

Jerzy Wróbel (redaktor)
Grzegorz Linkiewicz, Stanisław Skotnicki

Techniki komputerowe w bezpieczeństwie pracy

Warszawa 2010



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Politechnika Warszawska
Wydział Samochodów i Maszyn Roboczych
Studia Podyplomowe dla Nauczycieli Przedmiotów Zawodowych
02-524 Warszawa, ul. Narbutta 84, tel 22 849 43 07, 22 234 83 48
ipbmv.simr.pw.edu.pl/spin/, e-mail: sto@simr.pw.edu.pl

Opiniodawca: mgr inż. Jacek JUSIS

Projekt okładki: Norbert SKUMIAŁ, Stefan TOMASZEK

Projekt układu graficznego tekstu: Grzegorz LINKIEWICZ

Skład tekstu: Janusz BONAROWSKI

Publikacja bezpłatna, przeznaczona dla słuchaczy Studiów Podyplomowych dla Nauczycieli Przedmiotów Zawodowych.

Copyright © 2010 Politechnika Warszawska

Utwór w całości ani we fragmentach nie może być powielany ani rozpowszechniany za pomocą urządzeń elektronicznych, mechanicznych, kopiujących, nagrywających i innych bez pisemnej zgody posiadacza praw autorskich.

ISBN 83-89703-29-7

Druk i oprawa: Drukarnia Expol P. Rybiński, J. Dąbek Spółka Jawna,
87-800 Włocławek, ul. Brzeska 4

Spis treści

Przedmowa	5
1. Wprowadzenie do komputerowego wspomagania prac związanych z bezpieczeństwem człowieka w środowisku pracy i ergonomią	7
1.1. Projektowanie wspomagane komputerowo	8
1.2. Przedmiot informatyki	15
2. Bazy danych	25
3. Symulacja sytuacji niebezpiecznych w użytkowaniu maszyn	47
3.1. Wprowadzenie do symulacji	48
3.2. Technologia wirtualnej rzeczywistości	49
3.3. Przykłady wykorzystania wirtualnej rzeczywistości do symulacji sytuacji niebezpiecznych	51
4. Zakończenie	63
5. Bibliografia	67

Przedmowa

Techniki komputerowe w niemal rewolucyjny sposób zmieniają funkcjonowanie wszystkich dziedzin gospodarki i życia społecznego. Znaczący jest także udział tych narzędzi w podnoszeniu na wyższy poziom bezpieczeństwa człowieka w środowisku pracy a także ergonomii.

Niniejsze materiały zostały opracowane w ramach realizacji Programu Rozwojowego Politechniki Warszawskiej finansowanego ze środków PROGRAMU OPERACYJNEGO KAPITAŁ LUDZKI. Przeznaczone są dla słuchaczy Studiów Podyplomowych dla Nauczycieli Przedmiotów Zawodowych „Bezpieczeństwo człowieka w środowisku pracy i ergonomia” na Wydziale Samochodów i Maszyn Roboczych Politechniki Warszawskiej. Swoim zakresem obejmują zagadnienia określone w programie studium dla przedmiotu pt. „Informatyka i systemy informatyczne” opisanym w sylabusie opracowanym dla tego przedmiotu.

Podręcznik „Informatyka i systemy informatyczne” przedstawia wybrane możliwości zastosowania komputerów w tej tak ważnej problematyce.

W drugim wydaniu tego podręcznika zaproponowano, w stosunku do wydania z 2009 roku, nieznaczne zmiany wynikające z rocznych doświadczeń w prowadzeniu tego przedmiotu. W szczególności rozszerzono rozdział trzeci dotyczący symulacji sytuacji niebezpiecznych w użytkowaniu maszyn. Zaktualizowano także literaturę przedmiotu.

Podręcznik składa się z trzech części. W pierwszej zatytułowanej „Wprowadzenie do komputerowego wspomaganie prac związanych z bezpieczeństwem człowieka w środowisku pracy i ergonomią” przedstawiono na przykładzie procesu projektowania maszyn i stanowisk pracy jakie są podstawowe idee projektowania wspomaganego komputerowo. W procesie projektowania kształtowane są bowiem podstawowe charakterystyki bezpieczeństwa. W rozdziale tym przedstawiono też wybrane zagadnienia z teorii informacji.

Rozdział drugi poświęcony jest wprowadzeniu do komputerowych bazy danych. Technologie bazodanowe mogą być z powodzeniem stosowane w zagadnieniach bezpieczeństwa i ochrony człowieka w środowisku pracy.

Rozdział trzeci dotyczy symulacji sytuacji niebezpiecznych w użytkowaniu maszyn. Po krótkim wprowadzeniu do symulacji opisano podstawowo-

we możliwości technologii wirtualnej rzeczywistości W rozdziale tym opisano też przykłady wykorzystania wirtualnej rzeczywistości do symulacji sytuacji niebezpiecznych. Pierwszy przykład dotyczy użytkowania wózków widłowych w transporcie wewnątrzzakładowym. Zaproponowano prosty model wózka, w środowisku VRML, do symulacji jazdy po łuku obciążonego wózka widłowego. Przedstawiono i przeanalizowano przykładowe wyniki symulacji. Drugi przykład dotyczy wykorzystania wirtualnej rzeczywistości do optymalizacji linii montażowej przekładni zębatej.

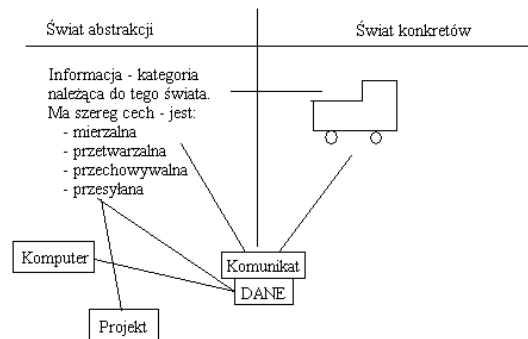
Symulacja komputerowa podobnie jak technologie bazodanowe może być z powodzeniem stosowane w wielu zagadnieniach związanych z bezpieczeństwem człowieka w środowisku pracy i ergonomią.

1

**Wprowadzenie do
komputerowego
wspomagania prac
związanych
z bezpieczeństwem
człowieka
w środowisku pracy
i ergonomią**

1.1. Projektowanie wspomagane komputerowo

Informacja jest pojęciem abstrakcyjnym, bezpośrednio związanym z otaczającą nas rzeczywistością. W praktyce na ogół następuje proces *kodowania informacji*. Zakodowana informacja zwana jest *komunikatem*. Podstawowym narzędziem związanym z kodowaniem i dekodowaniem informacji jest **komputer**, którego podstawową funkcją jest analizowanie przetwarzanych informacji zapisanych w postaci *danych*.



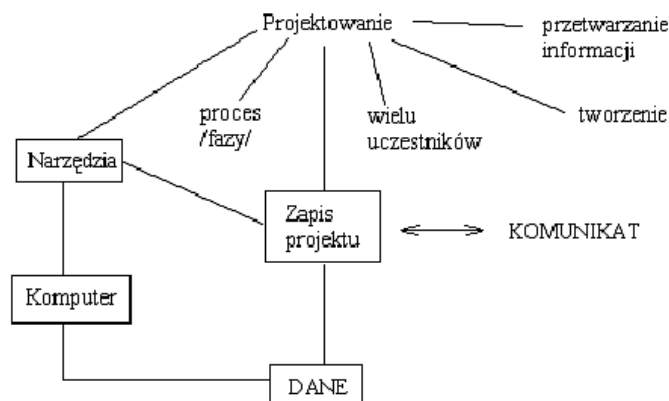
Rysunek 1.1. Projektowanie a przetwarzanie informacji

Szczególnym procesem przetwarzania informacji jest projektowanie.

Projekt – element świata abstrakcji - jest bardzo specyficznym rodzajem informacji stanowiącym obraz tego wszystkiego co ma być wykonane względnie przetworzone, zaś *proces projektowania* sprowadza się do przetwarzania wszelkich informacji o obiekcie, który ma być wykonany (wytworzony). Definiując projektowanie należy zatem pamiętać, że:

1. **projektowanie jest procesem**, często długotrwałym, składającym się z wielu faz;
2. w procesie projektowania bierze udział wielu uczestników;
3. projektowanie jest – z definicji – **procesem twórczym** nie dającym się zalgorytmizować;

4. istota projektowania wymaga *zapisu projektu*, który jest dla nas – z punktu widzenia przetwarzania informacji – *komunikatem*;
5. proces projektowania wymaga *wyspecjalizowanych narzędzi*, bez których zapis informacji byłby nietrwały.



Rysunek 1.2. Cechy projektowania

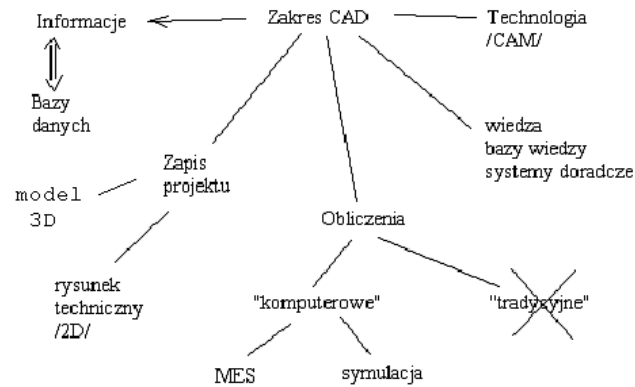
Projektowanie jest więc przetwarzaniem informacji, którego efektem powinien być zapis projektu. W procesie projektowania szerokie zastosowanie znajdują więc komputery.

CAD – (ang. *Computer Aided Design*) *Projektowanie Wspomagane Komputerowo* jest to taki **proces** projektowania, w którym komputer (wraz z oprogramowaniem) wykorzystywany jest na **każdym** etapie (wspomaga każdą czynność związaną z projektowaniem).

Zakres Wspomagania Prac Inżynierskich

Historycznie rzecz ujmując, projektowanie prawie zawsze obejmowało zapis projektu w różnych fazach projektowania. Do zapisu projektu (także w czasach przedkomputerowych) korzystamy z zasad rysunku technicznego (rzutowanie prostokątne) i opisów słownych (np. związanych z obliczeniami inżynierskimi).

Pojawienie się komputera w pierwszej kolejności uprościło obliczenia (początkowo komputer wykorzystano do klasycznego przetwarzania numerycznego). Oczywiście chodziło tutaj o obliczenia „tradycyjne” realizowane do tej pory na suwakach logarytmicznych, czy kalkulatorach.



Rysunek 1.3. Zakres CAD

Rozwój techniki komputerowej spowodował możliwość zaprzęgnięcia komputerów do wykonywania obliczeń bardziej skomplikowanych charakteryzujących się większą ilością danych, wymagających większego przetwarzania danych, obliczeń których wykonywanie dotychczas było znacznie ograniczone wyjątkowo dużą praco- i czasochłonnością. Taki sposób wykorzystania komputerów zaowocował wynikami znacznie bardziej niż dotychczas zbliżonymi do dokładnych. Bardzo intensywny rozwój „komputerowych” metod obliczeniowych (obliczenia charakteryzujące się bardzo dużą ilością danych, trudne lub wręcz niemożliwe do wykonania metodami tradycyjnymi) sprawił, że dziś dostępnych jest wiele programów rozwiązujących problemy np. Metodą Elementów Skończonych, symulujących pracę obiektu (badanie nie tyle samych maszyn, ale ich zachowań w czasie, niejednokrotnie z udziałem człowieka – symulatory). Ostatnio coraz większą popularnością cieszą się komputerowe metody optymalizacyjne.

Idea procesu projektowania praktycznie sprzęga obliczenia z zapisem projektu, obliczenia są prowadzone w tzw. środowisku projektowym. W chwili obecnej programy do obliczeń komputerowych mogą bezpośrednio współpracować z zapisem projektu, pojawia się integracja narzędzi projektowych.

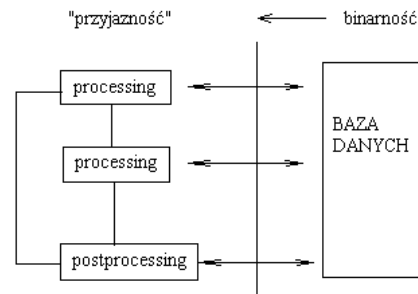
W miarę upływu czasu szybko okazało się, że dużo większe zyski pojawiają się w przypadku zastosowania komputerów nie tyle do obliczeń, ale całego procesu projektowania. Pierwsze opracowane z tą myślą narzędzia symulowały deskę kreślarską (edytory rysunku 2D – edytory dwuwymiarowe). Z czasem pojawiła się możliwość przejścia od dość uciążliwego modelowania dwuwymiarowego 2D do modelowania 3D.

