu aktywnych użytkowników Bazy Wiedzy będzie określany na podstawie specyfikacji funkcjonalnej.

Wypracowanie odpowiednich modeli dla informacji

Jak już wcześniej wspomniano przy opisie dokumentacji technicznej przechowywaną w Bazie Wiedzy, konieczne jest wypracowanie odpowiedniego sposobu przechowywania informacji o linii produkcyjnej w systemie informatycznym. Ponieważ informacja ta jest wielowymiarowa, a istotnym czynnikiem są relacje między elementami, konieczne jest wypracowanie odpowiednich widoków dla informacji. Widoki takie mogą przedstawiać struktury (modele)

lub tabele z danymi. Opracowanie odpowiednich filtrów pozwoli łatwo dotrzeć do informacji poszukiwanej, co będzie niezbędne wraz z rozwojem systemu i przyrostem danych.

Udostępnienie systemu komputerowego użytkownikom

Wszyscy użytkownicy systemu muszą mieć dostęp z własnych stanowisk pracy do bazy wiedzy. Oprogramowanie użytkownika, tzw. klienci, wymaga platformy Windows i usieciowania komputerów.

Systematyczność w użytkowaniu systemu

Użytkowanie Bazy Wiedzy wymaga systematyczności. Wszystkie dokonane zmiany dot. zautomatyzowanego systemu twórczego, ale również plany zmian powinny być wprowadzane Bazy Wiedzy. Dzięki codziennej pracy z systemem stanie się on rozszerzeniem takich narzędzi jak poczta e-mail oraz organizator (ang. *Personal Information Manager*). np.: Microsoft Outlook, Lotus Notes, GroupWise.

Utrzymanie i rozwój systemu Bazy Wiedzy

System Bazy Wiedzy będzie stanowił dodatkowy byt w organizacji. Będzie z tego zatem wynikała konieczność jego utrzymania i rozwijania. Odpowiednie przeszkolenie pracowników i nauczanie ich prawidłowych procedur użytkowania są niezbędne do utrzymania tego systemu. Typowy rozwój systemu będzie obejmował definicję nowych struktur danych i widoków. Rozwiązania te nie będą wymagały specjalistycznej wiedzy informatycznej. Natomiast poważne rozszerzenie funkcjonalności wiązać się będzie z zakupem *upgrade* oprogramowania i zasadniczo nie jest możliwe do wykonania samodzielnie.

5.5 Podsumowanie rozdziału piątego

W Rozdziale 5 zaproponowano model procesu wdrażania MOAW „Modelowania Obiektowego i UML dla Automatyzacji Wytwarzania”. W procesie tym wyróżniono trening, pilotaż oraz etap uruchamiania i użytkowania.

Przedstawiona propozycja wdrożenia nie stanowi kompletnej metodyki, a jest raczej „drogowskazem”. Szczególnie istotne są podrozdziały traktujące o budowaniu zespołu i motywacji, przebiegu treningu oraz software'owej platformie wspomagającej. Platforma ta, wraz z metodyką pracy MOAW, została nazwana Bazą Wiedzy o automatyzacji procesu produkcyjnego. Sformułowano podstawowe założenia dla tworzonego systemu oraz wymagania minimalizujące ryzyko nieudanego wdrożenia i użytkowania.

6 Weryfikacja modelowania obiektowego i ULM

Wymagania względem systemów produkcyjnych, scharakteryzowane w rozdziale 1.1, wymuszają konieczność modyfikacji wyposażenia technologicznego, dlatego modyfikacje na linii montażowej w zakładzie produkcyjnym należą do codzienności. Jak wskazano w niniejszej pracy doktorskiej, modyfikacje takie są zwykle procesem automatyzacji, którego efektywność i bezpieczeństwo zależą od wiedzy o stanie aktualnym systemu.

Podczas uruchamiania nowego systemu montażowego, wiedza o rozwiązaniach wewnętrznych automatyzacji została przetransferowana do przyszłych użytkowników w ograniczonym zakresie. Uzasadnione to było ograniczoną potrzebą i możliwościami jej przejęcia, dlatego dostawca pozostał koordynatorem wiedzy o rozwiązaniu, tzw. „outsourcing”.

W rozdziale 3.1.4 wykazano jednak, że wzrost stopnia automatyzacji przesuwają kompetencje utrzymania ruchu w kierunku rozwijania systemu, zatem *aktywnej automatyzacji*. Klasyczna organizacja zadaniowa nie sprzyja wymianie wiedzy. Konieczność szybkich, często doraźnych modernizacji oraz rosnąca rotacja personelu spowodowały niespójności między trzema głównymi produktami automatyzacji. Potrzeba oraz zalety z wprowadzenia „rozwiązań trzecich” do procesu montażowego, jak również konieczność tzw. dużych zmian (dostosowanie do Ramowej Specyfikacji Wymagań wiąże się z koniecznością przeprogramowania wszystkich stacji automatycznych) spowodowały potrzebę budowania wiedzy o automatyzacji posiadanej linii poprzez inżynierię odwrotną.

Kompleksowość linii montażowej, częste zmiany technologii oraz konieczności produkcji były motywacją do wprowadzenia narzędzia MOAW.

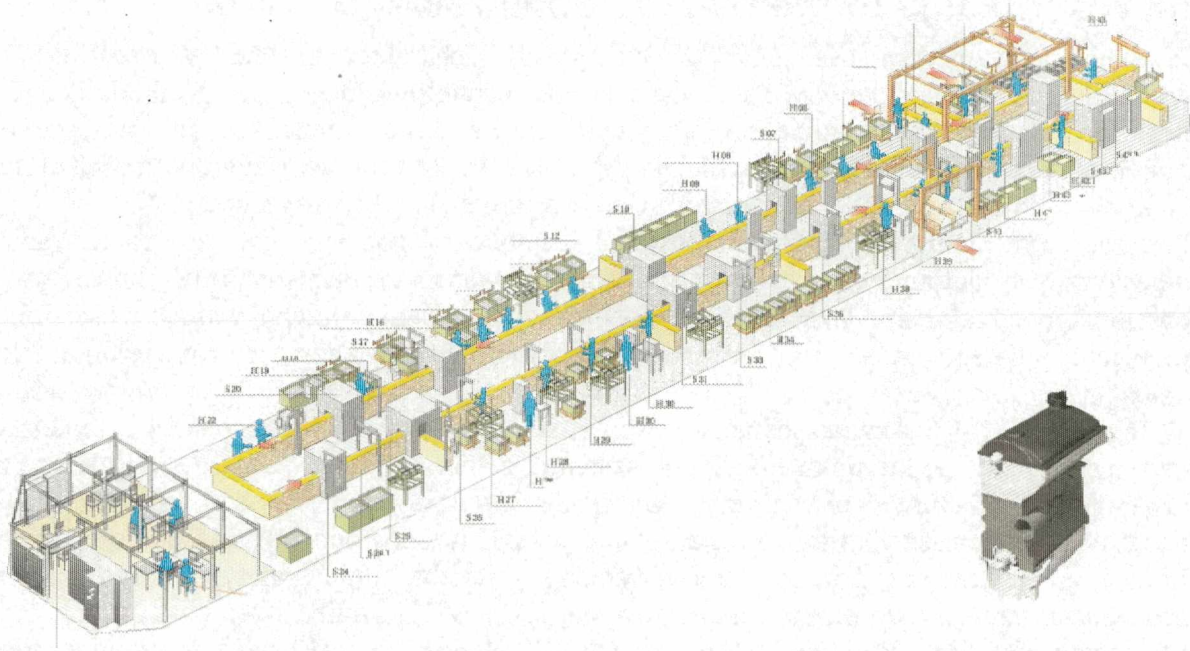
Weryfikację Modelowania Obiektowego dla Automatyzacji systemu produkcyjnego przeprowadzono na linii montażu silników, w okresie od maja do grudnia 2002. Uczestniczył w niej pięcioosobowy zespół. Weryfikacja ta była prowadzona w ramach projektu „RMdata+”.

6.1 Opis procesu produkcyjnego

Linia montażowa RM przedstawiona poglądowo na rys. 6.1, jest jedną z trzech linii montażu silników spalinowych o zapłonie samoczynnym. Elastyczność linii umożliwia montaż kilku różnych typów silnika. Na linii pierwszej montowana jest głowica silnika, na linii drugiej montowany jest kadłub silnika, na linii trzeciej wyposaża się silnik we wszystkie podzespoły uzupełniające. W ten sposób silnik zostaje przygotowany do uruchomienia w celu przetestowania podczas pracy, na stanowiskach kontrolnych. Na liniach montażowych dominuje montaż ręczny, a jedynie dokręcanie śrub i pomiary realizowane są automatycznie. Silnik montowany jest na palecie montażowej, która jest automatycznie transportowana między poszczególnymi stanowiskami. Stanowiska ręczne i automatyczne tworzą pojedynczy łańcuch, z czterema odgałęzieniami dla zadań poprawkowych i naprawczych. Strategia jakości montażu wymaga od monterów bezbłędnego wykonania czynności na stanowisku w czasie krótszym od czasu taktu linii, który wynosi ok. 45s.

Montaż silnika jest procesem częściowo zautomatyzowanym. Automatyzacja linii obejmuje:

- sterowanie przepływem palet,
- sterowanie pracą stanowisk automatycznych,
- akwizycję statusów i błędów operacji technologicznych,
- nadzorowanie statusów operacji technologicznych i w razie konieczności przekierowywanie palet na stanowiska naprawcze.



Rys. 6.1 Widok linii montażowej silników [RMdata+]

Podstawą powyższej automatyzacji są nośniki pamięciowe MOBY, dołączone do każdej palety transportowej. Nośnik MOBY zastosowany na paletach linii RM ma pojemność 48kBajtów pamięci, która jest zorganizowana w postaci 256 stron, każda po 256 Bajtów. Wszystkie stacje automatyczne i większość stacji ręcznych wyposażone są w stacje odczytująco-zapisujące SLS-MOBY. Przed wprowadzeniem palety do systemu, w pamięci MOBY jest zapisywany program marszruty technologicznej. Program ten składa się z maksymalnie 128 operacji technologicznych (OT). Każdej z operacji technologicznych przyporządkowany jest numer programu technologicznego z przedziału 0-7. Po zakończonych operacjach technologicznych na stacjach wyposażonych w SLS-MOBY, zapisywane są statusy ich wykonania. W razie błędu dokręcenia lub nieprawidłowego wyniku pomiarowego, transport palety nie jest wstrzymywany, lecz zapisywany jest status błędu. Dzięki temu nie dochodzi do blokowania linii, lecz paleta poddawana jest kolejnym czynnościom technologicznym, aż do możliwości przekierowania na najbliższą stację napraw. Po dokonaniu naprawy paleta wraca do głównego obiegu procesu technologicznego. Jeśli zostanie zauważony błąd krytyczny, którego usunięcie wykracza poza możliwości stanowiska montażowego, a nawet naprawczego, to silnik jest „neutralizowany”. Oznacza to, że do pamięci zostaje wpisany znacznik powodujący przejazd silnika do końca linii na stanowisko wyjściowe, bez dokonywania kolejnych operacji technologicznych.

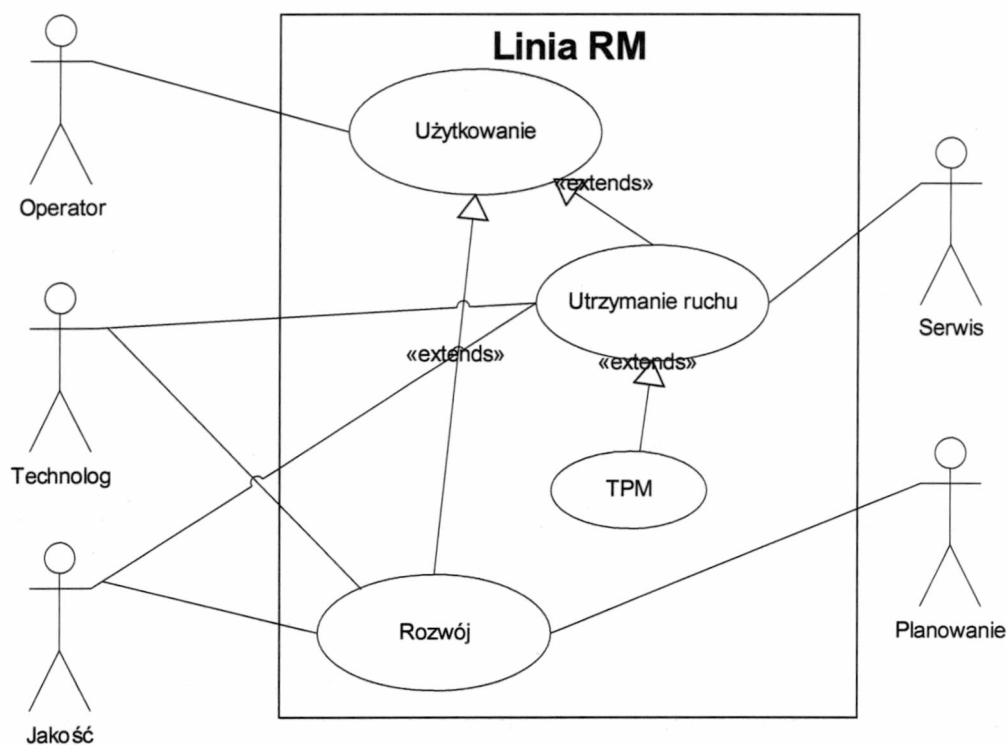
Za utrzymanie linii montażowych odpowiada Dział Serwisu, zaś za ich dalszy rozwój Dział Planowania.

Nowe typy silników oraz doskonalenie procesu produkcyjnego powodują, że zmiany na linii są niemal ciągłe. Tzw. „duże zmiany”, obejmujące modernizacje lub rozszerzenie stanowisk montażowych czy kontrolnych, dokonywane są przez dostawców zewnętrznych, natomiast drobne usprawnienia takie jak np. Poka-Yoka, wprowadzane są przez pracowników firmy.