**WPROWADZENIE DO KOMPUTEROWEGO WSPOMAGANIA PRAC ZWIĄZANYCH Z
BEZPIECZEŃSTWEM CZŁOWIEKA W ŚRODOWISKU PRACY I ERGONOMIA**

Rozpoczęto budowanie modeli bezpośrednio w przestrzeni 3D ze szczególnym uwzględnieniem reprezentacji bryłowej obiektów.

Należy pamiętać, że sam zapis projektu jest w formie binarnej i znajduje się w bazie danych, zaś to co ogląda użytkownik ma mieć – z założenia – formę możliwie „przyjazną”. Wyróżnia się trzy podstawowe etapy komunikacji użytkownika z bazą danych:

- **preprocessing** – proces pobierania informacji z bazy danych, oraz wprowadzania do niej wszelkiego rodzaju danych wejściowych;
- **processing** – przetwarzanie informacji niezbędne do zrealizowania jakiegoś celu (np. obliczenia);
- **postprocessing** – wizualizacja otrzymanych wyników processing-u (to także forma pobierania informacji z bazy danych i przyjaznego ich oglądania).



Rysunek 1.4. Bazy danych w projektowaniu

Kolejnym sposobem przyspieszenia i usprawnienia procesu projektowania stała się koncepcja wymiany tej samej dokumentacji pomiędzy różnymi uczestnikami procesu. Koncepcja ta zakłada **integrację obiegu informacji w przedsiębiorstwie**. Podstawą tej koncepcji jest zapis informacji o obiektach, zadaniach i stopniach zaawansowania poszczególnych prac w jednej, spójnej **inżynierskiej bazie danych**. Podstawową rolę pełni więc **Inżynierska Baza Danych**, sam system zarządzania taką bazą zaś nosi często nazwę **EDM** (*ang. Engineering Data Management*).

Koncepcja integracji obiegu informacji w przedsiębiorstwie często określana jest jako Zintegrowane Komputerowo Wytwarzanie (**CIM** – *Computer Integrated Manufacturing*) co zilustrowano na rysunku 1.5.

CAM to komputerowo wspomagane wytwarzanie. W systemach tych informacja z systemu **CAD** binarnie przesyłana jest do systemu **CAM**.

W chwili obecnej dąży się do *zintegrowania* wszystkich narzędzi związanych z projektowaniem i wytwarzaniem wyrobu w jeden sprawnie działający system (CIM). Dąży się też do integracji narzędzi związanych z całym cyklem „życia” wyrobu.

W procesie projektowania należy uwzględnić cały czas „życia” maszyny tj. projektowanie, wytwarzanie, eksploatację a także etapy związane z jej likwidacją (w tym recykling).



Rysunek 1.5. Koncepcja CIM

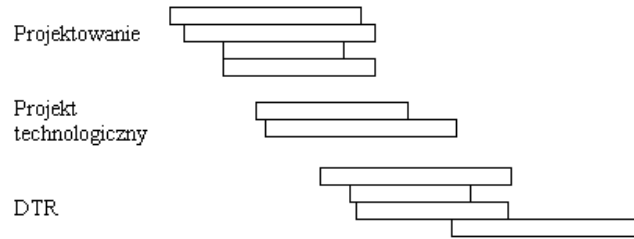
Wszystkie informacje dotyczące któregoś z etapów czasu „życia” maszyny zebrane są w **inżynierskiej bazie danych**. Z zasobu informacji zgromadzonych w inżynierskiej bazie danych korzystamy przez cały czas „życia” maszyny.

Informacje przechowywane w inżynierskiej bazie danych tworzą przede wszystkim projektanci (CAD). Równolegle z nimi mogą zacząć pracę technolodzy, specjaliści od przygotowania produkcji, ale także eksperci od marketingu, reklamy, pionu finansowego (koszty) itd.

Z inżynierskiej bazy danych korzystają różni ludzie, zaś z jej pewnych zasobów mogą korzystać nie tylko ludzie, ale także np. inne systemy komputerowe. W wielu wytwórniach powszechne w zastosowaniu są obecnie niektóre aspekty tej idei jak np.: obrabiarki sterowane numerycznie (CNC) czy roboty przemysłowe.

Koncepcja **CIM** ułatwia też tzw. **projektowanie współbieżne** zwane również **inżynierią współbieżną** (od angielskiego określenia: *concurrent*

engineering), umożliwiające **współbieżne realizowanie pewnych faz projektowania** (rysunku 1.6).



Rysunek 1.6. Projektowanie współbieżne

Wymagania w procesie projektowania

Tym, co odróżnia pracę projektanta maszyn od twórczej działalności np. artysty jest to, że projektant jest skrepowany różnymi **wymaganiami i ograniczeniami**. Podstawowymi wymaganiami, z jakimi spotyka się projektant są:

- **wymagania funkcjonalne** – klient lub inny zleceniodawca formułuje podstawowe wymagania – jakie funkcje ma realizować maszyna, jaki ma być jej koszt, sprawność, zużycie energii itp.;
- **wymagania wytrzymałościowe** – decydują one o niezawodności i poprawności działania; do spełnienia tych wymagań mamy wiele specjalizowanych narzędzi, jak chociażby MES;
- **wymagania ekonomiczne** – wymagania te dotyczą zarówno kosztów jak i czasu:
 - projektowania maszyny
 - wytwarzania maszyny
 - eksploatacji maszyny
 - „kasowania” maszyny
- **wymagania ergonomiczne i związane z ochroną środowiska** – wymagania te to coraz bardziej istotna grupa

wymagań; wiele z nich jest sformalizowanych (szczególnie wymagania dotyczące bezpieczeństwa);

- **wymagania estetyczne;**
- **wymagania technologiczne;**
- **wymagania kulturowe** – związane głównie z projektowaniem późniejszego procesu wytwarzania i eksploatacji (należy uwzględnić wpływ kultury technicznej przyszłych użytkowników na stopień skomplikowania maszyny);
- **wymagania prawne** – w większości krajów, aby sprzedać maszynę należy spełnić wiele przepisów prawnych związanych z jej dopuszczeniem do użytkowania (certyfikaty, homologacje itp.); proces ten jest bardzo uciążliwy i czasochłonny, dlatego należy go uwzględnić już na etapie projektowania;
- **wymagania związane normalizacją** – wymagania te mają charakter międzynarodowy i pomimo, że nie są obligatoryjne, to na ogół są przestrzegane w procesie projektowania.

Należy podkreślić, że wymagania są coraz bardziej sformalizowane. Źródłem wymagań są:

- zamawiający (projektodawca)
- przepisy i inne akty prawne
- normy
- doświadczenie firmy (know how)
- badania – przyszłych użytkowników, a także badania własne – związane z eksploatacją, z pracownikami itp.).

Dotychczas wszelkiego rodzaju wymagania dostępne były jedynie w postaci zapisów na papierze. Obecnie w coraz większym stopniu korzystamy z technologii **inżynierskich baz danych** oraz ze stosownych aktów prawnych dostępnych przez internet.

1.2. Przedmiot informatyki

Zarys historii maszyn liczących

Panuje przekonanie, że współczesny etap rozwoju gospodarki i społeczeństwa to etap początkowy ery cywilizacji informacyjnej (niektórzy mówią wręcz o rewolucji informatycznej), a więc takiej cywilizacji, która traktując samą informację jako specyficzne dobro niematerialne wysuwa ją na pozycję dominującą, dając jej pierwszeństwo nad dobrami materialnymi. Głównym czynnikiem rozwoju cywilizacji informacyjnej są komputery.

Komputer jest programowalnym urządzeniem (elektronicznym) przeznaczonym do przechowywania, przesyłania i przetwarzania informacji.

Historia rozwoju maszyn liczących jest długa - za jedną z pierwszych prymitywnych form maszyn liczących można uważać liczydła, stosowane już w czasach starożytnych. Mechaniczne urządzenia do liczenia pojawiły się jednak dopiero w XVII wieku, a Leibniza (Leibniz Gottfried Wilhelm (1646-1716), niemiecki filozof i matematyk,) uważa się za twórcę pierwszego arytmetometru, tj. maszyny wykonującej cztery podstawowe działania arytmetyczne. Za początek ery komputerów zgodnie uznaje się rok 1946, w którym zbudowano pierwszą elektroniczną maszynę cyfrową ENIAC. Electronic Numerical Interpreter And Calculator, pierwszy na świecie komputer, skonstruowany przez J.P. Eckerta i J.W. Mauchly'ego na Uniwersytecie Pensylwanii w latach 1943-1946, dla potrzeb obliczeń balistycznych Marynarki Wojennej USA posiadał 17468 lamp elektronowych, pobór mocy wynosił 130 kW). Masowe przetwarzanie informacji w ośrodkach komputerowych stało się jednak możliwe dopiero od 1964 roku, kiedy to firma IBM wyprodukowała komputer o nazwie IBM 360. Prawdziwą rewolucję spowodowało pojawienie się na rynku komputerowym mikrokomputerów klasy IBM PC (komputer osobisty — personal computer) w sierpniu 1981 roku. Odtąd obserwujemy gwałtowny rozwój zastosowań zarówno mikrokomputerów, jak i komputerów o większej mocy obliczeniowej połączonych w sprawnie działające sieci komputerowe.

Rynek komputerowy jest zaś jedną z najbardziej dynamicznie rozwijających się dziedzin gospodarki.

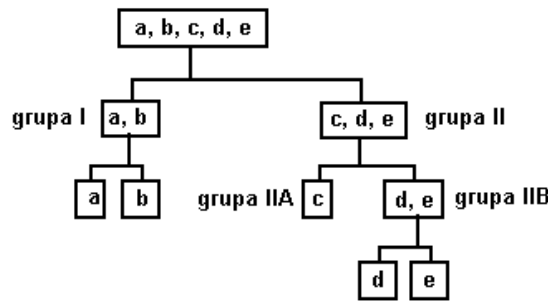
Elementy teorii informacji

Badaniem problemów ilości informacji, sposobów jej kodowania, przetwarzania i przesyłania zajmuje się *teoria informacji*. Twórcą tej teorii jest Claude E. Shannon.

Informacja jest mierzalną wielkością abstrakcyjną, która może być przechowywana, przesyłana i przetwarzana w pewnych obiektach, a także stosowana do sterowania obiektami, przy czym przez obiekty rozumie się organizmy żywe, urządzenia techniczne oraz systemy takich obiektów [21].

Do przesyłania informacji służą komunikaty. Komunikat zawiera tym więcej informacji, im mniejsze jest prawdopodobieństwo jego wystąpienia.

Rozważmy źródło komunikatów nadające 5 różnych komunikatów oznaczonych przez a, b, c, d, e (rysunek 1.7), przy czym nadanie komunikatu z I grupy (a, b) jest tak samo prawdopodobne, jak nadanie komunikatu z II grupy (c, d, e). Prawdopodobieństwo wystąpienia komunikatu a lub b w I grupie jest także jednakowe, podobnie jak prawdopodobieństwo nadania komunikatu z grupy IIA (c) i grupy IIB (d, e). Prawdopodobieństwo wystąpienia komunikatów d i e jest również jednakowe.



Rysunek 1.7. Struktura komunikatów

Oznaczając przez $P\{x \vee y \vee z\}$ prawdopodobieństwo zdarzenia, że wystąpi komunikat x lub y lub z , zauważamy natychmiast, że

$$P(a \vee b) = P(c \vee d \vee e) \tag{1.1}$$

$$P(a) = P(b) = P(c) = P(d \vee e) \tag{1.2}$$

$$P(d) = P(e) \tag{1.3}$$

Jeśli został nadany komunikat, to nastąpiło zdarzenie ($a \vee b$) lub ($c \vee d \vee e$), więc

$$P(a \vee b) + P(c \vee d \vee e) = 1 \quad (1.4)$$

a uwzględniając (1.1) otrzymamy

$$P(a \vee b) = P(c \vee d \vee e) = \frac{1}{2} \quad (1.5)$$

Jeśli zaistniało zdarzenie ($a \vee b$), to musiało nastąpić nadanie komunikatu a lub b . Korzystając ze znanego wzoru na prawdopodobieństwo warunkowe i uwzględniając (1.2) otrzymujemy

$$P(a) = P(b) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} = \left(\frac{1}{2}\right)^2 \quad (1.6)$$

analogicznie

$$P(c) = P(d \vee e) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} = \left(\frac{1}{2}\right)^2 \quad (1.7)$$

$$P(d) = P(e) = \frac{1}{8} = \left(\frac{1}{2}\right)^3 \quad (1.8)$$

Wzór na ilość informacji ma postać:

$$K = \log_2 \left(\frac{1}{p} \right) \quad (1.9)$$

Komunikat, którego prawdopodobieństwo wystąpienia wynosi p , zawiera K jednostek informacji.

Jednostką informacji jest *bit* (oznaczenie — b), tzn. taka jej ilość, jaką uzyskujemy po stwierdzeniu, że zaszło jedno z dwu jednakowo prawdopodobnych zdarzeń.

Komunikaty a , b , c zawierają po dwie takie jednostki, natomiast komunikaty d , e zawierają po trzy jednostki.

Jeśli rozpatrywane źródło może nadawać tylko jeden komunikat, którego prawdopodobieństwo wynosi jeden, to niesie on $\log_2(1/1) = 0$ bitów informacji.

Kodowanie informacji

Każdy komunikat można zakodować w różny sposób, zaś odbiorca komunikatu znając sposób kodowania może zrekonstruować (zdekodować) początkową postać komunikatu. Powszechnie przyjmuje się do kodowania alfabet składający się jedynie z dwóch różnych znaków, oznaczanych przez zero i jeden. Sposób ten jest praktyczny, gdyż łatwo daje się zrealizować (np. poprzez przesyłanie komunikatów za pomocą impulsów elektrycznych).

Kodowanie informacji, w którym używa się alfabetu dwuznakowego, zwanego też dwójkowym lub binarnym, jest właśnie stosowane w komputerach. Do przechowywania informacji w kodzie binarnym służy pamięć. W przeszłości funkcję nośnika informacji pełniła też papierowa karta lub taśma dziurkowana. Należy jednak podkreślić, że istnieją różne sposoby kodowania informacji dające podzielić się na dwie zasadnicze grupy: kodowanie dyskretne (zwane też znakowym) i kodowanie ciągłe.

Przy kodowaniu dyskretnym zapis dowolnej informacji powstaje z elementów pewnego, z góry ustalonego zbioru symboli (znaków). Przykłady kodowania dyskretnego to pismo alfabetyczne, notacja nutowa, dziesiętny zapis liczb, ale także opatrzone obrazkami (piktogramami) klawisze na różnych urządzeniach. Kodowaniem dyskretnym jest też kodowanie binarne. Jeśli znakami alfabetu binarnego są zero i jeden, to mówimy także o kodowaniu cyfrowym, a komputery przetwarzające informacje zakodowane w ten sposób nazywamy komputerami cyfrowymi.

Przy kodowaniu ciągłym zbiór stosowanych znaków jest nieskończenie liczny. Przykładem kodowania ciągłego jest mowa, muzyka, obraz malarski. Również zapis dźwięku na płycie gramofonowej czy na taśmie magnetofonowej jest uważany za kodowanie ciągłe. Natomiast zapis tego samego dźwięku na płycie kompaktowej jest binarny.

Maszyny matematyczne przetwarzające informacje kodowane ciągle nazywamy komputerami analogowymi (wykorzystują one pośrednio lub bezpośrednio analogie zjawisk). Maszyna hybrydowa zawiera część analogową i część cyfrową połączone ze sobą urządzeniami pozwalającymi na wymianę informacji. Komputery analogowe w prze-

szłości znajdowały zastosowanie do symulacji różnego rodzaju procesów ciągłych. Wzrost możliwości komputerów cyfrowych spowodował zanik stosowania tych maszyn. W dalszej części skryptu mówiąc komputer będziemy rozumieć przez ten termin komputer cyfrowy.

Komputer przetwarza informacje zapisane w kodzie binarnym. Taka postać jest bardzo niewygodna dla użytkowników, przyzwyczajonych do znakowego kodowania w tradycyjnie stosowanych alfabetach (np. alfabet polski czy łaciński). Należy więc posłużyć się urządzeniem przetwarzającym kody. Istota takiego przetwarzania jest bardzo prosta. Posługując się tablicą, w której w jednej kolumnie zgromadzono wszystkie symbole alfabetu komunikatu zewnętrznego, a w drugiej odpowiadające im ciągi znaków binarnych, zamieniamy poszczególne symbole komunikatu zewnętrznego na żadaną postać wewnętrzną lub odwrotnie — ciąg znaków binarnych może być zastąpiony odpowiednimi symbolami komunikatu zewnętrznego.

Przykładem takiej tablicy jest kod ASCII (American Standard Code for Information Interchange), który jest zgodny z międzynarodowym kodem ISO-7 (International Organization for Standardization). Kod ten jest stosowany do przetwarzania komunikatów zapisanych w alfabecie łacińskim, wzbogaconym o pewne symbole dodatkowe. Nazywamy go kodem alfanumerycznym. W kodzie tym każdemu symbolowi alfanumerycznemu odpowiada dwójkowy ciąg kodowy o długości 8, zwany bajtem (ang. byte).

Przykładowo:

A 01000001
a 01100001
l 00110001

Uwaga – w trzecim przykładzie *l* oznacza znak ASCII (cyfrę) a nie liczbę całkowitą!!! Kodowaniu liczb w komputerze poświęcimy kolejny punkt tego rozdziału.

Niektóre ciągi znakowe przeznacza się na symbole dodatkowe (np. przejście do nowego wiersza). Przykładem urządzenia przekształcającego symbole alfanumeryczne na ciągi impulsów elektrycznych o dwu wyraźnie różnych napięciach odpowiadających zeru i jedynce jest klawiatura komputera. Naciśnięcie klawisza opatrzonego np. znakiem litery A powoduje wysłanie ciągu impulsów odpowiadającego tej literze.

ROZDZIAŁ 1

Jak wspomniano jeden bajt zawiera osiem bitów

$$1 \text{ B} = 8 \text{ b}$$

a więc 1B pozwala reprezentować $2^8 = 256$ różnych symboli.

Większą jednostką informacji jest kilobajt

$$1 \text{ KB} = 2^{10} \text{ B} = 1024 \text{ B}$$

zawierający 1024 bajty, a dalej megabajt

$$1 \text{ MB} = 2^{30} \text{ B} = 2^{10} \text{ KB}$$

zawierający 1024 kilobajty, a dalej gigabajt

$$1 \text{ GB} = 2^{40} \text{ B} = 2^{10} \text{ MB}$$

zawierający 1024 megabajty, a dalej terabajt

$$1 \text{ TB} = 2^{50} \text{ B} = 2^{10} \text{ GB}$$

Kodowanie liczb

Poprzedni punkt poświęcony był kodowaniu informacji, w tym także kodowaniu liter i cyfr. Nie pisaliśmy jednak dotąd o kodowaniu liczb. Jednym z najbardziej popularnych sposobów kodowania liczb jest dziesiętny system pozycyjny. Liczby całkowite zapisuje się w tym systemie za pomocą 10 cyfr i ewentualnie znaku „-” (minus).

Liczby całkowite zapisuje się w dziesiętnym systemie pozycyjnym w następujący sposób:

$$c_n c_{n-1} \dots c_0 \quad (1.10)$$

w którym $c_n c_{n-1} \dots c_0$ są *cyframi* (0,1,...9) – co oznaczają jak wiadomo liczbę

$$c_n 10^n + c_{n-1} 10^{n-1} + \dots + c_0 10^0 \quad (1.11)$$

przykładowo „582” czytamy „pięćset osiemdziesiąt dwa”, zgodnie z definicją dziesiętnego systemu pozycyjnego reprezentuje wartość:

$$5 \cdot 10^2 + 8 \cdot 10^1 + 2 \cdot 10^0 = 582.$$

Zauważmy, że rzymski zapis liczb nie jest zapisem pozycyjnym. Przykładowo rzymski zapis „C” – oznacza liczbę „100”, a „CC” – oznacza liczbę „200”.

Do zapisu liczb wymiernych stosuje się znak kropki lub przecinka oddzielający część całkowitą od części ułamkowej.

Stwierdziliśmy już, że do zapisu dowolnych informacji w komputerze korzystnie jest stosować kod dwójkowy. Z tego względu również liczby przedstawiamy w kodzie dwójkowym. Najczęściej stosuje się do tego celu *dwójkowy system pozycyjny*. Innym sposobem jest *dwójkowe kodowanie cyfr dziesiętnych* (tzw. BCD - ang. Binary Code Decimals).

Liczby całkowite zapisuje się w dwójkowym systemie pozycyjnym w następujący sposób: Zapis

$$c_n c_{n-1} \dots c_0 \quad (1.12)$$

w którym $c_n c_{n-1} \dots c_0$ są cyframi dwójkowymi (cyfra dwójkowa (0,1) jest również zwana bitem), oznacza w dziesiętnym systemie pozycyjnym liczbę

$$c_n 2^n + c_{n-1} 2^{n-1} + \dots + c_0 2^0 \quad (1.13)$$

Tak więc zapis 10011 oznacza liczbę $2^4 + 2^1 + 2^0 = 19$.

W podobny sposób przedstawiamy w postaci rozwinięcia dwójkowego dodatnie ułamki właściwe. W dwójkowym systemie pozycyjnym zapis

$$c_{-1} c_{-2} \dots c_{-n} \quad (1.14)$$

w którym $c_{-1} c_{-2} \dots c_{-n}$ są cyframi dwójkowymi, oznacza w dziesiętnym systemie pozycyjnym liczbę

$$c_{-1} 2^{-1} + c_{-2} 2^{-2} + \dots + c_{-n} 2^{-n} \quad (1.15)$$

Tak więc zapis 1011, traktowany jako zapis ułamka właściwego, oznacza liczbę

$$2^{-1} + 2^{-3} + 2^{-4} = 11/16 = 0,6875.$$

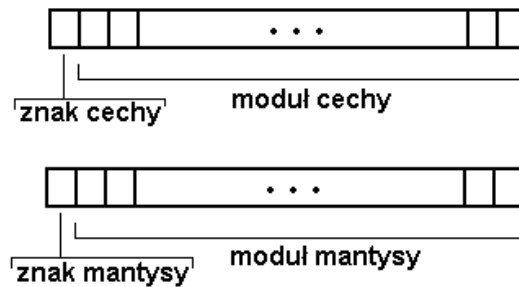
Liczby ułamkowe przedstawia się w postaci składającej się z części ułamkowej zwanej mantysą (m) i części całkowitej zwanej cechą (c). Wartość liczby określa zależność

$$x = \text{sgn}(x) m 2^c, \text{ gdzie } 0 \leq m < 1, \text{sgn}(x) = \begin{cases} 1 & \text{dla } x \geq 0 \\ 0 & \text{dla } x < 0 \end{cases} \quad (1.16)$$

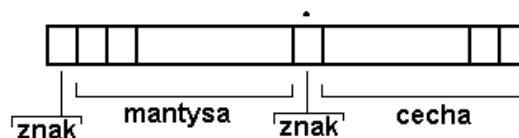
Na rysunku 1.6 przedstawiono schematycznie zapis liczby x , w którym oddzielnie koduje się moduł a oddzielnie cechę będącą liczbą całkowitą.

Taką postać zapisu nazywamy *stałoprzecinkową* (ang. fixed point). Przecinek oddzielający część całkowitą od ułamkowej jest w stałym miejscu (po pierwszej „znakowej” pozycji ciągów kodowych). Liczby w ten sposób zapisane nazywamy liczbami stałoprzecinkowymi.

Drugą formą zapisu jest postać *zmiennoprzecinkowa* (ang. floating point). Liczby kodowane są tak jak w (1.16) z tą różnicą, że w jednym słowie maszynowym. Na rysunku 1.7 przedstawiono schematycznie zapis liczby zmiennoprzecinkowej.



Rysunek 1.6. Zapis liczby stałoprzecinkowej



Rysunek 1.7 Zapis liczby zmiennoprzecinkowej.

System dwójkowy kodowania liczb jest bardzo wygodny z punktu widzenia konstruktorów komputerów, natomiast jest kłopotliwy dla ich użytkowników (kody binarne stają się bardzo długie, np. 16, 32, a nawet 64 bity). Powstał więc problem takiego zapisu liczb, który byłby bardziej zwężły niż dwójkowy, a jednocześnie pozwalałby na natychmiastowe odtworzenie ciągu bitów liczby dwójkowej. W praktyce stosuje się dwa takie sposoby. Są to ósemkowy i szesnastkowy system pozycyjny.