6.2 Użytkownicy linii RM i ich wiedza

Za utrzymanie i rozwój linii RM odpowiada wielu użytkowników. Przez pojęcie użytkownika rozumie się rolę, jaką odgrywa osoba lub inny system zewnętrzny względem omawianego systemu. W przypadku linii RM użytkownikami systemu są operatorzy, technolodzy, pracownicy nadzorowania jakości, pracownicy utrzymania ruchu, pracownicy działu rozwoju, co przedstawiono na diagramie przypadków użycia – patrz rys. 6.2. Również paleta z silnikiem, dla której linia RM spełnia określone usługi, jest użytkownikiem systemu. Każdy z użytkowników inaczej rozkłada swoje priorytety, zgodnie z zakresem obowiązków i odpowiedzialnością. Nadrzędnym celem dla wszystkich użytkowników jest system produkcyjny obejmujący: proces technologiczny, dane produkcyjne i wiedzę o linii reprezentowaną przez ludzi i dokumentację. Dlatego też wzajemne rozumienie i praca w grupie interdyscyplinarnej oraz doskonałe rozumienie procesów są podstawą efektywnego utrzymania i rozwoju tego systemu. Wiedza o linii RM i umiejętność jej wykorzystania stanowią zatem szczególnie cenny kapitał firmy. Przy kompleksowości problemu na poziomie linii RM, konieczny jest jednak określony formalizm w postaci dokumentacji o stanie poprzednim, aktualnym i przyszłym. Dokumentacja to sformalizowana wiedza dająca się wymieniać, archiwizować i przetwarzać. W celu jej przetwarzania mogą być użyte odpowiednie narzędzia technologii informatycznej. Zle przygotowana i przetwarzana dokumentacja może stać się jednak źródłem problemów, tj. błędnych decyzji, przeciążenia informacyjnego, itp.

Pracownicy, którzy wykonują zadania utrzymania i rozwoju linii, posiadając należytą wiedzę, informacje o procesie, metody i narzędzia wspomagające oraz umiejętności pracy grupowej, będą panowali nad procesem, dzięki czemu proces ten będzie stabilny i bezpieczny.



Rys. 6.2. Użytkownicy linii RM [jr]

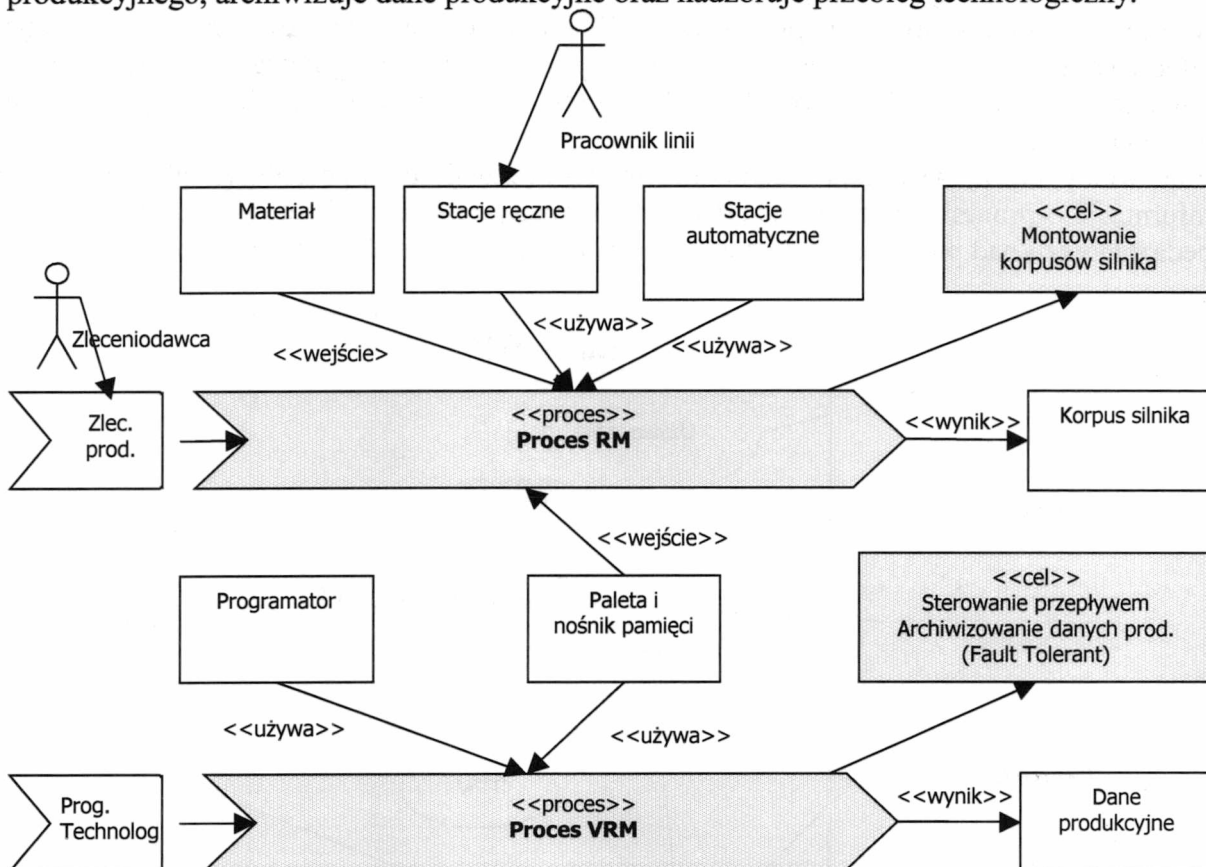
Celem strategicznym powinno być zatem zwiększanie wiedzy organizacji poprzez wspieranie pracy grupowej (motywacja i zapewnienie komunikacji), wspieranie kreatywności i innowacyjności oraz dokumentowanie informacji. Wraz ze zmieniającą się charakterystyką

socjo-technicznych systemów produkcyjnych, tj. wzrostem udziału oprogramowania, wzrostem elastyczności i częstymi zmianami, konieczne staje się dostosowywanie organizacji wspierającej te systemy [Reiner et al 2003]. Problematyką taką zajmuje się obszernie dyscyplina „Zarządzanie Wiedzą”.

6.3 Biznesowa perspektywa procesu RM

Dla lepszego zrozumienia problematyki linii RM na poziomie strategicznym, opracowano model biznesowy linii RM (rys. 6.3), którego interpretację przedstawiono poniżej.

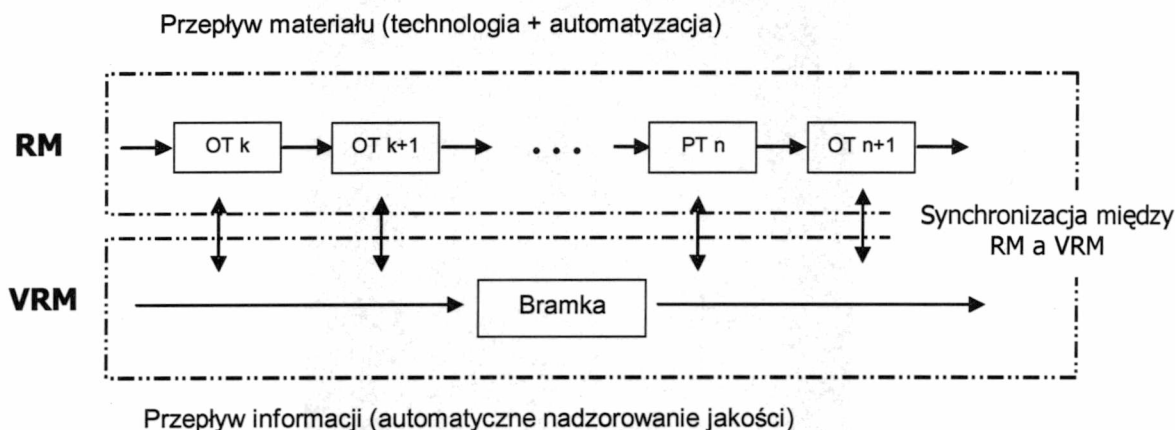
Proces montażowy obejmuje dwa współbieżne procesy: RM i VRM. Procesy te są wzajemnie synchronizowane poprzez paletę i nośnik pamięci. Celem procesu RM jest montaż korpusu silnika, zaś procesu VRM sterowanie przepływem palet, archiwizowanie danych produkcyjnych oraz zapewnienie odporności na błędy (ang. *Fault Tolerant System*). Proces VRM jest zatem procesem automatyzacji, wspierającym proces technologiczny. Proces RM używa stacji automatycznych i ręcznych do montażu części zgodnie ze zleceniem produkcyjnym montażu korpusu silnika. Proces VRM, po wprowadzeniu programu produkcyjnego, archiwizuje dane produkcyjne oraz nadzoruje przebieg technologiczny.



Rys. 6.3 Proces biznesowy linii montażu silników [jr]

Z zaprezentowanego na rys. 6.3 przykładu wynika, że nadrzędnym celem firmy jest uzyskanie wysokiej jakości montażu silnika. Osiągnięcie tego celu zależy jednak bezpośrednio od jakości „montażu silnika wirtualnego” VRM, tj. jego odwzorowania informacyjnego. System tolerancji błędów montażowych i nadzorowania jakości montażu funkcjonuje na bazie danych produkcyjnych - produktu wirtualnego. Wszelkie modyfikacje logistyczne procesu technologicznego muszą zatem być spójne z jego odpowiednikiem wirtualnym, jak to przedstawiono na rysunku 6.4.

Pojęcie silnika wirtualnego (VRM) zostało wprowadzone dla zobrazowania celu i znaczenia wymiany danych z nośnikami MOBY. Model danych produkcyjnych zawiera dane procesowe i parametryczne opisujące wymagania montażowe (programy dla operacji technologicznych, parametry operacji) oraz przebieg procesu montażowego, tj. statusy operacji, informacje o błędach i statusy lub wyniki pomiarów. Jakość produktów zależy od procesu RM, a ten zależy bezpośrednio od VRM, gdyż system automatycznego nadzorowania jakości VRM, podejmuje decyzje wyłącznie na podstawie danych zapisanych w nośniku MOBY. Dlatego też uniezależnienie systemu nadzorowania jakości od systemu montażowego jest warunkiem koniecznym dla zapewnienia niezawodności całego systemu.



Rys. 6.4 Istota synchronizacji RM i VRM [jr]

Wnikając szczegółowiej w koncepcję automatyzacji linii RM i nadzorowania jakości montażu, zauważamy, że jest ona realizacją systemu odpornego na błędy (ang. *Fault Tolerant System*). Oznacza to, że po wystąpieniu błędu (np. automatycznego dokręcania) przepływ palet nie jest wstrzymywany, a do nośnika pamięciowego MOBY wpisywany jest jedynie status niepoprawności operacji „NO”. Mimo statusu „NO” silnik poddawany jest kolejnym operacjom montażowym i kontrolnym, aż do osiągnięcia bramki kontroli strefy. W razie rozpoznania statusu „NO” przez bramkę strefy, paleta jest przekierowywana na stanowisko napraw. Na stanowisku tym dokonuje się naprawy i wyzerowuje flagę statusu błędu, poczym paleta ponownie jest wprowadzona do głównego obiegu transportowego systemu montażu RM i poddawana kontroli.

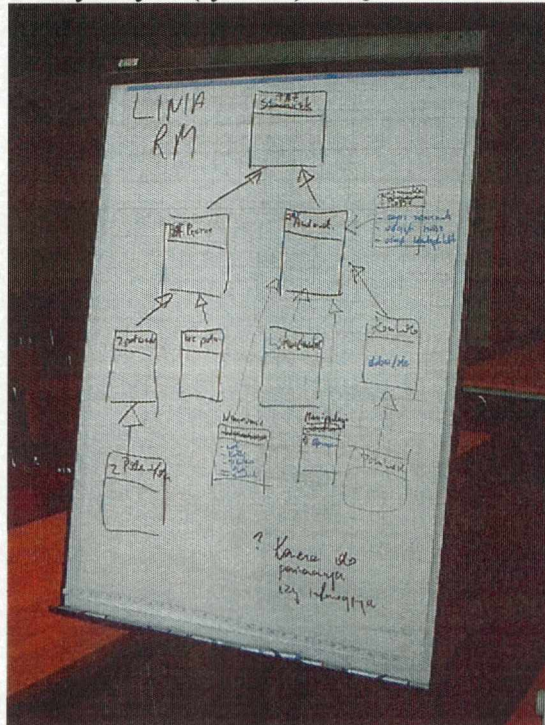
Ponieważ pojęcia „błąd” i stan silnika/palety „NO” są mylące, to konieczne jest wprowadzenie pojęcia „tryb błędu”. Tryb błędu wskazuje na stan wewnętrznej obsługi sytuacji nietypowej. Jeśli „tryb błędu” nie zostanie prawidłowo obsłużony przez system (np. przepuszczony przez bramkę strefy), to dopiero wówczas staje się on błędem. Dwupoziomową analizę błędów i trybów błędów powinien zapewnić program analizy danych produkcyjnych. Analiza częstości występowania „trybów błędów” podpowiada, że wydajność systemu jest zagrożona, a jakość niektórych połączeń technologicznych może być gorsza, etc. Otrzymujemy zatem ostrzeżenia, których przyczyny wymagają szczegółowego analizowania. Doraźne „łatanie” lub „majsterkowanie” w systemie „*Fault tolerant*” jest bardzo niebezpieczne. Zwykle system taki stanowi sieć wielopoziomowych powiązań. Analiza takiego systemu za pomocą metody FMEA może być niewystarczająca i dlatego też powinny być wykorzystywane znacznie bardziej zaawansowane metody analizy, takie jak:

- analiza Drzewa Błędów (ang. *Fault Tree*) oraz
- analiza ORM (ang. *Operational Readiness Management*).

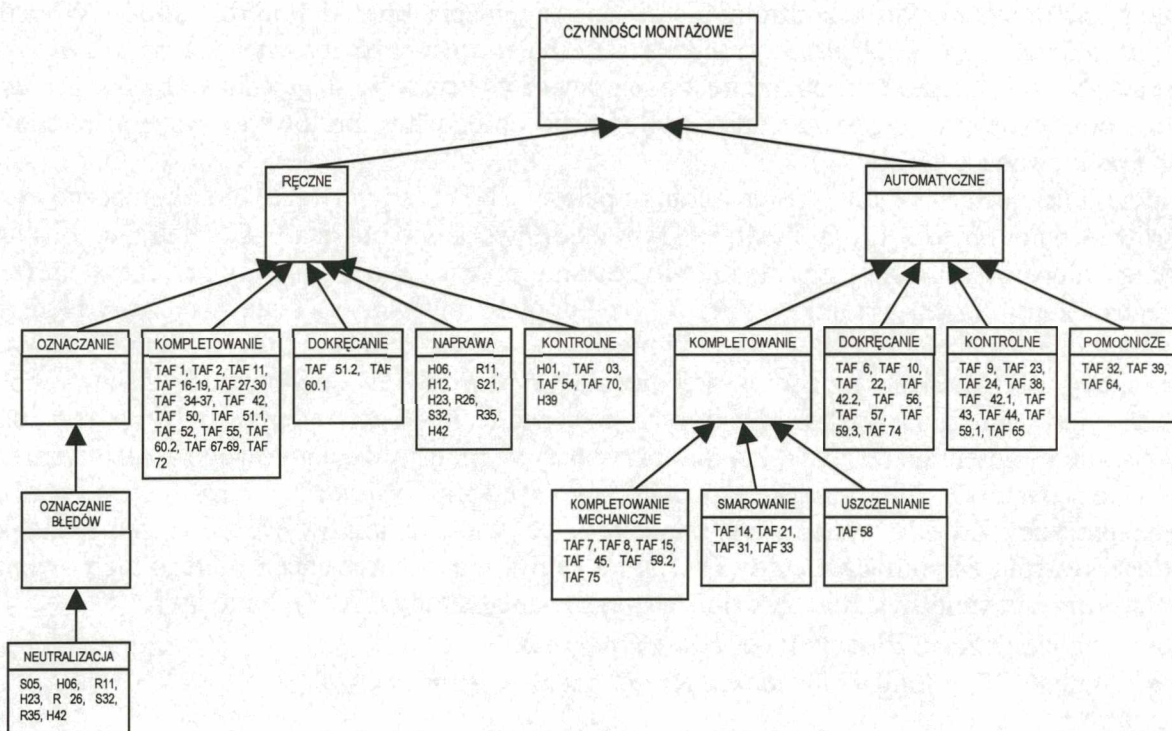
6.4 Zasoby i procesy RM

W pierwszym kroku analizy zasobów i procesów linii RM zespół uczący się MOAW utworzył diagram rozmieszczenia, identyfikujący stacje automatyczne, ręczne oraz pracowników je obsługujących. Diagramowanie prowadzone było na tablicy (patrz fotografia 6.5).

W kolejnym kroku dokonano klasyfikacji obiektowej czynności montażowych (rys. 6.6), czynności kontrolnych automatycznych (rys. 6.7) i błędów montażowych (rys. 6.8).



Fot. 6.5 Fot. jednego z arkuszy roboczych – klasyfikacja czynności montażowych RM [jr]

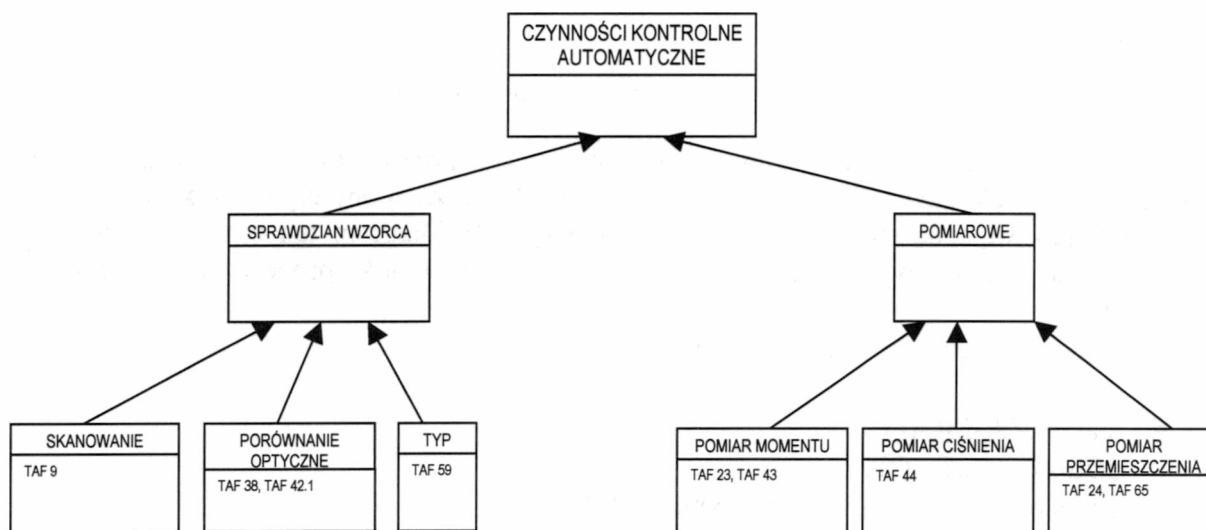


Rys. 6.6 Obiektowy model klas czynności montażowych [jr]

Dzięki przeprowadzonej klasyfikacji zredukowano liczbę czynności montażowych z 92 do 11 klas, co daje redukcję na poziomie 88%. Praktycznie oznacza to, że należy utworzyć po jednym szablonie (wzorcu) dla:

- 8 czynności neutralizacji,
- 25 czynności kompletowania ręcznego,
- 2 czynności dokręcania (ręcznego),
- 9 czynności naprawczych,
- 5 czynności kontrolnych (manualnych),
- 6 czynności kompletowania mechanicznego,
- 4 czynności oliwienia,
- 1 czynności uszczelniania,
- 8 czynności dokręcania,
- 9 czynności kontrolnych,
- 3 czynności pomocniczych.

Jak wynika z diagramu (rys. 6.6) powyższe klasy są specjalizacjami montażu ręcznego albo automatycznego, a te, są specjalizacjami czynności montażowych. Tak więc, cechy zdefiniowane w klasie nadrzędnej, dzięki mechanizmowi dziedziczenia, nie wymagają już powtórnego rozważania. Ważne jest, aby cechy klasy rozumieć abstrakcyjnie, tzn., że mogą one specyfikować dokumentację, przebieg pracy, przebieg konserwacji lub wymagania jakościowe, a nawet kolor czy styl urządzeń, etc.



Rys. 6.7 Obiektowy diagram klas czynności kontrolnych [jr]

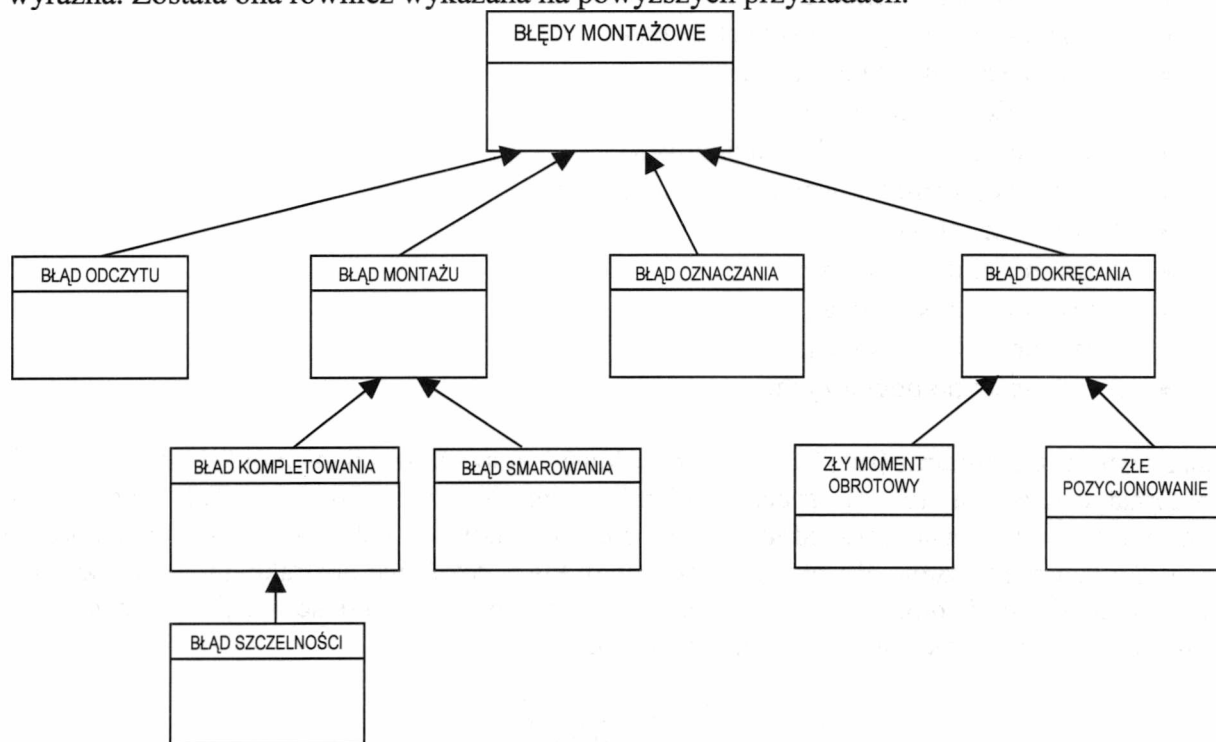
Klasyfikacja czynności kontrolnych nie przynosi tak spektakularnych wyników redukcji problemu jak powyższy przykład czynności montażowych.

Na podstawie dokumentacji błędów linii RM dokonano klasyfikacji obiektowej błędów montażowych (rys. 6.8) Problem został zredukowany z 366 błędów do 7 ich klas.

Zaletą klasyfikacji obiektowej, dzięki zasadzie propagacji, jest również możliwość szybszego usunięcia błędów zauważonych podczas użytkowania systemu.. Oznacza to, że wystąpienie błędu w jednej realizacji klasy (obiektie), wymaga podjęcia działań korygujących we wszystkich jej instancjach. Jeśli potrzeba taka nie występuje, to opracowana klasa nie jest reprezentatywna dla wszystkich problemów i wymaga wydzielenia.

Tworzenie powyższej klasyfikacji wymaga podejścia metodycznego, które nie jest nastawione na szybkie i tanie rozwiązanie problemu, lecz opracowanie sposobu jego

rozwiązania. W perspektywie lokalnej może okazać się, że nakład poniesiony na klasyfikację są większe niż w przypadku podejścia klasycznego. W perspektywie globalnej, szczególnie w przypadku problemów kompleksowych, przewaga przedstawionej metodyki jest intuicyjnie wyraźna. Została ona również wykazana na powyższych przykładach.



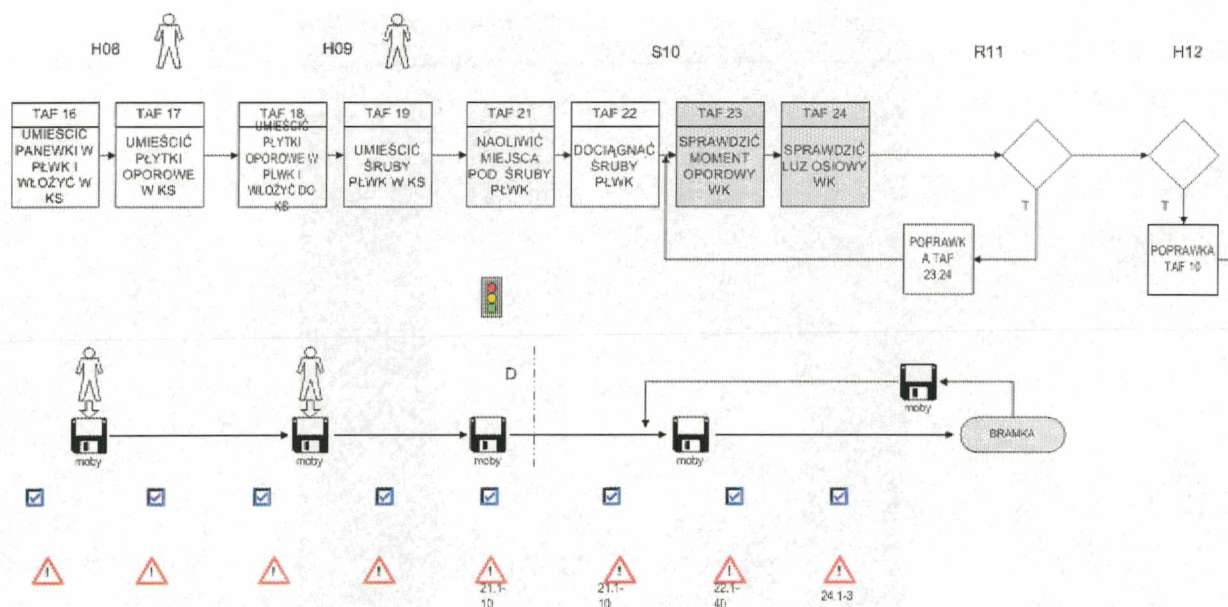
Rys. 6.8 Obiektowy model klas błędów montażowych [jr]

Na bazie diagramu rozmieszczenia przystąpiono do „rozwijania” Diagramu Przepływu Procesu (ang. *Process Flow Diagram*) dla linii RM. Za pomocą techniki diagramowej zobrazowano logiczną strukturę procesu – patrz rys.6.9. Diagram ten został skonstruowany jako widok warstwowy. Każda z warstw rozszerza widok procesu w następujących kategoriach:

- stanowiska ręczne i automatyczne,
- pracownicy,
- stacje SLS-MOBY,
- status operacji „OK”,
- zapis błędów montażowych (trybu błędu),
- Poka-Yoka,
- „neutralizacja”,
- inne uwagi.

Dzięki zebraniu wielu kategorii informacji lepiej zobrazowano wzajemne współzależności między poszczególnymi stacjami, jak również między zastosowanymi rozwiązaniami technicznymi..

Diagram ten pozwolił zidentyfikować i przeanalizować synchronizację między procesem RM a VRM. Okazało się, że w systemie tym występują obszary, których działanie nie jest typowe. Wynikają one ze specyficznych wymagań procesu albo uproszczeń wprowadzonych z uwagi na redukcję kosztów. Obszary specyficzne wymagają starannego udokumentowania, gdyż podobieństwo zewnętrzne może być mylące.



Rys 6.9 Fragment rozszerzonego diagramu przepływu procesu (stan V.2002) [jr]

Diagram na rys. 6.9 został opracowany za pomocą programu VISIO 2002 (Microsoft). Ponieważ diagram procesu RM powinien być wykorzystywany przez różnych specjalistów, tj. planowanie, wytwarzanie, jakość, automatyzacja i serwis, to jego współdzielenie i konieczność zapewnienia konsystencji odgrywają kluczową rolę. Powyższe narzędzie (VISIO) nie wspiera jednak pracy grupowej.

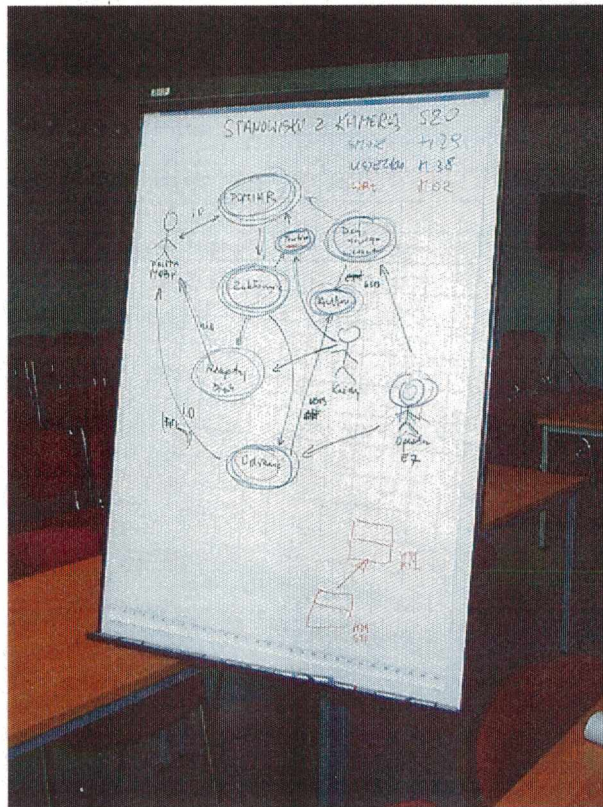
6.5 Przypadki użycia dla stanowisk na linii RM

Funkcjonalność urządzeń na linii oraz adresaci tej funkcjonalności zostali zamodelowani za pomocą diagramów przypadków użycia.