W *ósemkowym systemie pozycyjnym* korzysta się z cyfr: 0, 1, 2, 3, 4, 5, 6, 7. Wyrażenie określające wartość liczby w tym kodzie otrzymujemy natychmiast z (1.12) podstawiając zamiast 2 liczbę 8. Tak więc zapis $127_{(8)}$ przedstawiający liczbę ósemkową ma wartość:

$$1 \cdot 8^2 + 2 \cdot 8^1 + 7 \cdot 8^0 = 87_{(10)}.$$

Indeksy (8) i (10) oznaczają odpowiednio, liczbę ósemkową i dziesiętną. Tego typu indeksy stosujemy tam, gdzie podstawa zapisu nie jest oczywista. Zauważmy, że natychmiastowa jest konwersja zapisu liczby w systemie ósemkowym do zapisu w systemie dwójkowym i odwrotnie, co ilustrują następujące przykłady:



W szesnastkowym systemie pozycyjnym korzysta się z cyfr: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Symbole literowe oznaczają cyfry, których dziesiętne wartości wynoszą odpowiednio A = 10, B = 11, C = 12, D = 13, E = 14, F = 15. Wyrażenie określające wartość liczby w tym kodzie otrzymujemy natychmiast z (1.10) podstawiając zamiast 2 liczbę 16. Tak więc zapis $1E_{(16)}$ przedstawiający liczbę szesnastkową ma wartość: $1 \cdot 16^1 + 14 \cdot 16^0 = 30_{(10)}$. Indeksy (16) i (10) oznaczają odpowiednio liczbę szesnastkową i dziesiętną.

Zauważmy, że natychmiastowa jest konwersja zapisu liczby w systemie szesnastkowym do zapisu w systemie dwójkowym i odwrotnie, co ilustrują następujące przykłady:



Kodowanie liczb w komputerze

Jak podkreśliliśmy, kodowanie wszelkiej informacji w komputerze (znaków tekstowych, liczb, rysunków itp.) odbywa się za pomocą dwóch znaków. Taka postać jest bardzo niewygodna dla użytkowników, przyzwyczajonych do kodowania w tradycyjnie stosowanych alfabetach (np. alfabet polski czy łaciński) lub w innych systemach kodowania np. systemach zapisu liczb. Należy więc przyjąć powszechnie stosowaną umowę – przykładem takiej umowy jest omówiony już kod ASCII do kodowania znaków tekstowych. Umowa taka dotyczy także kodowania liczb.

Liczby – podobnie jak znaki tekstowe – koduje się w bajtach (a więc łańcuchach 8-bitowych). W jednym bajcie można zapisać zarówno jeden znak tekstowy (np.: literę A) jak i liczbę całkowitą z zakresu (-127 ... +127). Procesor komputera powinien więc „rozróżnić” czy informacja zapisana w tym bajcie to znak tekstowy czy też liczba!!! Do tego celu służy w procesie przygotowania programów komputerowych proces zwany deklarowaniem obiektów występujących w danym programie.

W przypadku kodowania liczb programista może stosować liczby całkowite, liczby rzeczywiste i liczby zespolone.

Do kodowania liczb całkowitych korzysta się z 1 lub 2 bajtów (może być także większa liczba), a do deklaracji w programie korzysta się ze specjalnych słów kluczowych (np. *integer*).

Do kodowania liczb rzeczywistych korzysta się z 4 bajtów, a do deklaracji w programie również korzysta się ze specjalnych słów kluczowych (np. *real*).

Do kodowania liczb zespolonych korzysta się z 8 bajtów, a do deklaracji w programie również korzysta się ze specjalnych słów kluczowych (np. *complex*).

Aby zwiększyć dokładność zapisu liczb rzeczywistych korzysta się z 8 bajtów – mówimy o podwójnej precyzji zapisu, a do deklaracji w programie korzysta się ze specjalnych słów kluczowych (np. *double precision*).



Bazy danych

Kanoniczne postacie (formy) przetwarzania informacji

J

Informacja jest wielkością abstrakcyjną, ale mierzalną, może być **przechowywana (kodowana), przesyłana** ale przede wszystkim może być **przetwarzana** przez organizmy żywe lub urządzenia techniczne, w tym przez komputery. Przetwarzanie informacji można podzielić na:

- przetwarzanie numeryczne (numeryka);
- przetwarzanie morfologiczne (morfologia);
- przetwarzanie semantyczne (semantyka).

Kryterium klasyfikacji przedstawionego powyżej podziału jest klasa obiektów, których dotyczy informacja. **Przetwarzanie informacji** wiążemy bowiem zwykle z jakimiś obiektami. Przetwarzanie to jest charakterystyczne dla organizmów żywych, ale w procesie tym można także stosować maszyny a szczególnie komputery

Numeryka

Numeryka to sposób przetwarzania informacji właściwy ludziom. Obiekty są tu bardzo specjalne, bowiem dają się sprowadzić do **liczb** - obiekty są liczbami! Na tych liczbach wykonujemy **działania**. Przetwarzaniem numerycznym zajmuje się matematyka.



Rysunek 2.1. Numeryka

W naturalny sposób pojawiła się też konieczność **zapisu liczb** np. 1, -2, $\frac{1}{2}$, itp. Kiedy tylko zaczynamy przetwarzać informacje związane z tymi obiektami, stosujemy różne **formuły** (dodawanie, odejmowanie, mnożenie, dzielenie, całkowanie, różniczkowanie, itp.). Tak jak w przypadku obiektów (tu liczb), tak i w przypadku formuł musimy dysponować ich **zapisem**, czyli “przepisem” mówiącym jak wykonać przetwarzanie i metodą jak należy postępować. W ten sposób powstają **algorytmy** (algorytm – “przepis” na przetwarzanie numeryczne). Mając algorytmy można tworzyć efektywne narzędzia (np. programy komputerowe) do **przetwarzania numerycznego**.

Morfologia

Przetwarzanie morfologiczne jest takim rodzajem **przetwarzania informacji**, w którym obiekty nie są już liczbami. Obiekty są dowolne (na ogół modelujące konkretny fragment rzeczywistości), ale dające się jednoznacznie opisać. Zdążamy więc do naturalnego przetwarzania informacji. Obiektem może być np. *student*. Mamy świadomość, że nie da się zbudować modelu *studenta* i nie da się go opisać liczbowo, ale dla pewnych celów, na przykład do obsługi pracy dziekanatu, można zbudować wystarczający, a jednocześnie jednoznaczny opis. W analogiczny sposób można opisać krew lub część maszyny:

Student (nr_indeksu, Nazwisko, Imię, ...)

ROZDZIAŁ 2

Krew (OB., RBC, WBC, ...)
Część (...)

Przetwarzanie morfologiczne to takie przetwarzanie, w którym obiekty są dowolne (na ogół modelujące konkretny fragment rzeczywistości), ale dające się jednoznacznie opisać. Obiekty mogą być powiązane ze sobą.

Proces przetwarzania polega przede wszystkim na **umiejętności tworzenia zapisu obiektu**, ale także na jego **modyfikowaniu**, **pozyskiwaniu informacji** (tworzenie zapytań). Ważne też jest specjalne przetwarzanie polegające na **powiązaniach**.

Tak więc proces przetwarzania morfologicznego jest zupełnie inny niż w przypadku numeryki – nie dodaje się numerów indeksów studentów, ale np. wybiera studentów, którzy mają ocenę bardzo dobrą z chemii i jednocześnie ocenę bardzo dobrą z matematyki.

Semantyka

W przetwarzaniu semantycznym obiekty – podobnie jak w przetwarzaniu morfologicznym – są także **dowolne, ale dotyczą pewnej sfery abstrakcji** (w zasadzie bardzo specyficznego rodzaju abstrakcji: wiedzy). Mamy tu do czynienia z bardzo specjalną klasą obiektów, które dotyczą wiedzy dającej się sformalizować.

Wiedza daje się sformalizować, można ją gromadzić i następnie przetwarzać. Należy zaznaczyć, że wiedza może być sformalizowana w różny sposób. Dysponujemy różnymi formami pozyskiwania i zapisu wiedzy (zapisem wiedzy są np. **fakty** czy **reguły**).

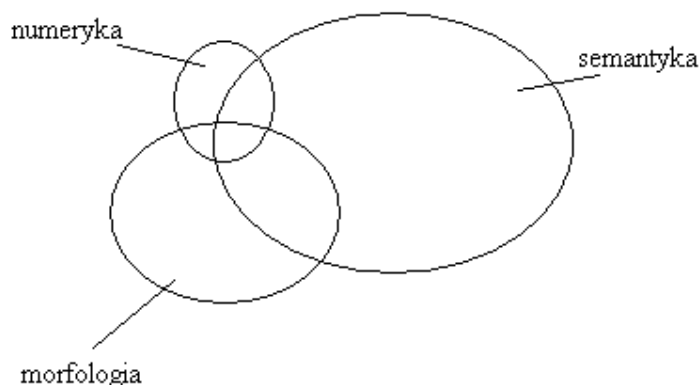
Proces przetwarzania polega w tym przypadku na wnioskowaniu, ale również na zapisie obiektów, ich **modyfikacji** i **edycji**.

W semantyce wyróżnia się zazwyczaj:

- **analizę semantyczną** (rozpoznawanie znaczenia analizowanej porcji informacji);
- **rozpoznawanie obrazów** (rozpoznawanie postaci np. poprzez wzrok, dźwięk, dotyk itp. – jest to powszechna forma przetwarzania semantycznego właściwa dla człowieka i innych organizmów żywych);
- **heurystykę** (w tym przetwarzaniu wnioski wyciągamy na podstawie niepełnych informacji – komputery umożliwiają budowę systemów doradczych / systemów eksperckich,

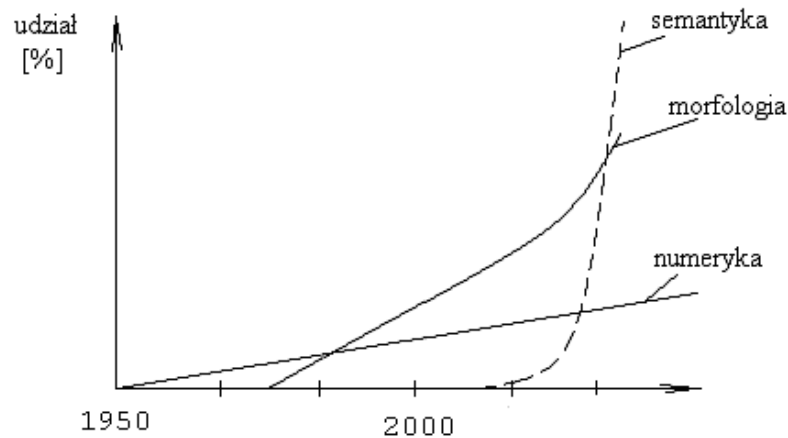
które na podstawie zgromadzonej wiedzy wspomagają proces podejmowania decyzji; wśród systemów doradczych wyróżniamy systemy diagnostyczne – systemy takie „obserwują” symptomy /informacje wejściowe/ diagnozowanych obiektów i na ich podstawie stawiają diagnozę).

Każdy z nas w swoim życiu a szczególnie w działalności zawodowej (na przykład w procesie projektowania) przetwarza informacje. Często częścią tego przetwarzania są obliczenia, a więc numeryka. Ale zazwyczaj występuje przetwarzanie morfologiczne (np. rysowanie, sporządzanie opisów). Na ogół występuje też przetwarzanie wiedzy, czyli semantyka.



Rysunek 2.2. Trzy postacie przetwarzania informacji

Zdając sobie sprawę z występowania trzech opisanych powyżej rodzajów przetwarzania informacji, poszukujemy narzędzi wspomagających proces przetwarzania informacji. Na rysunku 2.3 zilustrowano udział komputerów w przetwarzaniu informacji na przestrzeni ostatnich lat.



Rysunek 2.3. Udział komputerów w przetwarzaniu informacji na przestrzeni ostatnich lat

Przetwarzanie numeryczne – jest to pierwsze stosowanie komputerów - wynikało ono z możliwości pierwszych komputerów i z zapotrzebowania na obliczenia.

Przetwarzanie morfologiczne – dominuje w procesie przetwarzania informacji i jego rozwój z zastosowaniem komputerów jest niezwykle dynamiczny.