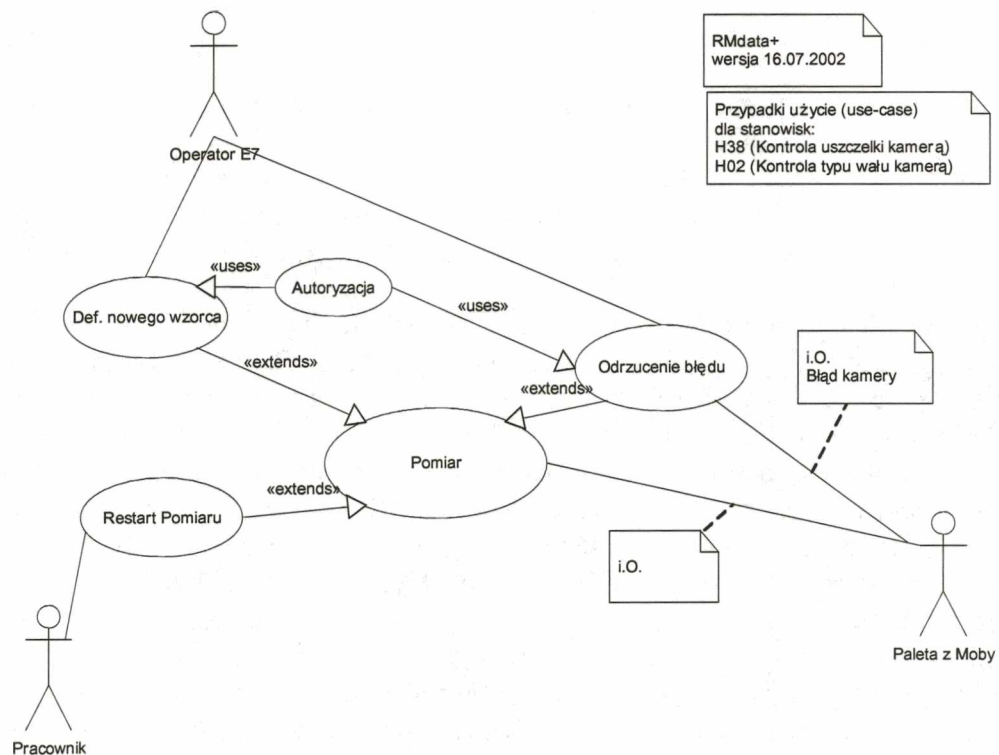
Na Rozszerzonym Diagramie Przepływu Procesu zidentyfikowano stanowiska z systemami wizyjnymi jako krytyczne, gdyż nie cechują się one 100% niezawodnością działania, ponieważ obraz z kamery podlega zakłóceniom np. od refleksów świetlnych, wpływu temperatury etc.

Aktualnie, po błędzie rozpoznawania przez system wizyjny, wstrzymywane jest działanie automatu z jednoczesną sygnalizacją błędu dla operatora. Do obowiązków obsługi należy ocena błędu przez jego wzrokowe sprawdzenie. Na tej podstawie operator podejmuje decyzję o wznowieniu pomiaru przez kamerę albo odrzuceniu sygnalizowanego przez system błędu. Ze względu na duże podobieństwo porównywanych elementów, operator może pomylić się przy ocenie wzrokowej części (np. kolejność znaczników na tłokach). Algorytm taki, dla obsługi stacji kontrolnej z kamerą jest źródłem potencjalnych błędów. Pozwala on bowiem operatorowi na skwitowanie błędu, tj. jego odrzucenie, mimo faktycznego jego wystąpienia.

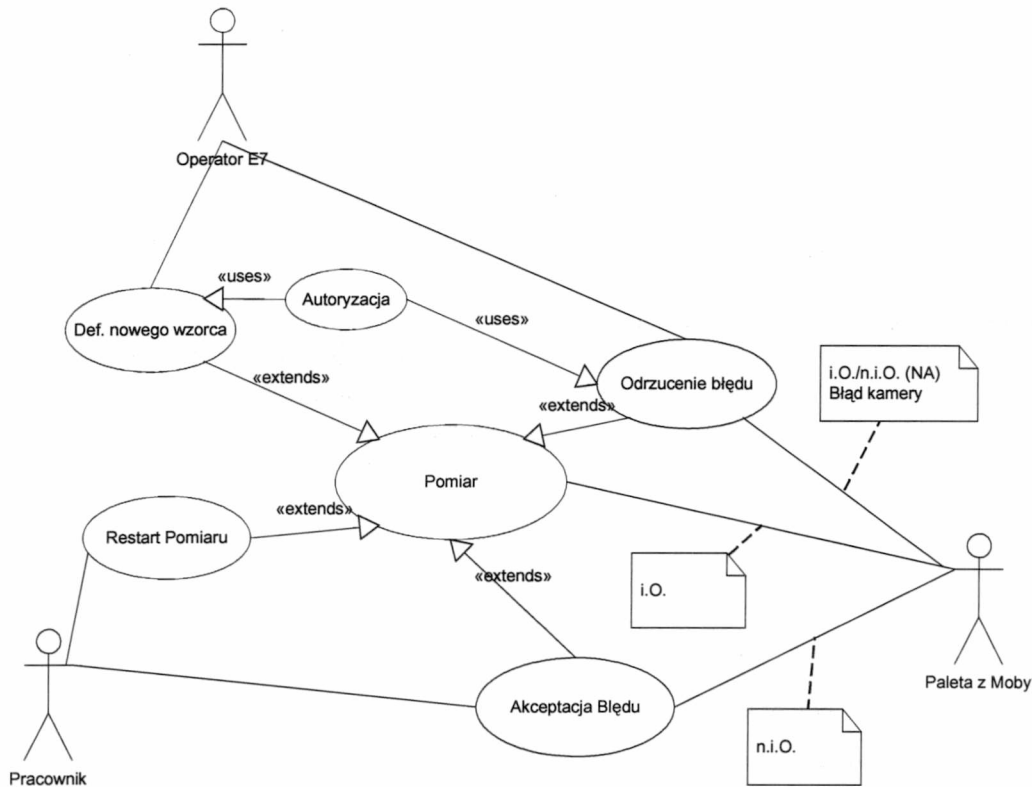
Przy rozwiązywaniu tego problemu grupa projektowa sporządziła diagram przypadków użycia w języku UML specyfikujący propozycję funkcjonowania stacji z kamerą (patrz rys. 6.10 i 6.11). Według nowej propozycji zdefiniowano trzech aktorów w systemie. Aktorami są badany silnik z nośnikiem MOBY oraz dwóch operatorów o różnych uprawnieniach. Po wystąpieniu błędu zgłoszonego przez system wizyjny, operator o standardowych uprawnieniach ma prawo do zainicjowania ponownego pomiaru, gdy uważa, że wystąpił błąd pomiarowy, a nie błąd montażu, albo też zaakceptowania błędu, co spowoduje, że silnik zostanie skierowany na najbliższą poprawkę. W ten sposób status „OK” może być wystawiony wyłącznie przez automat.



Rys. 6.10 Fot. jednego z arkuszy roboczych – przypadki użycia dla stacji z kamerami [jr]



Rys. 6.11 Przypadki użycia stacji kontroli H38, H02 [jr]



Rys. 6.12 Diagram przypadków użycia dla stacji z kamerami MA-BYC [jr]

W przypadku, gdy błąd jest spowodowany błędem wzorca (stacja S20 – sprawdzenie typu tłoka, gdy przesunięte są znaczniki na tłokach), to operator o uprawnieniach serwisowych (klucz „E7”) może odrzucić błąd pomiarowy, ale również montażowy, tj. wpisać status operacji ‘NO’, co będzie jednak w konsekwencji archiwizowane.

Takie zmiany wymagają ułatwień w sposobie definiowania nowego wzorca, co jest główną przyczyną problemów. W przeciwnym przypadku wydajność systemu zostanie ograniczona przez restrykcyjne wymagania bez dostosowania odpowiednich rozwiązań umożliwiających ich spełnienie. Uprawniony operator „E7” musi mieć możliwość natychmiastowego rozszerzenia bazy wzorców testowych.

Ze względu na podobieństwa funkcjonalne stacji kontrolnych z kamerami, powyższa specyfikacja działania może być zaimplementowana w każdym tego typu stanowisku (S22 – kontrola tłoków, H29 – kontrola smoka, H38 – sprawdzanie uszczelki, H02 – kontrola wału). Wdrożenie tej propozycji pociąga za sobą modyfikację oprogramowania sterowników PLC nadzorujących pracę stanowisk kontrolnych z kamerą. Powyższe diagramy stanowią specyfikację wymagań dla zleceniobiorcy oraz są podstawą przygotowania odbioru.

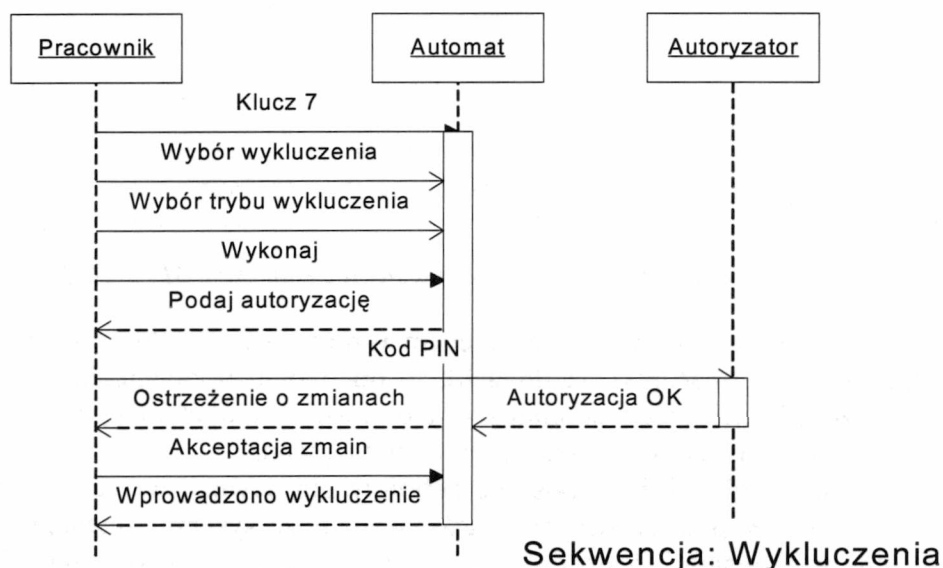
6.6 Zachowanie i działanie urządzeń linii RM

Zachowanie urządzeń i interakcja z operatorem jest wstępnie określana już podczas specyfikacji wymagań. Ponieważ jednak modelowanie przypadków użycia nie wspiera takiego widoku, dlatego też posłużono się Diagramem Interakcji i Diagramem Stanów.

Diagram Interakcji (rys.6.13), specyfikuje scenariusz interakcji operatora z systemem podczas „wykluczania operacji”. Wykluczanie operacji to funkcja konfiguracyjna stanowiska automatycznego, dzięki której możliwe jest przededefiniowanie jego działania. Przykładowo, gdy uszkodzeniu uległo jedno z wrzecion wkrętarki, to status operacji będzie wpisywany jako „NO.”. Skutkiem tego będzie automatyczne przekierowywanie wszystkich palet na stanowisko naprawcze w celu usunięcia błędu montażowego. Stanowisko napraw nie jest

jednak zaprojektowane do obsługi tak dużego ruchu, w wyniku czego będzie dochodziło do „zatykania systemu”. Wprowadzenie strategii awaryjnej, tj. dostawienie pracownika z kluczem dynamometrycznym rozwiązuje problem tylko po stronie RM. Rozwiązanie problemu po stronie VRM wymaga natomiast wprowadzenia możliwości wpisywania statusu „OK”, a więc stanowiska SLS-MOBY. W tym celu można wykorzystać konfigurowalność maszyny, tj. „wykluczenia”, za pomocą których wyłączane jest wpisywanie statusu operacji. Ponieważ operacja taka jest ingerencją w system nadzorowania jakości, to jej użycie musi być chronione.

Propozycja rozszerzenia mechanizmu ochrony dla wykluczeń została zamodelowana na rys. 6.13. Na diagramie tym wyróżniono trzech partnerów interakcji: pracownika, automat i „autoryzator”. Użycie klucza serwisowego dopiero w koniunkcji z kodem PIN umożliwia dokonanie zmiany. Na diagramie przedstawione zostały również klasy komunikatów otrzymywane przez operatora od systemu, np.: pytanie o PIN, ostrzeżenie o zmianach, potwierdzenie zmian.