Prosty przykład dodawania dwu liczb pokazuje w jaki sposób przebiega proces przetwarzania informacji

$$2 + 2$$

- najpierw jest przetwarzanie **semantyczne** (rozpoznajemy o co chodzi, z czym mamy do czynienia, następnie przypisujemy stosowne znaczenie);
- potem jest **morfologia** (w tym przypadku przeszukujemy naszą „bazę danych” i od razu znajdujemy wynik – w zasadzie nie korzystamy z narzędzi numerycznych).

Definicja bazy danych

Dane to **komunikaty** zapisywane w komputerze (binarnie), które mogą być jednoznacznie odczytane, przesyłane i przetwarzane.

Baza danych to zbiór danych (komunikatów) o obiektach lub procesach:

zapisywany w pamięci zewnętrznej komputera;

zamodelowanych i wzajemnie **powiązanych** (zgodnie ze strukturą logiczną jaką jest *logiczna baza danych*);

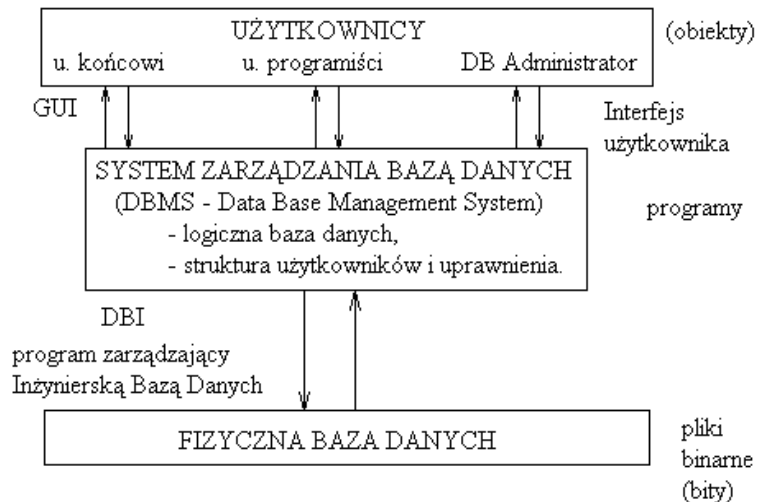
przygotowanych do wielu **różnych zastosowań** (czasami nawet nieznanych), związanych z przetwarzaniem morfologicznym (np.: modyfikacje, zapytania);

udostępnianych różnym użytkownikom o różnych uprawnieniach;

odpowiednio zabezpieczonych (zarówno na etapie przechowywania jak i przetwarzania informacji) w celu realizacji różnych zadań.

Jak widać **Baza Danych** to nie tylko same dane, ale również sposób jak z nich korzystać, jak je udostępniać.

Na rysunku 2.4 przedstawiono system bazy danych.



Rysunek 2.4. System bazy danych.

Fizyczna Baza Danych to pliki zapisane **binarnie**, często rozproszone (na różnych urządzeniach), z natury nieczytelne.

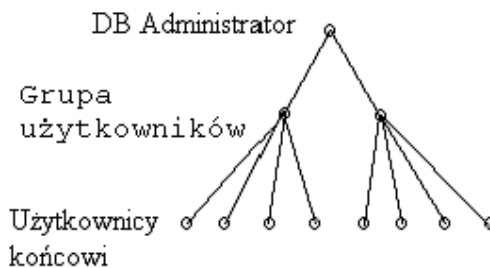
Użytkownicy bazy danych są często niewykwalifikowani, często może to być także nie człowiek, ale urządzenie np. czujnik. Aby więc udostępnić użytkownikom zasoby fizycznej bazy danych potrzebny jest

program (zwykle wiele programów) – zwany **systemem zarządzania bazą danych (DMBS)**.

System zarządzania bazą danych przede wszystkim organizuje komunikację z plikami zewnętrznymi (*DBI – Data Base Interface*), są to programy zarządzające fizyczną bazą danych. System zarządzania bazą danych pracuje oczywiście na podstawie zbudowanego wcześniej modelu, czyli **logicznej bazy danych**.

Misją programów będących systemami zarządzania bazą danych jest tworzenie systemu komunikacji z użytkownikiem (*UI – User Interface*, a w zasadzie *GUI – Graphic User Interface*). System GUI musi być zapewniony (dostarczony) przez system zarządzania bazą danych.

Spośród wielu potencjalnych użytkowników **DBMS**, najważniejszym jest tzw. **administrator**.



Rysunek 2.5. Użytkownicy bazy danych

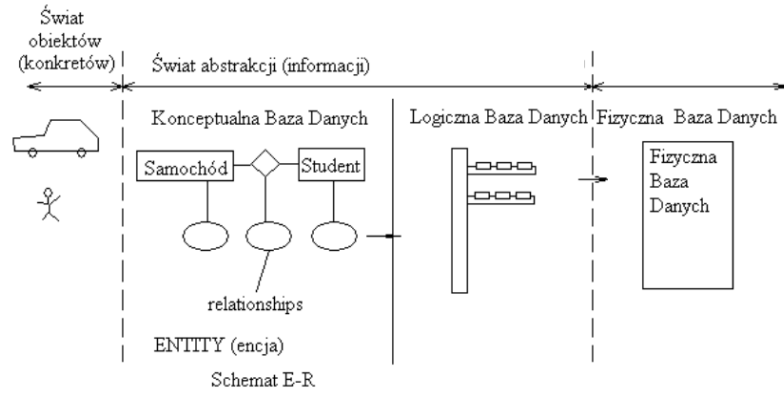
Użytkownicy końcowi to np. projektanci – muszą oni mieć dostarczony naturalny i przyjazny sposób komunikacji z bazą bez konieczności samodzielnego jej programowania.

Programiści – użytkownicy bardziej zaawansowani z punktu widzenia obsługi DBMS. Oni definiują nowe polecenia, ekrany graficzne itp.

Konceptualna, logiczna i fizyczna baza danych

W procesie projektowania baz danych kluczową rolę odgrywają modele. Modelem najbliższym związanym z realnym modelowanym fragmentem świata rzeczywistego jest konceptualna baza danych.

Modelem najbardziej związanym z komputerem jest fizyczna baza danych. Logiczna baza danych jest modelem pośrednim, co przedstawia rysunek 2.6.



Rysunek 2.6. Modele w bazach danych

Konceptualna baza danych jest próbą wydzielenia z otaczającej rzeczywistości pewnych obiektów i powiązania ich ze sobą.

Obiektami tymi mogą być

samochody, studenci itp.

Przykład powiązania:

studenci jeżdżą samochodami.

Zarówno *studenci* jak i *samochody* są tu jedynie modelami, ale za to szczegółowo i jednoznacznie opisanymi. Te obiekty w konceptualnej bazie danych są nazywane **encjami** (ang. *Entity*). Encje są ze sobą powiązane. **Schematy E-R** (zwane też diagramami związków encji) tworzą konceptualną bazę danych.

Na podstawie konceptualnej bazy danych tworzymy **strukturę logiczną** i to już robimy pod kątem konkretnego systemu zarządzania bazą danych. Konceptualna baza danych może być jeszcze dowolna, bowiem nie zależy od narzędzi wybranego DBMS.

Logiczna baza danych zależy od konkretnego narzędzia, jakie mamy do dyspozycji i musi się mieścić w jednym z trzech przedstawionych poniżej schematów.

Logiczne bazy danych

Wyróżniamy trzy podstawowe sposoby organizacji logicznej bazy danych:

- relacyjne bazy danych;
- hierarchiczne bazy danych;
- sieciowe (obiektowe) bazy danych.

Schematy logicznej bazy danych pozwalają nam przedstawić otaczający nas świat językiem baz danych (można powiedzieć, że schematy logicznej bazy danych to sposób, w jaki widzimy otaczający nas świat ale posługując się językiem baz danych). Każdą logiczną bazę danych należy doprowadzić do któregoś z trzech ww. schematów.

Relacyjne Bazy Danych

Najbardziej podstawową i najczęściej spotykaną strukturą logiczną baz danych jest właśnie relacyjna baza danych. Relacyjna baza danych to zbiór powiązanych ze sobą relacji. Relacja jest zbiorem obiektów charakteryzujących się identycznymi atrybutami. Tak więc jeśli tylko potrafimy zebrać różne obiekty o takich samych atrybutach, to z pewnością możemy te obiekty połączyć w relację.

Na rysunku 2.7 przedstawiono przykład relacji w postaci listy atrybutów, a na rysunku 2.8 przykład tej samej relacji w postaci tablicy:



Rysunek 2.7. Przykład relacji

STUDENCI

Nr_indeksu	Nazwisko	Imię	Data_urodzen
1001	Nowak	Jan	74-01-23
1005	Kowal	Anna	75-06-09

Kolumna

Krotka

Rysunek 2.8. Przykład relacji w postaci tablicy

Każdej kolumnie można przypisać określoną **szerokość** (długość) i **typ**.

Relacja – ma nazwę (np. STUDENCI)

Atrybuty – mają nazwę (np. Nr_indeksu, nazwisko, imię, datę urodzenia) oraz są określonego typu (np. liczbowy, tekstowy, data, logiczny); z typu wynika często długość pola;

Na rysunku 2.9 zebrano trzy podstawowe pojęcia związane z relacyjnymi bazami danych niezależnie definiowane poprzez:

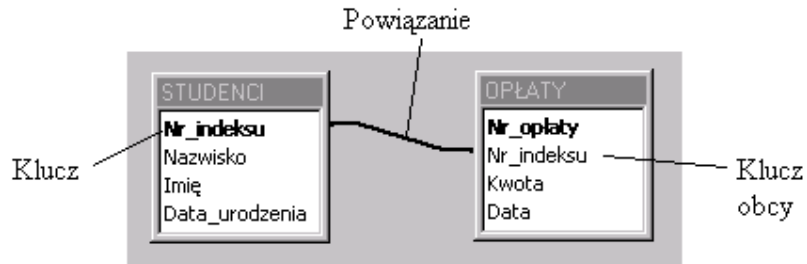
- teorię relacyjnych baz danych;
- język SQL do obsługi relacyjnych baz danych
- system operacyjny komputera i związany z tym system organizacji plików.

Bazy relacyjne	SQL	System operacyjny
Relacje	Tabela (tablica)	Plik (o dostępie bezpośrednim)
Krotka	Wiersz	Rekord
Atrybut	Kolumna	Pole

Rysunek 2.9. Podstawowe pojęcia związane z relacyjnymi bazami danych

Zbiór powiązanych ze sobą relacji (tabel) nazywany jest relacyjną bazą danych!

Powiązanie relacji realizowane jest za pomocą specjalnych atrybutów zwanych kluczami. Kolumna, która jednoznacznie określa daną krotkę nosi nazwę klucza. Kluczem jest kolumna (lub zbiór kolumn), która jednoznacznie określa **krotkę (wiersz)** np. *Nr_indeksu* jest przypisany tylko jednemu studentowi, a więc jednoznacznie go opisuje – jest **kluczem**. **Klucz główny** to klucz jednoatrybutowy. Na rysunku 2.10 przedstawiono dwie powiązane ze sobą relacje: STUDENCI oraz OPŁATY.



STUDENCI

Nr_indeksu	Nazwisko	Imię	Data_urodzen
1001	Nowak	Jan	74-01-23
1005	Kowal	Anna	75-06-09

OPŁATY

Nr_opłaty	Nr_indeksu	Kwota	Data
101	1005	100,00 zł	98-09-28
102	1001	100,00 zł	98-10-02
103	1005	200,00 zł	98-10-02

Rysunek 2.10. Powiązanie pomiędzy relacjami za pomocą kluczy

Klucz obcy – taki klucz, który w jakiejś innej tabeli jest kluczem głównym. Gdyby nie było kluczy obcych nie moglibyśmy łączyć ze sobą tabel.

Referencyjna integralność

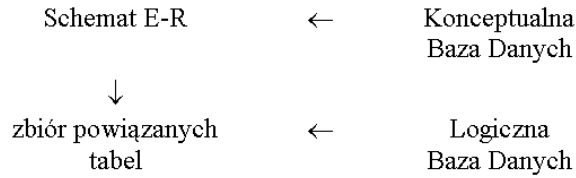
Referencyjna integralność jest wymaganiem stawianym relacyjnym bazom danych.

Każdy klucz obcy (child) musi mieć swego rodzica (parent).

Tak więc każdy klucz obcy musi być kluczem głównym w innej tabeli w tej samej bazie.

Powiązania pomiędzy tabelami

Jak już wspomniano w konceptualnej bazie danych operujemy schematami E-R (encja – powiązanie), z których to z kolei przechodzimy w sposób bezpośredni do logicznej bazy danych.