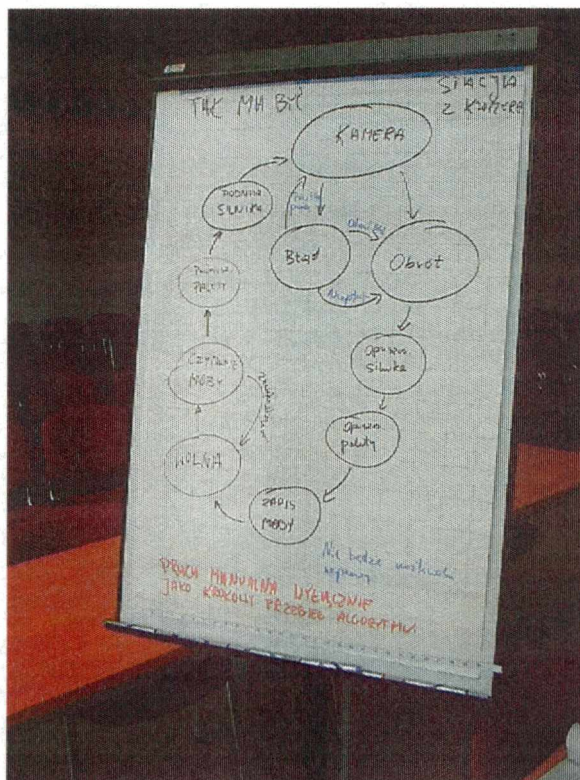


Rys. 6.13 Scenariusz wykluczenia [jr]

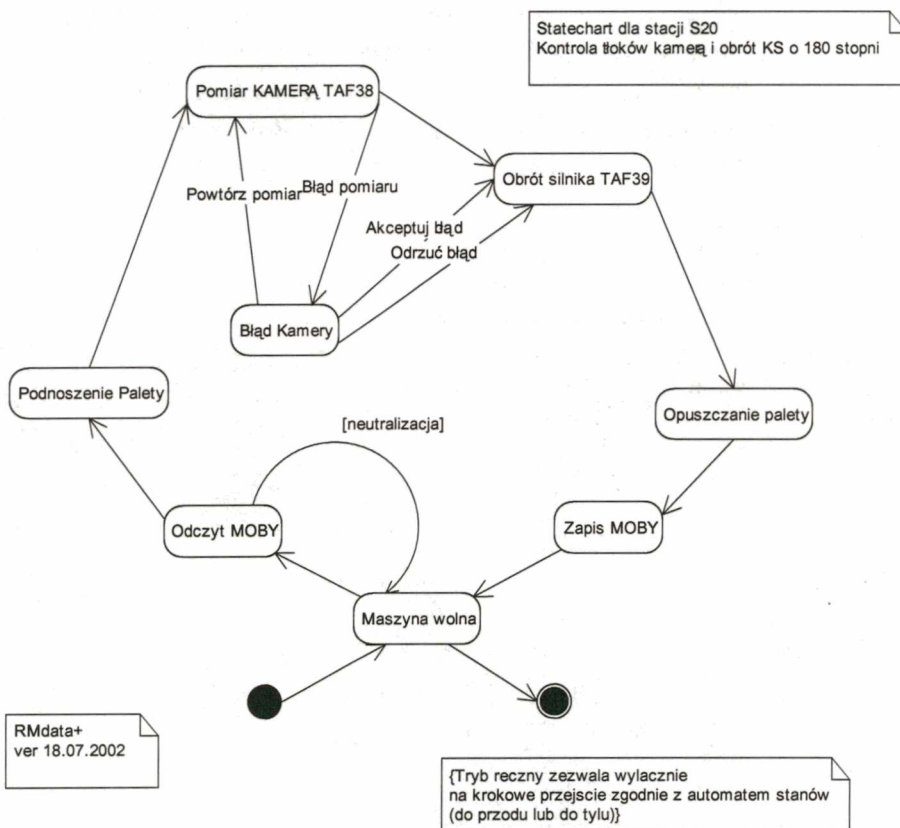
Do zamodelowania zachowania stacji skorzystano z diagramu stanów – patrz rys. 6.14, 6.15. Weryfikacja pierwszego z opracowanych modeli zachowania dla stacji S20 z systemem wizyjnym wykazała, że nie jest on spójny, gdyż przedstawia zachowanie rzeczywiste i idealne. W celu wyspecyfikowania „rzeczywistego działania” stacji, konieczne okazały się obserwacje przy stanowisku, podczas których uczestniczył jeden z pracowników obsługujących stanowisko. Dzięki prostocie i intuicyjności składni i semantyki diagramu zachowania, operator sam przedstawił zweryfikowane działanie stanowiska w przyjętej konwencji.

Podczas konstruowania wymaganego schematu działania stacji zwracano uwagę na uogólnienie (generalizację), dzięki której dążono do zastosowania opracowanych rozwiązań dla stacji tej samej klasy.

Na diagramie (rys. 6.15), przedstawiono specyfikację sekwencji działania stacji. Z modelu tego wynika, że ograniczono w nim działanie trybu ręcznego. Ponieważ tryb ręczny narusza bezpieczeństwo systemu i nie jest chroniony kluczem E7, to konieczne jest ograniczenie jego funkcjonalności jedynie do krokowego przebiegu algorytmu.



Rys. 6.14 Fot. jednego z arkuszy roboczych – działanie stacji z kamerą [jr]

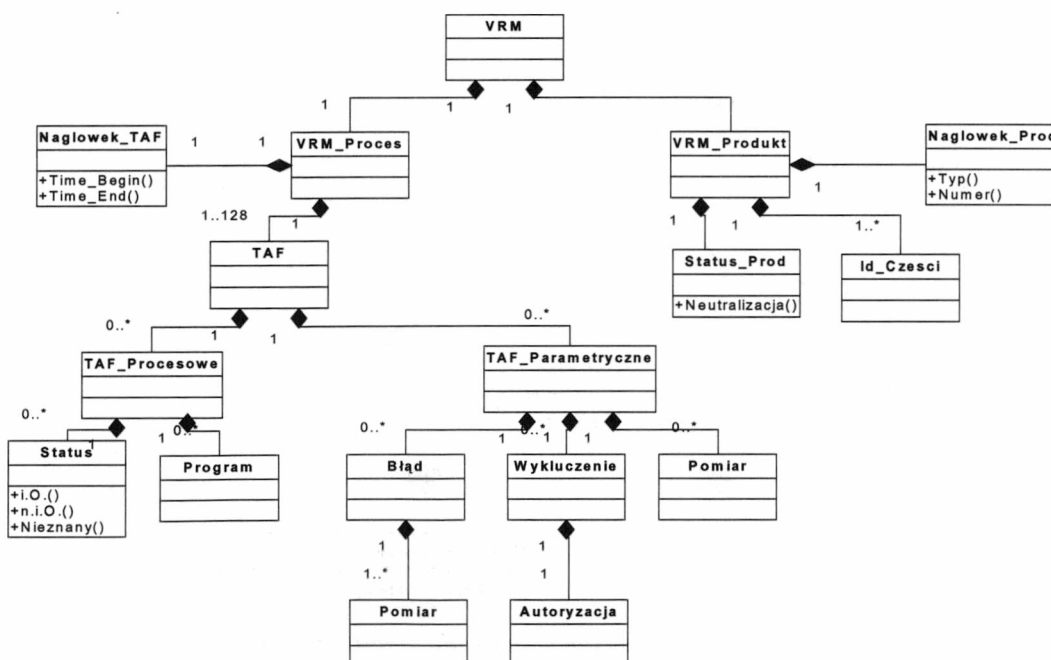


Rys. 6.15 Diagram stanów dla stacji S20 [jr]

6.7 Model danych procesowych

Analiza ramowej specyfikacji wymagań dla montażu silników przeprowadzona w ramach projektu „RMdata+” wykazała, że konieczne jest wprowadzenie dodatkowego statusu stanu silnika „NIEZNANY”. Do wprowadzenia powyższych modyfikacji oraz wprowadzenia rejestracji trybu wykluczenia, konieczne jest opracowanie nowej struktury danych produkcyjnych. W tym celu grupa projektowa opracowała nowy model danych produkcyjnych, gromadzonych w nośniku MOBY. Opracowany model charakteryzuje się dużą otwartością i elastycznością. Zaprojektowana na jego podstawie struktura danych może być implementowana zarówno na linii RM, jak i na innych podobnych liniach montażowych w zakładzie. Do przedstawienia modelu danych produkcyjnych zastosowano diagram klas języka UML (rys. 6.16).

W modelu tym wyróżniono dane procesowe VRM (VRM_Proces) i dane produkcyjne VRM (VRM_Produkt). Dane produkcyjne VRM są to informacje o silniku zawierające status silnika (neutralizacja), informacje nagłówkowe (typ silnika, numer) oraz w przyszłości można tam umieścić informacje o częściach składowych. Dane procesowe są to informacje odzwierciedlające proces montażu. W ramach tych danych zdefiniowano grupy zawierające informacje nagłówkowe, do których można zaliczyć np. czas rozpoczęcia i zakończenia montażu oraz dane związane ze stanem operacji technologicznych (OT). Dane opisujące operacje zawierają dane procesowe OT, które mają bezpośredni wpływ na przebieg procesu montażowego oraz dane parametryczne. Do danych procesowych OT należy zaliczyć status operacji (OK, NO, NIEZNANY) oraz numer wykonywanego w danej operacji programu. Dostęp do danych procesowych musi być szybki ze względu na to, że na ich podstawie podejmowane są decyzje w sytemie zapewnienia jakości. Dla implementacji zaleca się zatem umieszczenie tych danych na jednej stronie pamięci nośnika MOBY. Dane parametryczne zawierają wszystkie dodatkowe informacje związane z operacją. Do tej grupy informacji zaliczają się rozszerzenia związane z rejestrowaniem typów błędów, rejestrowaniem statusu wykluczeń, identyfikatora osoby wykluczającej i wartości pomiarowe.



Rys. 6.16 Obiektowy model danych produkcyjnych [jr]

Wdrażanie nowej struktury danych nośnika MOBY wiąże się jednak ze zmianą oprogramowania wszystkich elementów odwołujących się do danych produkcyjnych. Do elementów tych należy zaliczyć oprogramowanie do przygotowania marszruty technologicznej, oprogramowanie sterowników zarządzających pracą stanowisk montażowych, bramek kontrolnych i stanowisk poprawkowych oraz oprogramowanie do gromadzenia i przetwarzania danych produkcyjnych. Wdrożenie takie nie może być realizowane w etapach. Inwestycja taka wiąże się z dużymi kosztami, przestojami linii montażowej oraz ryzykiem wystąpienia błędów w systemie nadzorowania jakości.

Zastosowanie nowej struktury danych pozwoliłoby na lepsze dopasowanie się do wymogów „Ramowej Specyfikacji Wymagań dla Silników”⁶ oraz rozszerzenie informacji o stanie procesu montażowego i produkowanych silników.

Dzięki modelowi obiektowemu, struktura danych oparta na nowym modelu VRM, mogłaby być rozszerzana w przyszłości o dodatkowe informacje, bez konieczności przeprogramowywania całego systemu VRM. Struktura ta jest również podstawą do konstruowania bazy danych SQL oraz programu do przetwarzania danych produkcyjnych.

6.8 Narzędzie informatyczne wspierające Bazę Wiedzy

Efektywne użytkowanie MOAW w warunkach przemysłowych wymaga wsparcia komputerowego. W celu wyboru odpowiedniego narzędzia stanowiącego platformę Bazy Wiedzy, sformułowano wymagania. Wynik analizy przedstawiono w tabeli 6.1.

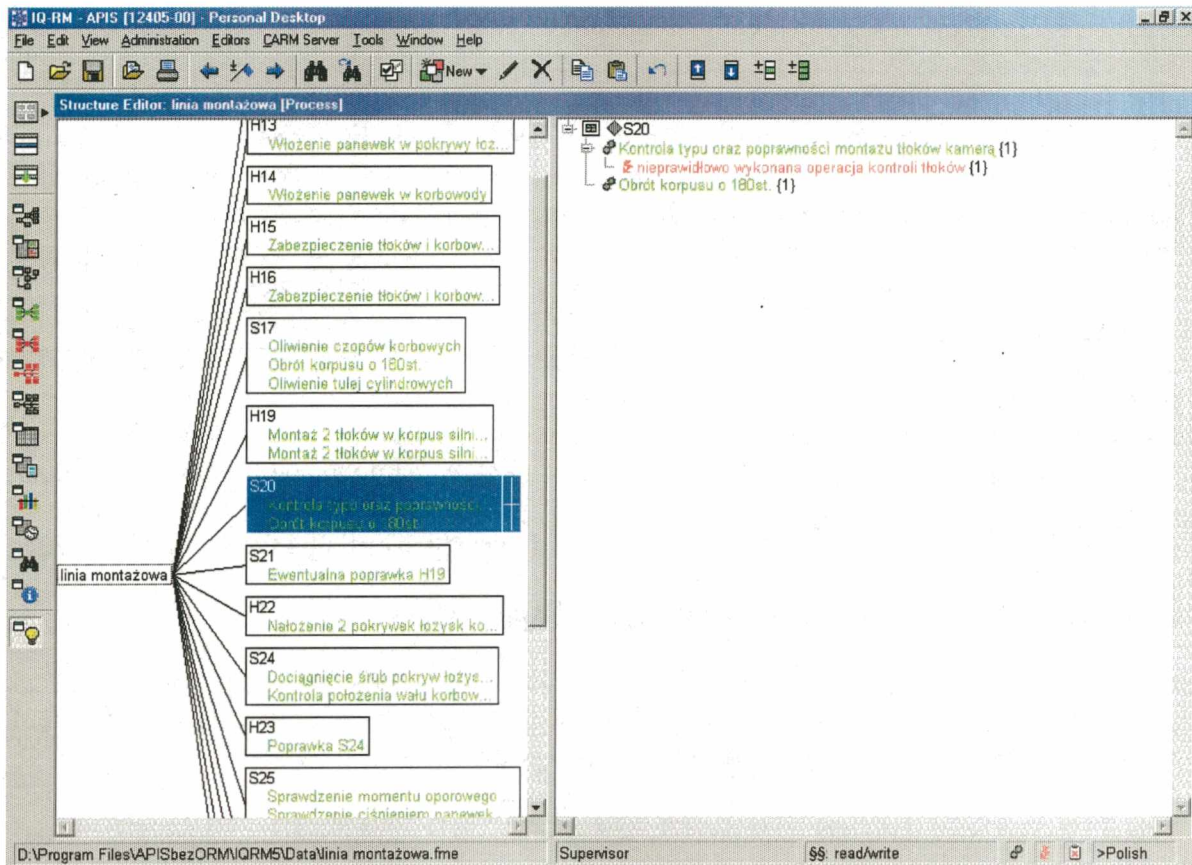
Tab. 6.1 Zestawienie oprogramowania do budowy platformy Bazy Wiedzy

PROGRAM	FMEA-Pro 5	IQ-RM-X 5	Process Analyzer
PRODUCENT	DYADEM	APIS	OMNEX
Kraj	Kanada	Niemcy	U.S.A
Diagram procesu	+	+	+
Analiza FMEA	+	+	+
Analiza Pareto	+	+	?
Statystyki	+	+	+
Dodatkowe analizy i moduły (np: Diagram Ishikawy)	+	+	+
Zgodność z VDA	-	+	-
Wielojęzyczność	?	+	?
Klasa „Baza Wiedzy”	-	+	+
Dostęp do wersji demo	-	+	-
Przedstawicielstwo w Europie	?	+	(+)

Z wymienionych wyżej programów najwyżej został oceniony IQ-RM-X firmy APIS. Oprogramowanie IQ-RM-X jest komercyjnym produktem powstałym jako wynik projektu Baza Wiedzy prowadzonego przez koncern DaimlerChrysler. Firma APIS (Niemcy), producent oprogramowania IQ-RM, rozprowadza je jednak jako platformę do analizy FMEA, a nie platformę Bazy Wiedzy. Wynika to z braku „przepisu” jak w warunkach przemysłowych zbudować Bazę Wiedzy o procesie produkcyjnym.

W ramach projektu RMdata+, opracowano strukturę dla budowania Bazy Wiedzy o linii RM montażu silników w środowisku IQ-RM firmy APIS (rys. 6.17).

⁶ niem. *Rahmenlastenheft* – dokument wymagań obowiązujący w koncernie



Rys 6.17 Struktura Bazy Wiedzy dla linii montażu RM w środowisku IQ-RM f. APIS [jr]

6.9 Ocena Procesu Weryfikacji

Dostępna dokumentacja techniczna linii RM słabo specyfikuje poziom logiczny zastosowanych rozwiązań automatyzacji, zaś wymagania i założenia strategiczne są zupełnie pominięte. Konieczność modyfikacji zautomatyzowanego procesu montażowego, wynikająca z nowych uruchomień produktów oraz nowych wymagań technologicznych, jest powodem zapotrzebowania na wiedzę o użytkowanej linii.