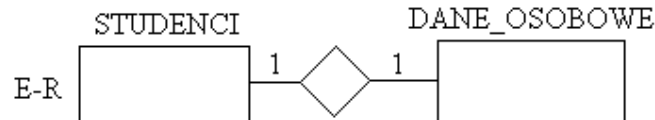


Najprostszym powiązaniem jest takie, w którym jednemu elementowi z jednej encji odpowiada dokładnie jeden element z innej encji.

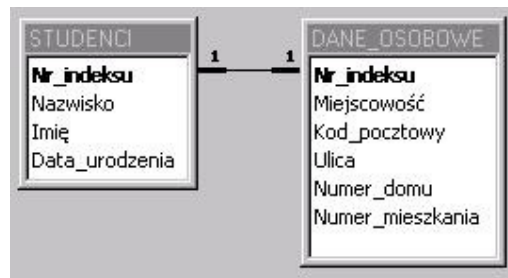
POWIĄZANIE "1:1" (JEDEN DO JEDNEGO)

Powiązanie to polega na tym, że jeden element danej encji odpowiada dokładnie jednemu elementowi innej encji.

Na rysunku 2.11 przedstawiono przykład schematu E-R z takim właśnie powiązaniem. Przy budowie modelu relacyjnego każdej encji można w naturalny sposób przypisać relację co pokazano na rysunku 2.12



Rysunek 2.11. Przykład schematu E-R z powiązaniem „1:1”



Rysunek 2.12. Przykład logicznej bazy danych uzyskanej ze schematu na rysunku 2.11

Ten typ powiązania znajduje zastosowanie w przypadku:

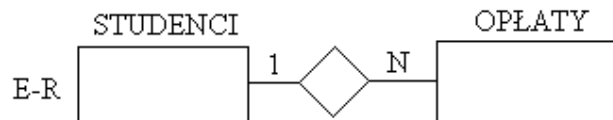
- potrzeby wprowadzenia uprawnień dostępu dla użytkowników do danych tabel; może istnieć potrzeba dostępu danego użytkownika do listy studentów, ale bez możliwości oglądania ich danych osobowych;

- potrzeby zwiększenia szybkości pracy bazy; nie zawsze konieczne jest korzystanie ze wszystkich danych np. z danych osobowych korzystamy stosunkowo rzadko, głównie potrzebujemy przetwarzać dane podstawowe studentów tzn. imię, nazwisko i numer indeksu.

POWIĄZANIE "1:N" (JEDEN DO WIELU)

Powiązanie to polega na tym, że **jednemu elementowi danej encji odpowiada wiele elementów drugiej encji**.

Na rysunku 2.13 przedstawiono przykład schematu E-R z takim właśnie powiązaniem. Tu również, przy budowie modelu relacyjnego każdej encji można w naturalny sposób przypisać relację co pokazano na rysunku 2.14



Rysunek 2.13. Przykład schematu E-R z powiązaniem „1:N”



Rysunek 2.14. Przykład logicznej bazy danych uzyskanej ze schematu na rysunku 2.13

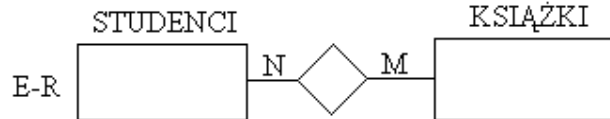
Powiązanie tego typu występuje najczęściej w relacyjnych bazach danych.

POWIĄZANIE "N:M" (WIELE DO WIELU)

Powiązanie to polega na tym, że **jeden element z jednej encji może być powiązany z wieloma elementami drugiej encji i odwrotnie**.

Na rysunku 2.15 przedstawiono przykład schematu E-R z takim właśnie powiązaniem. W przykładzie, modelującym proces wypożyczania książek każdy student może wypożyczyć wiele książek. Jednocześnie książka może być wypożyczona wielokrotnie różnym studentom.

W tym przypadku, przy budowie modelu relacyjnego każdej encji można w naturalny sposób przypisać relację co pokazano na rysunku 2.16, ale dodatkowo do zamodelowania powiązania „N:M” należy zastosować dodatkową relację (np. WYPOŻYCZENIA). Schemat z powiązaniem „N:M” trzeba więc rozbić na dwa powiązania „1:N” oraz „1:M”



Rysunek 2.15. Przykład schematu E-R z powiązaniem „N:M”



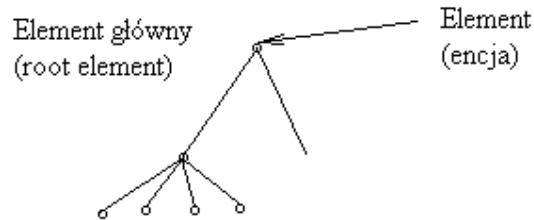
Rysunek 2.16. Przykład logicznej bazy danych uzyskanej ze schematu na rysunku 2.15

Należy zaznaczyć, że w relacyjnych bazach danych mogą występować wyłącznie powiązania **1:1** i **1:N**, zaś każde powiązanie typu **N:M** należy rozbić na powiązania **1:N** oraz **1:M**!

Szczególnym przypadkiem powiązania typu **N:M** jest tzw. **związek rekurencyjny**, tzn. takie powiązanie, w którym element danej tabeli jest powiązany z innym elementem tej samej tabeli. Związek rekurencyjny także musi być sprowadzony do postaci dwóch powiązań **1:N** oraz **1:M**.

Hierarchiczne Bazy Danych

Hierarchiczna baza danych polega na stworzeniu bardzo zhierarchizowanej struktury elementów co zilustrowano na rysunku 2.17.



Rysunek 2.17. Schemat hierarchicznej bazy danych

Wszystkie elementy (z wyjątkiem root) są podległe innym elementom. Każdy element (inny niż root) ma swego właściciela. Czyli struktura elementu jest następująca:

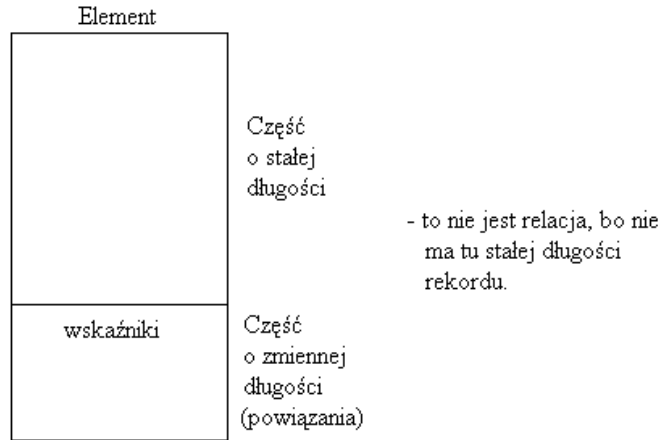
(adres, adres właściciela, ...inne_informacje_związane_z_elementem...)

Struktura ta jest łatwa do zorganizowania pod kątem przeglądania danych oraz charakteryzuje się dużą szybkością pracy aplikacji. Podstawową wadą jest jednak fakt, że struktury „czysto hierarchiczne” nie występują w życiu, nie znajdują więc zastosowania.

Hierarchiczne bazy danych należy więc traktować wyłącznie jako swego rodzaju pomysł na bazę danych, w praktyce nie istnieją takie struktury logiczne.

Sieciowe Bazy Danych

Tego typu struktury związane są z CAD, a dokładniej z zapisem dokumentacji rysunkowej. Odchodzimy więc tu od relacji, natomiast zakładamy, że pewne elementy mogą być ze sobą powiązane poprzez tzw. **system wskaźników** będących **adresami logicznymi**. Poniżej przedstawiono strukturę elementu.

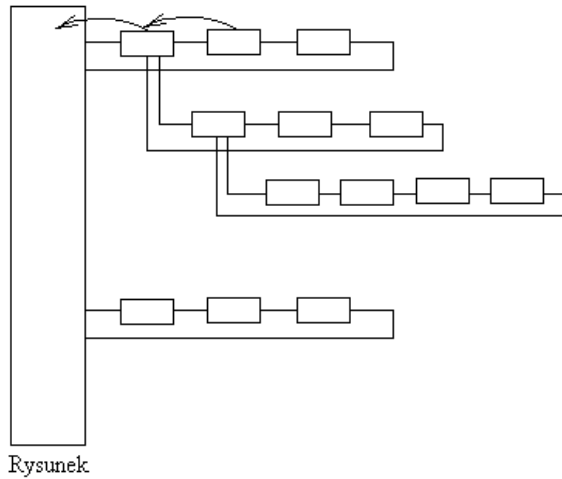


Rysunek 2.18. Struktura elementu w bazie sieciowej

Zwykle w edytorach rysunków CAD stosujemy tzw. bazy hierarchiczno-sieciowe.

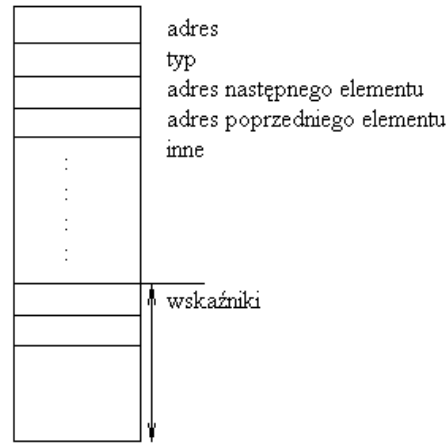
Hierarchiczno-sieciowe Bazy Danych

Tego typu struktura znajduje zastosowanie w edytorach rysunków CAD, przy czym z danym rysunkiem wiążemy elementy tego samego typu. Każdy element pamięta adres poprzedniego i następnego elementu.



Rysunek 2.19. Przykład logicznej bazy danych systemu CAD

Każdy element może być właścicielem innych elementów. Również elementy różnego typu mogą być ze sobą powiązane. Na rysunku 2.20 przedstawiono strukturę takiego elementu.



Rysunek 2.14. Przykład struktury elementu w bazie CAD

Warto zaznaczyć, że w bazach danych o takiej strukturze elementy są zazwyczaj predefiniowane.

Etapy projektowania baz danych

Projektowanie baz danych jest procesem podobnym do projektowania maszyn czy innych obiektów i składa się z następujących etapów:

I Założenia – na tym etapie należy określić podstawowe cele i parametry przyszłej bazy danych.

II Projekt wstępny – jest to najważniejszy z punktu widzenia modelowania etap budowy bazy; projekt wstępny zawiera w sobie przede wszystkim:

- schemat E-R,
- logiczną bazę danych jako konsekwencję schematu E-R,
- strukturę przyszłych użytkowników – określenie jacy użytkownicy lub grupy użytkowników i z jakimi uprawnieniami będą korzystać z bazy.

III Projekt szczegółowy – etap ten obejmuje sobą przede wszystkim projektowanie interfejsu poszczególnych użytkowników lub grup użytkowników:

- projektujemy poszczególne polecenia wchodzące w skład interfejsu – polecenia mogą mieć charakter zapytania, formularzy do wypełnienia przez użytkownika lub raportów; każde z poleceń może wiązać się z organizacją dialogu z użytkownikiem np. za pomocą specjalnie zaprojektowanych okien dialogowych
- polecenia formułujemy w postaci menu – powszechnie stosowaną techniką jest stosowanie grafiki do organizacji menu, w tym stosowanie odpowiednio zaprojektowanych piktogramów zamiast nazw poleceń czy dialogu graficznego zamiast dialogu alfanumerycznego

IV Programowanie i testowanie - etap ten polega na „przetłumaczeniu” projektu szczegółowego na program „zrozumiały” przez procesor komputera a następnie na przetestowaniu otrzymanego produktu. Do programowania można stosować dowolne języki programowania ale na ogół stosuje się specjalizowane narzędzia. Przy zastosowaniu tych narzędzi etap programowania może być stosunkowo prosty i szybki. Testowanie wymaga często wstępnego „nawodnienia” bazy i służy do usunięcia ewentualnych usterek.

V Instalacja i pielęgnowanie - etap ten obejmuje wszystkie czynności związane z instalacją i utrzymaniem (czyli pielęgnacją) bazy. Często tworzy się specjalny program instalacyjny a wdrożeniu bazy danych do funkcjonowania przedsiębiorstwa towarzyszy złożony system szkoleń a nawet zmian organizacyjnych. Pielęgnacja (ang. maintenance – utrzymanie) to w wielu przypadkach złożony proces nadzoru nad użytkowaniem bazy, połączony z bieżącą modyfikacją zarówno struktury logicznej jak i organizacji interfejsu poszczególnych użytkowników.

Do projektowania i budowy baz danych organizowane są często duże zespoły różnych specjalistów wyposażone w wyspecjalizowane narzędzia informatyczne.

Narzędzia do budowy baz danych

Jak już podkreślono w poprzednim punkcie do budowy baz danych można stosować dowolne języki programowania ale na ogół stosuje się specjalizowane narzędzia wchodzące w skład systemu zarządzania bazą danych (DBMS) albo bardziej specjalizowane narzędzia wchodzące w

skład technologii określanej mianem CASE (Computer Aided Software Engineering).

Z punktu widzenia generacji, języki programowania możemy sklasyfikować następująco:

1 GL	język wewnętrzny komputera – tylko program w 1GL jest „zrozumiały” przez procesor komputera i może być wykonany
2 GL	asemblerzy (w chwili obecnej praktycznie już nie wykorzystywane); w assemblerach wprowadzono adresy symboliczne; programy napisane w tego typu języku potrzebują jednak programów tłumaczących adresy symboliczne na adresy fizyczne.
3 GL	języki do przetwarzania numerycznego; języki te wyposażone są w translatory aby przetłumaczyć program w 3GL do programu w języku 1GL.
4 GL	języki do przetwarzania morfologicznego lub semantycznego; przykładem może być język SQL charakterystyczny dla relacyjnych baz danych lub LISP stosowany w bazach systemów CAD.

W językach 3GL, zwanych językami algorytmicznymi lub proceduralnymi, należy w programie źródłowym napisanym w tym języku (w postaci procedury) odpowiedzieć na pytanie JAK należy przetwarzać informacje np. poprzez napisanie stosownego algorytmu.

W językach 4GL, zwanych językami zapytań lub językami nieproceduralnymi, nie musimy już definiować JAK przetwarzać informacje ale wystarczy poprawnie sformułować CO ma być zrobione! Przykładem takiego języka jest SQL.

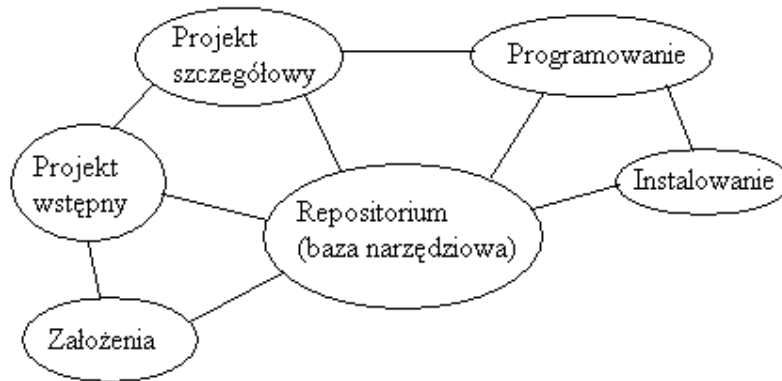
W interpretery zapytań napisanych w języku 4GL wyposażone są systemy zarządzania bazą danych (DBMS). DBMS służą również do projektowania bazy danych. Na rynku dostępnych jest wiele różnych systemów zarządzania bazą danych (DBMS). Przykładem DBMS dla baz relacyjnych może być produkt firmy Microsoft o nazwie MS Access. Przykładem DBMS dla baz CAD może być produkt firmy AutoDesk o nazwie AutoCAD.

Ze względu na coraz większą złożoność systemów baz danych dążymy do skomputeryzowania wszystkich faz procesu ich projektowania korzystając z narzędzi typu CASE.

Technologie CASE

Specjalizowane narzędzia wchodzące w skład technologii określanej mianem CASE (Computer Aided Software Engineering) obejmują wszystkie fazy budowy bazy danych. Narzędzia te mają strukturę bazodanową – umożliwiają jednoczesną pracę nad budową bazy przez wielu programistów i innych specjalistów zaangażowanych w budowę bazy np. przedstawicieli zamawiającego.

Systemy CASE mają swoją niezależną bazę narzędzi (repositorium) i z nich korzystają wszystkie osoby projektujące określoną bazę danych co przedstawiono na rysunku 2.15. Z repositorium korzysta się na wszystkich etapach projektowania bazy danych.



Rysunek 2.15. Struktura systemu CASE

Niektóre bazy (np. edytory rysunku) mają już predefiniowaną strukturę bazy danych. Natomiast mają one możliwość dostosowywania do indywidualnych wymagań użytkownika (tzw. kustomizacji) poprzez możliwość budowania własnych poleceń, własnej organizacji menu itp.

Warto zaznaczyć, że relacyjna baza danych także może posiadać predefiniowaną strukturę logiczną. Są to aplikacje, które po zainstalowaniu są od razu gotowe do pracy (pierwszym etapem pracy z tego typu bazą jest jej „nawodnienie”, aczkolwiek są bazy wstępnie nawodnione). W większości sytuacji projektujemy bazy danych od początku tj. zaczynając od określenia założeń.

Praktycznie niemożliwe jest kupienie gotowej bazy wspomagającej proces projektowania żurawi czy samochodów. Trzeba taką bazę

ROZDZIAŁ 2

budować od początku, często implementując w niej różne struktury (relacyjną, sieciową oraz hierarchiczno-sieciową).

3

Symulacja sytuacji niebezpiecznych w użytkowaniu maszyn

3.1. Wprowadzenie do symulacji

Symulacja komputerowa jest metodą prowadzenia eksperymentów, także z aktywnym udziałem człowieka, na modelach obiektów lub całych systemów. Stosowana jest niemal we wszystkich dziedzinach nauki i techniki, w tym szczególnie w problematyce bezpieczeństwa i higieny pracy.

Szczególnie ważne jest wykorzystanie komputerów do symulacji sytuacji niebezpiecznych w użytkowaniu maszyn.

Symulacja wymaga zastosowania różnych metod modelowania. Model powinien dostatecznie wiernie odtwarzać badaną sytuację rzeczywistą (np. zagrożenie, maszynę lub jej element) pod wybranym kątem widzenia, jednocześnie być możliwie prosty i łatwy do badania.

Rozróżniamy modele abstrakcyjne (wyrażające się w postaci pewnych pojęć, właściwych dla danej dziedziny) i modele materialne.

Mechanika klasyczna posługuje się *modelami abstrakcyjnymi*, jak np. punkt materialny (określony wartością masy), ciało sztywne (określone położeniem środka masy i macierzą bezwładności), ciało odkształcalne (określone dodatkowo macierzą sztywności). W wytrzymałości materiałów korzysta się np. z modelu pręta, belki lub płyty. W mechanice płynów modelem jest np. ciecz newtonowska.

Modelem materialnym drgań wybranych elementów maszyny może być przepływ prądu elektrycznego w przewodniku odpowiedniego obwodu. Jest nim też model geometrycznie podobny, w którym badane są te same zjawiska, co w obiekcie modelowanym, z zachowaniem pewnej skali.

Dalsze rozważania dotyczą modeli abstrakcyjnych.

Należy podkreślić, że zbytne uproszczenie modelu może doprowadzić do pominięcia istotnych własności obiektów, a zbytne rozbudowanie — do niepotrzebnych kosztów związanych z jego budową i badaniem.

Proces modelowania abstrakcyjnego składa się z budowy modelu nominalnego i następnie modelu matematycznego.

Model nominalny stanowi wyidealizowany opis obiektu za pomocą pewnych pojęć. W mechanice modele nominalne (np. punkt materialny,

ciało sztywne, pręt) zwane są *modelami fizycznymi*, gdyż są wyrażane za pomocą pojęć fizycznych. W działalności inżynierskiej występują także problemy ekonomiczne, organizacyjne, społeczne itp., do modelowania których nie wystarczą pojęcia fizyczne. W dalszej części zajmować się będziemy modelami fizycznymi.

Informacje niezbędne do budowy modelu fizycznego można uzyskać w wyniku badań eksperymentalnych, krytycznej analizy podobnych problemów, korzystając z dotychczasowej wiedzy (publikacje) a także doświadczenia inżynierskiego. Proces modelowania polega na wydzieleniu obiektu z jego otoczenia z jednoczesnym ustaleniem jego struktury i analizą oddziaływań zewnętrznych i wewnętrznych. Im prostsza jest struktura, tym łatwiej ją badać. W układach mechanicznych istotną decyzją jest, czy układ należy traktować jako dyskretny (o skończonej liczbie stopni swobody), czy też ciągły (z nieskończoną liczbą stopni swobody).

Po utworzeniu modelu fizycznego można przystąpić do budowy *modelu matematycznego*. Ustala się zbiory zmiennych stanu (np. przemieszczeń) i parametrów (np. wymiarów), a następnie korzystając z praw fizyki (lub innych nauk) buduje się zależności matematyczne (równania lub nierówności). Model matematyczny odpowiada jednoznacznie modelowi fizycznemu.

Modele matematyczne powstałe z modeli fizycznych należą do tzw. *modeli numerycznych* (modele numeryczne nazywa się też modelami obliczeniowymi). W działalności inżynierskiej dużą rolę odgrywają także modele morfologiczne i semantyczne.

3.2. Technologia wirtualnej rzeczywistości

Rozwój techniki komputerowej, a szczególnie technologii rzeczywistości wirtualnej (VR - Virtual Reality) umożliwia uzyskanie realistycznie wyglądających „światów” i znajdujących się w nich obiektów trójwymiarowych (3D). Technologia VR rozszerza tradycyjną symulację komputerową o nowy interfejs „człowiek-maszyna”, który pozwala użytkownikowi nie tylko oglądać symulowany proces, ale także bezpośrednio oddziaływać na ten proces. Wirtualna rzeczywistość daje

użytkownikowi, którym może być projektant a także przyszły użytkownik, możliwość wnikania do wnętrza zamodelowanego „świata” za pomocą specjalistycznych narzędzi [6,7,8]. Takie podejście do symulacji jest bardzo pomocne przy inscenizowaniu, a także obserwowaniu wszelkich sytuacji niebezpiecznych. Umożliwia wirtualne, czynne uczestniczenie w procesie użytkowania.

Popularnym narzędziem pozwalającym na opis obiektów 3D w ich otoczeniu oraz na kreowanie wirtualnych światów, jest język modelowania rzeczywistości wirtualnej (*Virtual Reality Modeling Language* – VRML). Dzięki niemu można tworzyć interaktywne światy wirtualne, w których istnieje możliwość „chodzenia” pośród zdefiniowanych obiektów (wyświetlanie obiektów trójwymiarowych w czasie rzeczywistym), spotkania się z innymi uczestnikami danego świata (światy Multiuser), definiowania dowolnej ilości czujników interakcyjnych zmieniających wygląd świata w zależności od życzenia jego uczestnika. O atrakcyjności języka VRML decyduje:

- dostępność - czyli możliwość odwiedzenia świata wirtualnego za pomocą Internetu co oznacza, że jest on dostępny dla bardzo szerokiej grupy osób.
- możliwość ruchu w świecie wirtualnym - wszystkie obiekty na scenie trójwymiarowej można obejrzeć z każdej strony, ale nie na zasadzie wyświetlania ich kolejnych rzutów, czy też animacji (choć to też jest możliwe), ale poprzez „chodzenie” wokół nich. Nie można uzyskać tego efektu w tradycyjnych aplikacjach do grafiki trójwymiarowej takich jak CATIA czy 3DStudio. Dlatego twórcy grafiki w tych programach bardzo często uciekają się do konwersji swoich prac do formatu *.wrl.
- interakcyjność - każdy uczestnik „świata” wirtualnego może mieć wpływ na jego wygląd, jeśli twórca umieścił w opisie sceny różnego rodzaju czujniki. Otoczenie może się zmienić pod wpływem kliknięcia, „rzucenia okiem”, czy też zbliżenia się do danego obiektu. Bardzo ciekawe efekty można uzyskać wprowadzając skrypty VRML, które dają możliwość tworzenia mechanizmów interaktywnych.

Plik źródłowy VRML jest plikiem tekstowym ASCII i jest interpretowany przez przeglądarkę VRML na podobnych zasadach jak język HTML. Przeglądarka formatu VRML to zwykle dodatek (wtyczka, ang. plug-in) do przeglądarki internetowej. Podobnie jak przy interpretacji plików

HTML, wizualizacja jest sterowana przez komputer klienta w drodze wykonywania poleceń zawartych w pliku źródłowym. Większość oprogramowania dotyczącego VRML dostępna jest bezpłatnie na stronach internetowych. Najbardziej popularne przeglądarki, to CosmoPlayer i Cortona Client.