Przeprowadzona weryfikacja miała charakter pilotażowy i demonstracyjny. Obejmowała inżynierię odwrotną (ang. *reverse engineering*), podczas której na podstawie dostępnej dokumentacji, stanu wiedzy członków zespołu i badań na linii RM, tworzone spójne rozumienie problemów, co zostało odpowiednio udokumentowane. Kolejnym krokiem było zaproponowanie nowych rozwiązań, które były dokumentowane za pomocą wprowadzonej metodyki i języka UML. Należy podkreślić, że żaden z uczestników zespołu projektowego wcześniej nie znał języka UML i podejścia obiektowo zorientowanego.

W pierwszym kroku weryfikacji, zgodnie z zaleceniami przedstawionymi w rozdziale 5, zawiązano zespół projektowy. Szczególną uwagę zwrócono na motywację tego zespołu i komunikację między jego członkami. W pierwszym kroku konieczne było wprowadzenie poziomów abstrakcyjnego rozumowania. Zasada taka została wprowadzona bardzo kategorycznie, zabraniając uczestnikom dyskusowania szczegółów implementacyjnych i technicznych charakterystycznych wyłącznie dla ich ról. W ten sposób, omawiając model danych procesowych, nie dyskutowano o bitach i bajtach, a technolog stał się projektantem modelu danych. Dzięki temu zauważono poprawę komunikacji między członkami zespołu o różnych kompetencjach.

Wprowadzenie diagramu biznesowego, który wyróżnił proces rzeczywistego silnika RM i silnika wirtualnego VRM spotkało się z dużą akceptacją pracowników uczestniczących w

realizowanym projekcie. Strategiczne rozumienie obu procesów i potrzeby synchronizacji między procesami zmieniło sposób myślenia o stosowaniu stacji SLS-MOBY. Poprzednio, argument redukcji kosztów był najistotniejszy. Po wyjaśnieniu strategii zautomatyzowanego montażu silników i roli człowieka, proponowane rozwiązania były spójne. Zaproponowano również kilka modyfikacji systemu MOBY dla ujednolicenia systemu.

Opracowanie diagramów klas było odpowiedzią na zalecenie ponownego użycia i wykorzystania, zawarte w zaleceniach wewnętrznych firmy. Podstawową zaletą klasyfikacji obiektowej była wyraźna redukcja problemu i jego strukturyzacja. Przyjęcie obiektowej klasyfikacji, wprowadzone podczas weryfikacji metodyki, było inspiracją dla kierownika jakości do wykorzystania w swojej pracy dyplomowej pt.: *„Analiza FMEA procesu jako sposób myślenia i postępowania na linii montażu bloku silnika w celu doskonalenia jakości”* Politechnika Wrocławska, WCTT 2002, opracowanej w ramach Podyplomowych Studiów Zarządzania Jakością.

Rozszerzony diagram przepływu procesu stał się „platformą” do dyskusji o linii RM. Podczas jego analizy pracownicy sami identyfikowali newralgiczne punkty systemu montażowego i proponowali stosowne rozwiązania. „Model na papierze” stał się katalizatorem i jednocześnie kontrolerem dyskusji, ograniczając jej zakres jedynie do wymaganego obszaru. Brak narzędzia wspierającego modelowanie UML oraz pracę grupową poważnie ograniczył efektywność ich stosowania.

Podczas analizy rzeczywistej i wymaganej funkcjonalności stanowisk i urządzeń technologicznych, diagramy przypadków użycia zostały zaakceptowane i bardzo pozytywnie ocenione przez praktyków. Za ich pomocą przeprowadzono analizę stanowisk kontrolnych z systemami wizyjnymi. Zauważono dużą przydatność diagramów przypadków użycia dla specyfikacji wymagań, zobowiązań oraz specyfikacji odbiorów maszyn (patrz kryteria sukcesu klient – dostawca).

Specyfikacja zachowania prowadzona była z wykorzystaniem uproszczonej syntaktyki diagramów stanów, które wiążą kod oprogramowania i działanie maszyn.

„ ... właśnie w taki sposób chciałbym się dowiadywać, co zostało przeprogramowane w sterowniku maszyny! A nie być zbywanym, że oprogramowano kolejny blok OB.” - [odpowiedzialny za technologię montażu]

Zdanie powyższe wskazuje na akceptację przedstawionych narzędzi MOAW przez nie-automatyków. Automatyzacja jednak nie może być nadrzędnie traktowana w stosunku do technologii.

Opracowywanie rozszerzonego modelu danych produkcyjnych było prawdziwą rewolucją intelektualną dla uczestników projektu. Po raz pierwszy byli oni pytani o potrzebne dane, a jednocześnie barierą nie był sposób organizacji pamięci MOBY, czy też bazy danych gromadzącej dane produkcyjne. Zaproponowany model jest rozwiązaniem poziomu strategicznego i musi zostać zmodyfikowany o właściwości i ograniczenia dostępnych nośników pamięci. Mimo, że model został opracowany zgodnie z podejściem obiektowym, to jego implementacja może być prowadzona w sposób dowolny.

Praktyczne zastosowanie powyższego modelowania wymaga jednak narzędzi wspierających modelowanie UML i pracę grupową. Celem większości dostępnych obecnie narzędzi modelowania UML jest głównie wsparcie generacji kodu w językach C++ czy Java, etc. Poprzez rozbudowane opcje konfiguracyjne i oferowane możliwości parametryzacji są one skomplikowane w użyciu i drogie. Dlatego też są one nieprzydatne do zaproponowanego powyżej zastosowania.

W ramach projektu pilotażowego wybrano narzędzie informatyczne IQ-RM i opracowano strukturę dla budowy Bazy Wiedzy o zautomatyzowanym procesie produkcyjnym RM.

7 Wnioski

Przedstawiona praca jest wynikiem zauważenia szczególnej roli człowieka w automatyzacji wytwarzania. Bezpieczne i efektywne automatyzowanie procesów wytwórczych, z wykorzystaniem najnowszych zdobyczy techniki wymaga szczególnej wiedzy i kompetencji technicznych i społecznych od użytkowników.

Przy formułowaniu wyników niniejszej pracy korzystano z interdyscyplinarnego podejścia do problemu, tj. inżynierijsko – humanistycznego. Doświadczenia wielu praktyków zestawiono z wiedzą teoretyczną z zakresu automatyzacji, technologii produkcji, jakości, ale również psychologii, socjologii, a nawet filozofii. Zaproponowane rozwiązanie MOAW „Modelowanie Obiektowe dla Automatyzacji Wytwarzania” jest oryginalnym narzędziem zarządzania wiedzą, opracowanym w ramach niniejszej pracy. Sformułowano dla niego niezbędne podstawy teoretyczne, zakres kompetencji, jak również proces wdrożenia. Weryfikacja przeprowadzona w zakładzie przemysłowym przemysłu motoryzacyjnego potwierdziła jego przydatność przemysłową. Ocena efektów wprowadzenia tego narzędzia wymaga „zmiany kultury pracy” i nie może być dokonana w krótkim czasie, stąd potrzeba dalszego rozwoju i upowszechnienia przedstawionego rozwiązania.

7.1 Cele i teza pracy doktorskiej

Na podstawie studium przeprowadzonego w rozdziale 1, wskazano, że spełnienie rozległych wymagań klienta wobec produktu, wymaga zastosowania wyrafinowanych rozwiązań automatyzacji. Stosowanie nowoczesnych sterowników PLC, komputerów przemysłowych, sieci komunikacyjnych, wymaga coraz bogatszej wiedzy od projektantów. Wiedza ta obejmować musi zagadnienia związane z rozwiązaniem, ale również wymagania technologiczne procesu. Wskazano, że wzrost udziału oprogramowania w nowoczesnych zautomatyzowanych systemach wytwórczych, nie jest należycie wspierany przez inżynierię oprogramowania. Dlatego komunikacja wiedzy o automatyzacji między członkami grupy projektowej, wskazana została jako źródło poprawy efektywności automatyzacji.

W rozdziale 2 przyjęto główne zadania i cele dla realizowanej pracy doktorskiej.

Opracowanie narzędzia zarządzania wiedzą obejmującego założenia organizacyjne i środki techniczne.

W rozdziale 3 dokonano analizy procesu automatyzacji w ujęciu wiedzy, której wnioski wskazały na przydatność modelowania dla podniesienia jej efektywności. Analiza aktualnie stosowanych technik modelowania w automatyzacji (bloki funkcyjne) wykazała ich nieefektywność, gdyż tkwią w paradygmacie elektrotechnicznym. Jako rozwiązanie, w rozdziale 4, zaproponowano paradygmat obiektowy i język UML będący standardem modelowania obiektowego. Propozycja ta jest oryginalnym zastosowaniem języka UML dla komunikacji między nie-informatykami (technologami) i nie-technologami (automatykami) realizującymi wspólnie zadania automatyzacji. W pracy przedstawiono przykłady wykorzystania UML dla modelowania Elastycznego Gniazda Montażowego w Instytucie Technologii Maszyn i Automatyzacji Politechniki Wrocławskiej. Do tego modelowania wykorzystano środowisko Rhapsody firmy iLogix. Zaproponowane rozwiązanie, tj. „Modelowanie Obiektowe dla Automatyzacji Wytwarzania” (MOAW) jest narzędziem zarządzania wiedzą.

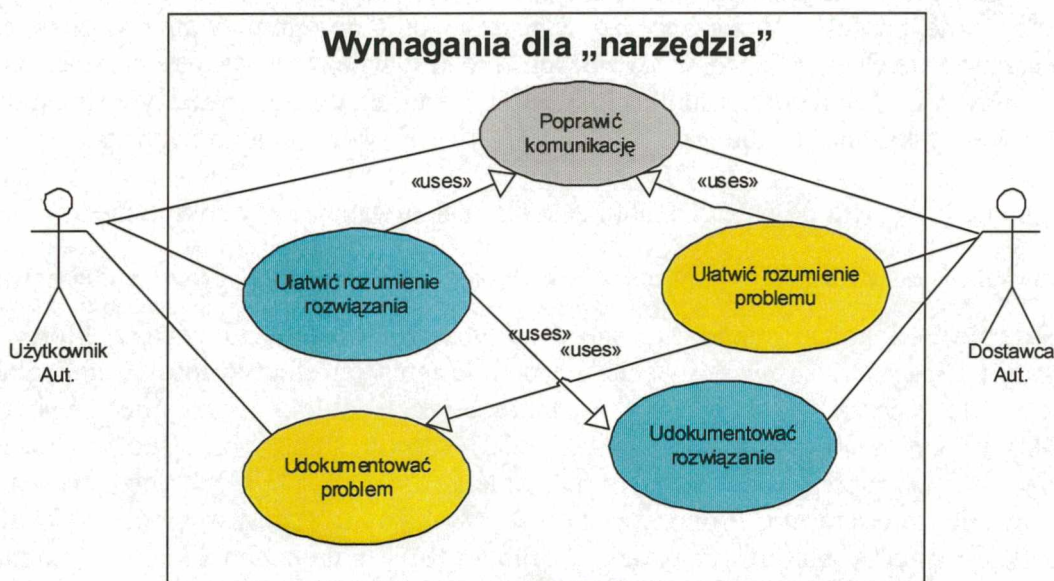
Omówienie procesu wdrażania opracowanego narzędzia i jego utrzymania.

Efektywność środków organizacyjno-technicznych, z których korzysta człowiek, czy zespół, zależy w głównej mierze od metodyki ich wdrażania i późniejszego utrzymania. Dlatego tak ważne jest rozumienie „filozofii” tego narzędzia. W rozdziale 5 zaproponowano szkieletową metodykę wdrażania MOAW, obejmującą budowę zespołu i motywacji, trening, pilotażowe uruchomienie, skalowanie oraz użytkowanie i utrzymanie. Zdefiniowano również pojęcie Bazy Wiedzy oraz podano kryteria doboru platformy software’owej.

Poddanie weryfikacji przemysłowej opracowanego narzędzia oraz metodyki jego wdrażania.

Weryfikacja została przeprowadzona w zakładach motoryzacyjnych w ramach projektu „RMdata+”, w zespole odpowiedzialnym na utrzymanie i rozwój linii montażu kadłubów silnika. Przebieg i wyniki tych prac zostały przedstawione w rozdziale 6. Zrealizowany projekt obejmował proces treningowy oraz wdrożenie pilotażowe. W jego wyniku zainicjowano zmiany w „kulturze pracy” oraz opracowano modele klas czynności, błędów, stanowisk, rozszerzony model przepływu procesu, modele interakcji, modele stanów i modele przypadków użycia, włączone do dokumentacji linii montażowej. Z przyczyn nie-technicznych, w ramach projektu nie udało się wprowadzić narzędzia wspierającego zaproponowaną metodykę oraz zaimplementować ją jako obowiązującą w całym zakładzie. Jednak akceptacja pracowników uczestniczących w weryfikacji, wykorzystanie tej metodyki w pracy dyplomowej jednego z jej uczestników oraz w specyfikacji wymagań, wskazują, że będzie ona znajdowała tam rozwijana i będzie znajdowała coraz szersze zastosowanie.

Dla opracowywanego narzędzia MOAW, w rozdziale 2 wyspecyfikowano pięć głównych wymagań, co przedstawia diagram rys. 7.1.



Rys. 7.1 Wymagania dla narzędzia zarządzania wiedzą dla automatyzacji [jr]

Krótką charakterystyką osiągnięcia każdego z tych wymagań została przedstawiona poniżej:

Poprawić rozumienie automatyzacji przez użytkowników pracujących z systemami wytwórczymi.

W pracy zaproponowano modelowanie biznesowe automatyzacji, które przybliży użytkownikom strategię systemu, tj. cele, zasoby, wejścia i wyjścia, czynności i ich kolejność. Zaproponowano również język modelowania obiektowego, do specyfikowania wymagań funkcjonalnych, struktury i zachowania. Specyfikacje opracowane za pomocą modeli obiektowych są „bliższe” rozwiązaniu. Dzięki temu, ich weryfikacja na poziomie wizji i ostatecznego rozwiązania, jak również ich porównanie ze specyfikacją są łatwiejsze. Wykazano, że modelowanie obiektowe i UML mogą być stosowane także przez nie-automatyków i nie-programistów.

Poprawić rozumienie wymagań technologii i jakości systemów wytwórczych przez dostawców rozwiązań automatyzacji.

Dzięki specyfikowaniu wymagań funkcjonalnych, struktury i zachowania za pomocą modeli obiektowych i UML, są one precyzyjniejsze, a ich rozwijanie nie wymaga transformacji, która w sposób stratny (podobnie jako kompresja danych MPEG) przekształca informacje. Dzięki MOAW dostawca współpracuje z klientem, którego świadomość techniczna jest wyższa, a język komunikacji lepiej dostosowany do dziedziny rozwiązania, co powinno zaowocować efektywniejszym przebiegiem procesu automatyzacji i produktami o wyższej jakości.

Poprawić pracę w interdyscyplinarnych grupach użytkowników zautomatyzowanych systemów wytwórczych.

Dzięki wprowadzeniu paradygmatu obiektowego i języka UML (narzędzie MOAW) zauważono poprawę w komunikacji wewnętrznej między członkami zespołów opiekujących się systemami wytwórczymi. Z jednej strony nie-automatycy zostali przeszkoleni do używania narzędzi abstrakcji oraz określonego formalizmu, a drugiej automatycy i programiści otrzymali narzędzie dla abstrakcyjnego wyrażania języków programowania.

Poprawić komunikację użytkownik – dostawca rozwiązań automatyzacji.

Modelowanie obiektowe i UML, szczególnie specyfikacja wymagań funkcjonalnych, struktury i zachowania, zostały wykorzystane do komunikacji ze zleceniodawcą. Opracowana specyfikacja wymagań stanowi również podstawę oceny wyników projektów automatyzacji.

Poprawić jakość dokumentacji o wymaganiach i rozwiązaniu automatyzacji.

Modelowanie obiektowe i UML generują zapisy w postaci diagramów o sformalizowanej syntaktyce i semantyce. Diagramy te stanowią dokumentację stanu aktualnego, planowanych zmian i stanu wcześniejszego. Opracowanie diagramów UML może być wspierane poprzez różne narzędzia, poczynając od wspomagania diagramowania aż po systemy z bazami wiedzy.

W wyniku przeprowadzonej rozprawy doktorskiej dowiedziono zatem słuszności postawionej tezy, że: **"Poprawę efektywności automatyzacji można osiągnąć dzięki odpowiednim kompetencjom w zakresie modelowania obiektowego i języka UML oraz poprzez odpowiednie zarządzanie nimi"**

7.2 Wyniki naukowe

Poniżej przedstawiono w postaci syntetycznej najistotniejsze wyniki poznawcze wynikające z przedstawionej pracy doktorskiej.

7.2.1 Automatyzacja wytwarzania

- Zamodelowano proces automatyzacji w kontekście WIEDZY,
- Scharakteryzowano znaczenie trzech klas produktów automatyzacji,
- Wskazano na potrzebę zmiany zakresu kompetencji Działu Utrzymania Ruchu
- Zidentyfikowano i zamodelowano interakcje klient-dostawca,
- Wskazano na „miękkie” wskaźniki efektywności automatyzacji,
- Zidentyfikowano interfejsy między etapami automatyzacji jako źródła poprawy jej efektywności,
- Wykazano potrzebę interdyscyplinarności dla efektywnego rozwiązywania problemów technicznych automatyzacji (filozofia, psychologia, socjologia).

7.2.2 Narzędzie zarządzania wiedzą

- Przeanalizowano łańcuch komunikacyjny,
- Wskazano na rolę paradygmatu i języka dla komunikacji wiedzy,
- Wskazano na nieefektywność paradygmatu i języków graficznych w automatyzacji dla komunikacji wiedzy o automatyzacji,
- Zaproponowano Modelowanie Obiektowe i język UML dla Automatyzacji Wytwarzania (MOAW) - jako narzędzie zarządzania wiedzą dla automatyzacji.

7.2.3 Metodyka wdrażania narzędzia wiedzy MOAW

- Zaproponowano model wdrażania narzędzia MOAW,
- Wskazano na czynniki motywacji zespołu,
- Przeanalizowano problematykę pracy grupowej,
- Scharakteryzowano przebieg treningu MOAW,
- Sformułowano cechy Bazy Wiedzy,
- Wskazano na subiektywne kryteria sukcesu projektu wiedzy,
- Zidentyfikowano wymagania i zagrożenia podczas wdrażania MOAW,

7.3 Wyniki uytylitarne

Poniżej zestawiono najistotniejsze wyniki praktyczne będące efektem realizacji pracy doktorskiej.

7.3.1 Narzędzie zarządzania wiedzą dla poprawy automatyzacji wytwarzania

Opracowane narzędzie MOAW obejmuje gruntowne podstawy teoretyczne dla komunikacji w interdyscyplinarnych zespołach automatyzujących wytwarzanie oraz pakiet kompetencji z zakresu modelowania obiektowego i UML. Podstawy teoretyczne stanowią „filozofię”

zaproporowanego narzędzia, która jest niezbędna w celu jego prawidłowego i efektywnego wykorzystania.

7.3.2 Sposób wdrożenia narzędzia zarządzania wiedzą

Cennym dorobkiem pracy jest również metodyka proponująca sposób wdrożenia opracowanego narzędzia MOAW do praktyki przemysłowej. Metodyka ta nie jest jednak „jedynym dobrym” sposobem wdrożenia, ale stanowi raczej szkielet z omówieniem krytycznych faz procesu, takich jak budowa zespołu, motywacja, ocena wyników. Problemy te są szczególnie ważne w takiej sytuacji, gdy biznesowo-ekonomiczne uzasadnienie projektów zarządzania wiedzą jest niemożliwe.

7.3.3 Weryfikacja - Projekt „RMdata+”

W ramach przeprowadzonego projektu przemysłowego dokonywano weryfikacji zaproponowanego narzędzia MOAW i metodyki jego wdrażania. Osiągnięto następujące wyniki:

- Zamodelowano proces montażowy kadłubów silnika,
- Opracowano model obiektowy danych produkcyjnych dla linii montażu silników,
- Wyszpecyfikowano wymagania JEST/MA BYĆ dla wybranych stacji montażu (UML),
- Zamodelowano zachowanie JEST/MA BYĆ dla wybranych stacji montażu (UML),
- Przeszkolono grupę pracowników dla użycia MOAW w celu poprawy efektywności użytkowania i rozwoju zautomatyzowanej linii montażowej,
- Zweryfikowano możliwości zastosowania oprogramowania dostępnego na rynku dla budowy Bazy Wiedzy,
- Sformułowano założenia dla narzędzia informatycznego wspomagającego MOAW w badanym zakładzie przemysłowym.

7.4 Propozycje dalszych prac

W celu dalszego rozwoju zaproponowanego narzędzia wiedzy MOAW konieczne jest opracowanie lub dostosowanie oprogramowania wspierającego. Oprogramowanie to, jak przedstawiono w rozdziale 5, powinno tworzyć Bazę Wiedzy o systemie produkcyjnym.

Literatura

- [Arena 00] <http://www.arenasimulation.com/>
- [Belbin 96] Belbin, M., *Management Teams- why they succeed or fail?*, Butterworth-Heinemann 1996
- [Brussel, et al. 98] Van Brussel, H., Valckenaers, P., Bongaerts, L., Lyns, J., *Architectural and System Design Issues in Holonic Manufacturing Systems*, Distributed Enterprise, Intelligent Automation and Industrial Benchmarking, Proceedings of the ESPRIT-COPERNICUS Symposium, Wrocław Poland, 23-24 April 1998
- [Burton 94] Burton, Ch., *The Strategic use of psychological testing within a team context*, <http://members.tripod.lycos.co.kr/jbk/m-test.htm> D181
- [Buzan 89] Buzan, T., *Use Both Sides of Your Brain* by Tony Buzan 1989
- [Buzan 91] Buzan, T., *The mind map book*. New York: Penguin 1991
- [Buzan, Buzan 93] Buzan, T., Buzan, B. *The Mind Map Book How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential*, Penguin Group/Dutton 1993 ISBN 0-525-93904-0
- [Cempel web] Cempel, C., *Nowoczesne Zagadnienia Metodologii i Filozofii Badań*, e-Skrypt, <http://neur.am.put.poznan.pl/index.htm>, 2003
- [cetus web] <http://www.cetus-links.org/>
- [Chmielowski] Chmielowski, J.B.Ks., *Pierwsza polska encyclopedia*
- [Cierpisz 02] Cierpisz, T., *Subiektywna akceptacja jako pragmatyczna miara sukcesu*, VI Konferencja Project Management - To Ludzie robią projekty, 16-17 Października 2002, Gdańsk – SPMP s.65-72
- [cio web] <http://www.cio.com>
- [Coad 93] Coad, P., Nicola, J., *Programowanie Obiektowe*, Oficyna Wydawnicza READ ME, Warszawa 1993, *Object-Oriented Programming*, Prentice Hall 1993
- [ConceptDraw] ConceptDraw www.conceptdraw.com/en/
- [Cook, Daniels 94] Cook, S., Daniels, J., *Designing Object Systems. Object-Oriented Modelling with Syntropy*, Prentice Hall 1994
- [CTR 96] CTR, Computer Technology Research Corporation, Knowledge Management Report 1996
- [Ctrl. Eng. 00] Control Engineering, *MicroPLCs – more popular than ever*, <http://www.controleng.com/archives/2000/ctl0702.00/e0007micro.htm>
- [Ctrl. Eng. 99] News & Analysis, *Control Technology Receives Patent for Object-Oriented Programmable Controller*, 11/30/99, Plant Automation.com: Virtual community for the manufacturing automation industry, Control Engineering, <http://www.manufacturing.net>
- [CXO 03] <http://www.cxo.pl/artykuly/29930.html>
- [Douglas 99] Douglas, B.P., *The UML for System Engineering*, 1999, <http://www.ilogix.com>
- [Druckner 93] Druckner, P.F., *Post Capitalist Society*, Butterworth Heinemann, Oxford, 1993
- [EFQM web] <http://www.efqm.org/>
- [Elmuti 97] Elmuti, D., *The Perceived impact of team-based management systems on organisational effectiveness*, Team Performance Management, Vol 3, No 3, 1997, pp 179-192, Elsevier.
- [EN50170] General Purpose Field Communication System Norma Europejska obejmująca: PROFIBUS, P-Net, Word-FIP
- [EN50245] Norma Europejska obejmująca standard Interbus-S
- [EPSSSES] Raport z projektu: Entwicklung, Produktion und Service von Software für eingebettete Systeme in der Produktion (*Opracowanie, produkcja i serwisowanie oprogramowania dla systemów wbudowanych dla produkcji*), <http://www.broy.informatik.tu-muenchen.de/proj/va/>

- [Eriksson, Penkner 00] Eriksson, H.E., Penkner, M., *Business Modelling with UML: Business Patterns at Work*, Jon Wiley & Son, Inc, 2000
- [Ettinger 01] Ettinger, H., *Build your project team first, your new plant second*, RBECCompany Lawrence <http://www.rbeco.com/team.htm>
- [Eversheim, et al. 02] Eversheim, W., et al.: *Mit e-Engineering zum i3- Engineering*. Conference Proceedings, AWK 2002, Wettbewerbsfaktor Produktionstechnik - Aachener Perspektiven.
- [Farnschlaeder et al. 03] Farnschläder, F., Bimazubute, R., Pfeiffer, U., Stoyan, H., *Bewertung von Wissensmanagementprojekten nach dem Vorbild der Bewertung von EDV, Marketing und Personalwesen*, WM 2003 Luzern
- [Filemonowicz 01] Filemonowicz, K., *Elementy metodyki realizacji przedsięwzięć najczęściej wykorzystywane przez zespoły projektowe*, VI Konferencja Project Management - To Ludzie robią projekty, 16-17 Października 2002, Gdańsk – SPMP, s73-82
- [Freudenthaler et al. 03] Freudenthaler, K., Eisenhauer, R., Stenger, H.J., Borghoff, U.M. *Entwicklung eines KM Framework und Implementation Guide*, WM2003” Professionelles Wissensmanagement – Erfahrungen und Visionen, [WM 03]
- [Gronau 03] Gronau, N., Palmer, U., Schulte, K., Winkler, T., *Modellierung von wissensintensiven Geschäftsprozessen mit der Beschreibungssprache K-Modeller*, [WM 03]
- [Hofman 97] Hofman, P., *Einfluss objektorientierter Techniken auf den Entwicklungsprozess von eingebetteten Systemen im Automobil*, Daimler-Benz, 1997
- [Horvath 01] Horvath & Partner (Hrsg.), *Balanced Scorecard umsetzen*, Schäffer-Poeschel Verlag Stuttgart 2001
- [HP] Freurer, R., Chaharbaghi, K., Wargin, J., *Developing creative teams for operational excellence*, International Journal of Operations & Production Management, vol. 16, no. 1, 1996
- [Humphries, et al. 98] Humphries, N., Gare, C., Dawson, L., D., *FMEA in design, manufacture and service or Murphy's law overruled* : Victoria:Productivity Strategies, 1998
- [HyperKernel] <http://www.nematron.com/Software/HyperKernel.shtml>
- [IDC 02] <http://www.idc.com/>
- [IDEF web] IDEF Family of Methods <http://www.idef.com/> 03.2001
- [IEC61131] Norma międzynarodowa programowania sterowników PLC
- [IEEE94] IEEE Standard 830-1993: *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Standards Collection, Software Engineering, 1994
- [Inspiration] Inspiration www.inspiration.com
- [InTime] <http://www.intime-online.com/>
- [iwb 03] Münchener Kolloquium 27/28 Luty 2003 <http://www.mueko.tum.de>; 2003
- [Jacobson, et al. 95] Jacobson, I., Ericsson, M., Jacobson, A., *The Object Advantage: Business Process Reengineering with Object Technology*. Addison-Wesley 1995
- [Jennings, Haughton 02] Jennings, J., Haughton, L., *Szybkość jako atut w biznesie. To nie duzi zjadają małych, ale szybcy opieszających*, MT Biznes 2002
- [JetWeb] <http://www.jetweb-technology.com/>
- [Kamiske web] Kamiske, G.F., *Der Weg zur Spitze*, Hanser Verlag http://www.deming.de/iso9000/iso_tqm3.html
- [Kiess 95] Kiess, J., *Objektintegrierte Modellierung von Automatisierungssystemen*. Software Engineering für Embeded Systems, Springer 1995.
- [Koch et al.. 00] Koch, J., Mrzygłód, M., Reiner, J., Skura, K., Smalec, Z. *Kierunki rozwoju systemów komunikacyjnych w automatyce przemysłowej*. Oficyna Wydaw. PWroc. 2000

- [Koch, et al. 97a] Koch, J., Reiner, J., Skura, K., Smalec, Z., Trzciński, P., *Neue Tendenzen in der Entwicklung von Kommunikationsnetzen in der Produktionsautomatisierung*, Proceedings of the International Conference on Computer Integrated Manufacturing CIM'96, Zakopane 12-14 May 1996, Vol. III s.215-222
- [Koch, et al. 98] Koch, J., Smalec, Z., Reiner, J., Mrzygłód, M., Trzciński, P. *Rozproszony system sterowania gniazdem wytwórczym*, 1998
- [Koch, et al. 99] Koch, J., Smalec, Z., Reiner, J., Skura, K., *Communication System in Manufacturing and their role in automation.*, Human Systems Management 1999 vol. 18 nr 3,4
- [Koch, T., 01] Koch, T., Lean Manufacturing czyli jak wdrażać konkurencyjność, www.lean.org.pl
- [Koch, Weck 96] Koch, T., Weck, M., *COSMOS an Open Architecture for Discrete Cell Control*, Manufacturing Systems, Vol.25, 1996, No.1, S.17-21.
- [Kosmol, et al. 00] Kosmol, J., Mrzygłód, M., Reiner, J., Skura, K., Smalec, Z., *Układy sterowania Maszyn – Sterowniki PLC i CNC*, Automatyzacja Produkcji 2000, Wiedza Technika Postęp, Tom 1. Referaty plenarne, Wrocław 2000
- [Kossowska 01] Kossowska, M., *Ocena i rozwój umiejętności pracowników*, Akade, Kraków 2001
- [MBS 98] Microsoft Bookshelf 98 (Software on CD-ROM)
- [McDermott, et al. 96] McDermott, R.,D., Mikulak, R., J., Beauregard, M., R., *The basics of FMEA*, Portland : Productivity, 1996.
- [Mikulczyński Samsonowicz 97] Mikulczyński, T., Samsonowicz, Z., *Automatyzacja dyskretnych procesów produkcyjnych* Wydawnictwa Naukowo-Techniczne, W-wa 1997
- [Mikulecky web] Mikulecky, D.C., Definition of Complexity, Professor of Physiology, Medical College of Virginia Commonwealth University <http://views.vcu.edu/~mikuleck/>, 2003
- [Mindjet 02] Improving Project Success Through Efficient Front-End Planning, White Paper 2002, Mindjet
- [MindManager] MindManager 2002 Enterprise Edition www.mindjet.com
- [MindMapper] Mind Mapper <http://www.mindmapperusa.com/>
- [MoWiMa97] http://www.isw.uni-stuttgart.de/projekte/mowima/index_d.htm
- [Mrzygłód, et al., 97] Mrzygłód, M., Reiner, J., Smalec, Z., Trzciński, P., *Zagadnienia komunikacji w sieci PROFIBUS na przykładzie wybranych rozwiązań sprzętowych i programowych*, Pomiar, Automatyka, Robotyka, PIAP, 5/6 1997
- [Nieścior web] Nieścior, A., *Refleksje pokonferencyjne, czyli odgłupianie doskonałości*, <http://www.kadry.info.pl/artykuly/4308.htm>
- [Nonaka, Takeuchi 00] Nonaka, I., Takeuchi, H., *Kreowanie wiedzy w Organizacji*, Wyd. Poltext, Warszawa 2000
- [Nosal 89] Nosal, A., *Psychologiczne modele umysłu*, Wydawnictwa Szkolne i Pedagogiczne Warszawa 1989.
- [OSACA 97] *OSACA-Handbook* – ESPRIT IV Draft version 06.1997. <http://www.osaca.org>
- [PLCopen] <http://www.plcopen.org/>

- [Płoski 99] Płoski, Z., Słownik Encyklopedyczny – Informatyka, Wydawnictwa Europa. 1999.
- [PMBOK 00] Project Management Institute, *A Guide to the Project Management Body of Knowledge*, PMBOK Guide 2000 Edition
- [PNO 01] Materiały marketingowe PNO, PROFInet www.profibus.com
- [Popadiuk 02] Popadiuk A., *Coaching – nowa filozofia przywództwa*, <http://www.kadry.info.pl/biblioteka/>
- [PP web] Profesjonalny Portal Psychologiczny, <http://www.psychologia.edu.pl>
- [Pree 97] Pree, W., *Component-Based Software Development – A New Paradigm in Software Engineering?*, *Software – Concepts and Tools*, Springer-Verlag, 1997 no.18, s.167-174
- [Pritschow 00] Pritschow, G., Waælde, K., Uebersicht ueber offene Prozessschnittstellen - Zukunftsperspektiven, Schnittstellen in der Produktionstechnik, Nr 593, Fortschritt-Berichte VDI
- [Pritschow Daniel 96] Pritschow, G., Daniel, G., *Open Control System – A Future-Oriented Concept*, *Manufacturing Systems*, vol.25, 1996, No.1, s.9-16
- [Probst, et al. 97] Probst, G., Raub, S., Romhardt, K.: *Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. Frankfurt am Main, Gabler, 1997
- [PWN] Encyklopedia PWN © Wydawnictwo Naukowe PWN S.A. <http://encyklopedia.wp.pl> , 2003
- [QSS 98] Quality Systems and Software, *DOORS Questions & Answers*, 1998, Why are requirements important for quality, s.2 <http://qssinc.com>
- [Rational 01] Rational Software, Rational Unified Process, Best Practices for Software Development Teams, White Paper, 01/2001, www.rational.com , 2003
- [Reiner et al. 03] Koch, J., Krebs, I., Reiner, J., Schnabel, S., Siech, T., *Knowledge Management Issues for Maintenance of Automated Production Systems*, Human Aspects in Production Management, Conference proceedings, Karlsruhe, October 2003 (przyjęto do druku i ogłoszenia)
- [Reiner, et al. 00] Reiner, J., Mrzygłód, M., Koch, J., Smalec, Z. *Obiektowo zorientowane modelowanie sterowania dla zatoki montażowej w środowisku Rhapsody. Automatyzacja - nowości i perspektywy. Automation '2000. 2000*
- [Reiner, Koch 00] Reiner, J., Koch, J., *Object Oriented Paradigm for Manufacturing Automation*, IFAC Intelligent Manufacturing Systems 2001, A Proceedings volume from the 6th IFAC Workshop, Poznań, Poland 24-26 April 2001, Pergamon 2001
- [Reiner, Koch 03] Reiner, J., Koch, J. *Cooperation problems at the interdisciplinary teams of providers and users in manufacturing automation*. Oficyna Wydawnicza Politechniki Wrocławskiej 2003
- [Reiner, Koch, 98] Reiner, J., Koch, J., *Rozproszone sterowanie dla gniazd wytwórczych*, Prace Instytutu Technologii Maszyn i Automatyzacji w jubileuszowym roku 1998 w zakresie technologii maszyn. Posiedzenie sekcji PAN – KBM, Wrocław 1998, Prace ITMA nr 71, Konferencje nr 32, s.163-168
- [Reinhart, et al. 99] Reinhart, G., Bauer, L., Meier, H., Wagner, P., Weissenberger, M., *Vernezte Entwicklung komplexer mechatronischer Produkte*“ ZWF 4 1999, Carl Hanser Verlag, München, s 191-194

- [Reinner 99] Reiner, J., *Zastosowanie uniwersalnych modułów wejść/wyjść WAGO I/O SYSTEM dla sieci Fieldbus to źródło oszczędności w nowoczesnej automatyzacji*, Pomiary Automatyka Kontrola nr 3 1999, s. 9
- [Reuters 00] Reuters, *Windows 2000: Nobody's Perfect*, 17.02.2000, (3:30PM) <http://www.cmpnet.com>
- [RFG 02] Total Cost of Ownership for Linux in the Enterprise <http://www.rfgonline.com>
- [Rhapsody 00] I-Logic, *Rhapsody Reference Guide*, Release 2.3, I-Logic Inc
- [RMdata+] Raport z projektu przemysłowego, WCTT Politechnika Wrocławska, 2002
- [Rockwell 97] Rockwell International Corporation, *Automation Investment Life Cycle, CHOICES 1997 A White Paper*, <http://www.rockwell.com>
- [Rockwell 97b] Control & Information Group of Rockwell Automation, *CHOICES: Perspectives on the Future Automation Control*, Real-Time Magazine 1997, no4, s16-30
- [Rother et al. 99] Rother, M., Shook, J., Womack, J., Jones, D., *Learning to see*, Lean Enterprises Institute 1999
- [RTX] <http://www.vci.com/index.asp>
- [Seel 00] Seel, W., *Die Simatic-Kopie*, Computer & Automation, 7-8 2000, s. 44-46
- [Senge 98] Senge, P., *Piąta dyscyplina – teoria i praktyka organizacji uczących się*, Wydawnictwo ABC, Warszawa 1998
- [SEofP 02] Stanford Encyclopedia of Philosophy 2002 – Diagrams
- [SGML] Standard Generalized Markup Language, <http://www.oasis-open.org/cover/sgml-xml.html>
- [Shannon, et al. 99] Shannon, B., Yarow, R., Jonson, L.J., *UML can save your next project*, Enterprise Development, January 1999, <http://www.uml-zone.com>
- [Shen Norie] Shen, W., Norrie, D., *Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art. Survey*
- [Siemens 96] Siemens, SIMATIC Software, System Software for S7-300 and S7-400, Program design, Programming Manual, Siemens AG 1996
- [Skura 02] Skura, K., Analiza i badania wybranych rozwiązań systemów komunikacyjnych do automatyzacji procesów przemysłowych, Praca doktorska, Politechnika Wrocławska, Instytut Technologii Maszyn i Automatyzacji, 2002
- [Smalec et al. 02] Smalec Z., Mrzygłód M., Reiner J., Szeplik P., *Open Control Systems and Examples of Slot- and Soft-CNC*; Conference Proceedings: Modern Trends in Manufacturing, First International CAMT Conference, 7-8 February 2002, Wrocław
- [Smalec et al. 97] Smalec, Z., Skura, K., Mrzygłód, M., Reiner, J., Trzciński, P., *Sieciowe systemy komunikacyjne integrujące automatyzacje wytwarzania. Sprawozdanie końcowe z realizacji projektu*. Raporty Inst. Technol. Masz. Automat. PWroc. 1997
- [Smith 99] Smith, D.A., *The fourth industrial revolution: Objects replace synchronicity*, I&CS Control Software Forum, August 1999, s59-60
- [Smolski, et al. 99] Smolski, R., Smolski, M., Stadtmüller, E.H., *Słownik Encyklopedyczny Edukacja Obywatelska* Wydawnictwa Europa, 1999

- [Sparks 00] Sparks, G., *An introduction to modelling software systems using the Unified Modelling Language: The Business Process Model*, 2000
http://www.sparxsystems.com.au/WhitePapers/The_Business_Process_Model.df
- [Sperling Lutz 96] Sperling, W., Lutz, P., *Enabling Open Control Systems – An Introduction to the OSACA Platform*, Robotisc and Manufacturing, Volume 6, ISRAM'97, ASME Press New York 1996
- [Storr 95] Storr, A., *Software Engineering for Control Technology – Definitions and Requirements*, Software Manufacturing Systems, Methods and CASE tools, IFIP TC5 international conference on Software Engineering for Manufacturing Systems 28-29 March 1996, Stuttgart Germany S1-13.
- [Strosłup 96] Stroustrup, B., *Projektowanie i Rozwój Języka C++*, Wydawnictwa Naukowo Techniczne, Warszawa 1996
- [Sturz web] Sturz, W., *Der Faktor Mensch: Wissensmanagement im Spannungsfeld zwischen Kultur und Technik*, Steinbeis Transferzentrum für Wissensmanagement & Kommunikation, 2003 <http://www.steinbeis-wissensmanagement.de/index.html>
- [Subieta 98] Subieta, K., *Wprowadzenie do obiektowości cz. 1 i cz. 2*, Instytut Podstaw Informatyki PAN Warszawa, Polsko-Japońska Wyższa Szkoła Technik Komputerowych, Warszawa 1998.
- [Subieta 99] Subieta, K., *Słownik terminów z zakresu obiektowości*, Akademicka Oficyna Wydawnicza, Warszawa 1999
- [SWO web] Słownik Wyrazów Obcych, Wydawnictwa Europa, pod redakcją naukową prof. Ireny Kamińskiej-Szmaj, autorzy: Mirosław Jarosz i zespół, 2001.
- [Tannenbaum 96] Tannenbaum, S. I., Salas, E. & Cannon-Bowers, J. A., *Promoting team effectiveness*. In West, M. A., *Handbook of workgroup psychology*. Wiley & Sons: Chichester 1996
- [TCO web] <http://h18000.www1.hp.com/tco/models.html>
- [Tuchowska 02] Tuchowska, M., *Jak zbudować zespół wirtualny? Przepis na sukces*. VI Konferencja Project Management - To Ludzie robią projekty, 16-17 Października 2002, Gdańsk – SPMP s.127-131
- [Tversky 02] Tversky, B., *What do Sketches say about Thinking?*, AAAI Spring Symposium: Sketch Understanding, March 25-27, 2002 Stanford University
- [Tyszka 99] Tyszka, T., *Psychologiczne pułapki oceniania i podejmowania decyzji*. roz. 1-9. Gdańskie Wydawnictwo Psychologiczne, Gdańsk 1999
- [UCM web] Use Case Modelling Capturing User Requirements,
http://www.zoo.co.uk/~z0001039/PracGuides/pg_use_cases.htm
- [Ueda 96] Ueda, K., Ohkura, K., *A Biological Approach to Complexity in Manufacturing Systems*, Manufacturing Systems vol25, 1996, No 1, s.57-65
- [UML 02] Boosch, G., Rumbaugh, J., Jacobson, I., *UML przewodnik użytkownika, inżynieria oprogramowania*, WNT Warszawa 2002
- [UML 1.4] OMG Unified Modeling Language, Specification, wersja 1.4, wrzesień 2001
<http://www.omg.org>
- [UML 99] OMG, *OMG Unified Modeling Language Specification*, version 1.3, 1999
- [UML] Fowler, M., Scott, K., *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Addison-Wesley Longman, 2000

- [VISIO] VISIO 2000, VISIO 2002 www.microsoft.com/viso/
- [VisualMind] Visual Mind <http://www.visual-mind.com/>
- [Vliet 99] Van Vliet, H., *Software Engineering: Principles and Practice*, Second Edition, Willey 1999
- [Welenc 02] Welenc, P., *Psychologia stosunków międzyludzkich w projekcie informatycznym*. VI Konferencja Project Management - To Ludzie robią projekty, 16-17 Października 2002, Gdańsk – SPMP s.141-150.
- [Wiem 01] Wielka Internetowa Encyklopedia Multimedialna 2001, <http://wiem.onet.pl>; 2002
- [Wieśniak 02] Analiza FMEA procesujako sposób myślenia i postępowania na linii montażu bloku silnika w celu doskonalenia jakości, Praca dyplomowa, Studium podyplomowe zarządzania jakością, WCTT PWr 2002
- [Winn, Sutherland 89] Winn, W. D. & Sutherland, S. W., *Factors influencing the recall of elements in maps and diagrams and the strategies used to encode them*. Journal of Educational Psychology, 81(1), 33-39. 1989
- [WM 03] Wissensmanagement Conference, Materiały konferencyjne 2-4.04.2003, Lucerna, Szwajcaria
- [Wojciechowski] *Wojciechowski, Knige für deutsche Unternehmer in Polen, Industrie- und Handelskammer Frankfurt (Oder)*
- [WOW 03] Workshop Ontologie-basiertes Wissensmanagement [WM 03]