Opis sceny 3D (w języku VRML) przesyłany jest z serwera i odtwarzany w przeglądarce VRML. W pętli symulacji uaktualniany jest punkt widzenia, badane są zmiany wynikające z akcji podjętej przez użytkownika i ze sposobów zachowań obiektów sceny 3D, wreszcie renderowana jest scena i na ekranie monitora wyświetlany jest nowy obraz wirtualnego świata.

3.3. Przykłady wykorzystania wirtualnej rzeczywistości do symulacji sytuacji niebezpiecznych

W tym podrozdziale zostaną przedstawione dwa przykłady wykorzystania wirtualnej rzeczywistości do symulacji sytuacji niebezpiecznych. Pierwszy przykład dotyczy użytkowania wózków widłowych w transporcie wewnątrzzakładowym. Drugi przykład dotyczy wykorzystania wirtualnej rzeczywistości do optymalizacji linii montażowej przekładni zębatej.

Koncepcja aplikacji do symulacji pracy wózków widłowych

Pierwszy przykład dotyczy wykorzystania wirtualnej rzeczywistości symulacji sytuacji niebezpiecznych związanych z transportem wewnątrzzakładowym na hali zakładu produkcyjnego. Obiektem badań jest wózek widłowy, w którym znajduje się operator.

Według opracowań dokonanych przez amerykański Narodowy Instytut Zdrowia i Bezpieczeństwa Zawodowego (National Institute for Occupational Safety and Health – NIOSH), wózki widłowe należą do najbardziej niebezpiecznych w użytkowaniu maszyn [41]. W statystykach Instytutu NIOSH, dotyczących liczby obrażeń i chorób wywołanych wykonywaną pracą, całkowita liczba wypadków mających związek

z używaniem wózków widłowych wynosi rocznie ponad 18tys. Różne są skutki tych wypadków.

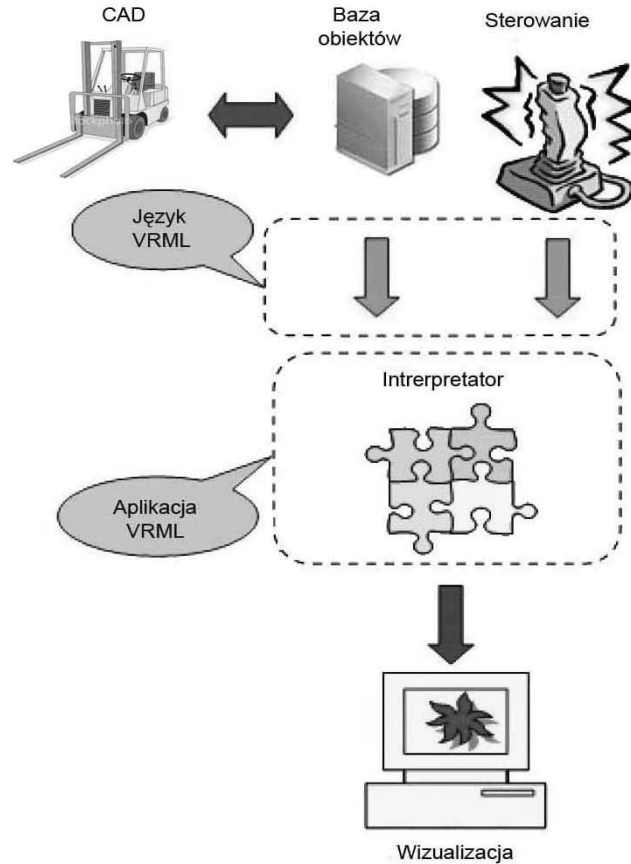
Dlatego w wielu krajach, w tym także w Polsce, prowadzi się liczne badania nad podniesieniem stanu bezpieczeństwa w użytkowaniu tych maszyn. We wszystkich badaniach [41] stwierdzono, że głównym źródłem zagrożenia jest ryzyko wywrócenia wózka i ograniczone pole widzenia podczas pracy.

Bezpieczeństwu użytkowania wózków widłowych poświęconych jest wiele przepisów i norm międzynarodowych i krajowych (np. [6,7,8,9]). Podstawowe parametry techniczne wózka widłowego są umieszczane na tabliczce znamionowej przez producenta wózka. Ze względów bezpieczeństwa, nie wolno przekraczać dopuszczalnego maksymalnego obciążenia podanego na tabliczce. Przepisy bezpieczeństwa podkreślają, że przy podniesionym ładunku wózek jest mniej stabilny i należy tak kontrolować położenie środka ciężkości, by nie dopuścić do przewrócenia się wózka (na przykład ładunek należy przewozić przy ustawieniu wideł 30 cm powyżej nawierzchni i maszynie przechylonym w stronę operatora). Do obsługi wózka widłowego konieczne jest posiadanie odpowiednich uprawnień oraz pozytywny wynik badań psychotechnicznych. Uprawnienia do obsługi wózków widłowych wydają firmy posiadające stosowny certyfikat. W Polsce najczęściej kursy obsługi wózków widłowych organizuje i wydaje uprawnienia Zakład Doskonalenia Zawodowego.

Szczególnie niebezpieczna jest jazda obciążonym wózkiem po łuku, gdy operator, poprzez nieodpowiednią obsługę, może narazić nie tylko siebie na niebezpieczeństwo, ale także innych pracowników. Symulacja w środowisku wirtualnej rzeczywistości, jazdy obciążonym wózkiem widłowym po łuku, jest głównym celem przykładu.

Na rysunku 3.1 przedstawiono schemat blokowy aplikacji do symulacji sytuacji niebezpiecznych związanych z procesem ładowania palety i procesem jazdy załadowanego wózka widłowego, w którym znajduje się operator na hali zakładu produkcyjnego.

Jednym z ważniejszych elementów koncepcji aplikacji jest system CAD. Zastosowanie programów CAD w aplikacji może dotyczyć: grafiki 2D - do tworzenia opisów, schematów, tekstury itd. i grafiki 3D - do tworzenia wirtualnego świata.



Rysunek 3.1. Schemat blokowy aplikacji

Obiekty wcześniej stworzone w programach CAD mogą być przesyłane i magazynowane w bazach obiektów. W bazie obiektów mogą znajdować się obiekty zapisane w formacie CAD jak i VRML. Istnieje możliwość bezpośredniego konwertowania modeli CAD na język VRML. Język VRML ma także możliwość modyfikacji obiektów już wytworzonych jak również daje możliwość tworzenia obiektów prostych. Jest także możliwość w przypadku powtarzania się elementu w „świecie” VR jego powielania co istotnie wpływa na zmniejszenie wielkości pliku VRML.

Następnym ważnym elementem aplikacji jest sterowanie które powoduje że aplikacja staje się interaktywna. Sterowanie staje się platformą łączącą użytkownika ze „światem” wirtualnym. Użytkownik może

ROZDZIAŁ 3

aktywnie ingerować w zawartość sceny i manipulować poszczególnymi obiektami, np. zabierać i przenosić różne przedmioty.

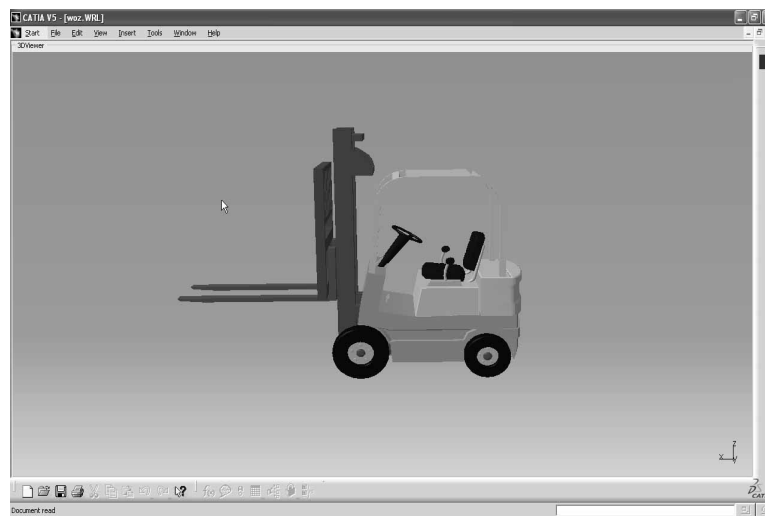
Wizualizacja jest niezbędnym etapem procesu symulacji. W zależności od potrzeb odbiorcy jest ona różnie realizowana. Daje też możliwość porównania wyników symulacji z rzeczywistymi sytuacjami niebezpiecznymi takimi jak załadunek, wyładunek, jazda po łuku.

Implementacja modelu symulacyjnego polega na sporządzeniu aplikacji z wykorzystaniem narzędzi symulacyjnych lub samodzielnie napisanego programu, działającego zgodnie z formalną specyfikacją modelu.

Obiekty wykorzystywane w aplikacji są modelowane w programach CAD (Catia V5 i 3ds Max) a następnie przekazywane do „świata” wirtualnego (do programu w języku VRML), za pomocą specjalnego narzędzia – jest nim program VrmIpad, za pomocą którego programista ma bezpośredni wgląd do składni języka VRML.

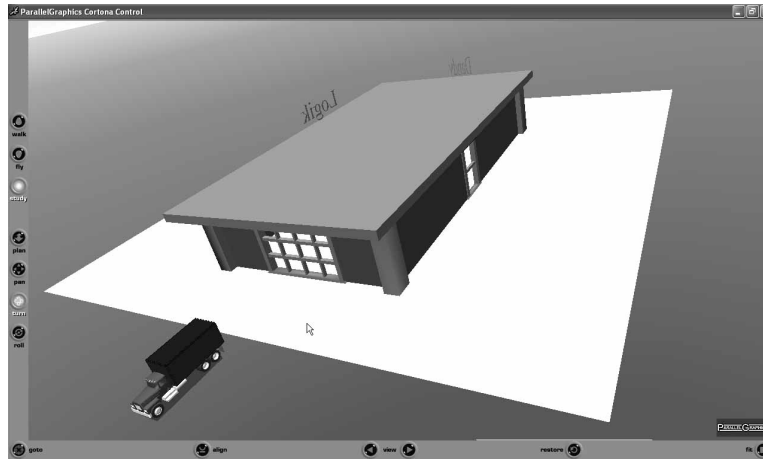
Aby umożliwić wykreowanie wirtualnego „świata” należało wcześniej stworzyć obiekty które tą rzeczywistość kreują.

W tym celu na wstępie zamodelowano wózek widłowy. Wymiary i budowa wózka zostały oparte na modelu FB30H-2R firmy KOMATSU. Na rysunku 3.2 przedstawiono kopię ekranu z modelem wózka.



Rysunek 3.2. Model wózka widłowego wykonany w CATIA V5

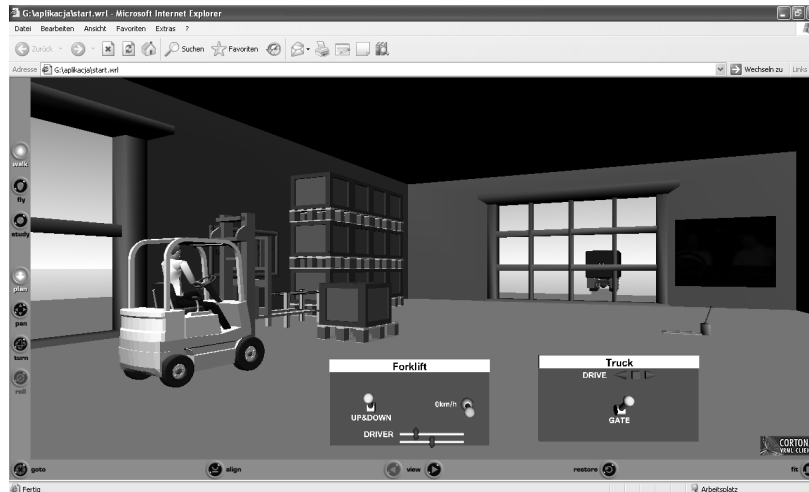
Następnie stworzono środowisko pracy wózka (rysunku 3.3) czyli halę produkcyjną, w obrębie której wózek widłowy może się poruszać.



Rysunek 3.3. Model hali produkcyjnej wykonany w CATIA V5

Sterowanie aplikacją

Użytkownik ma możliwość sterowania dzięki dwóm kokpitom „Forklift” i „truck”(rysunek 3.4).



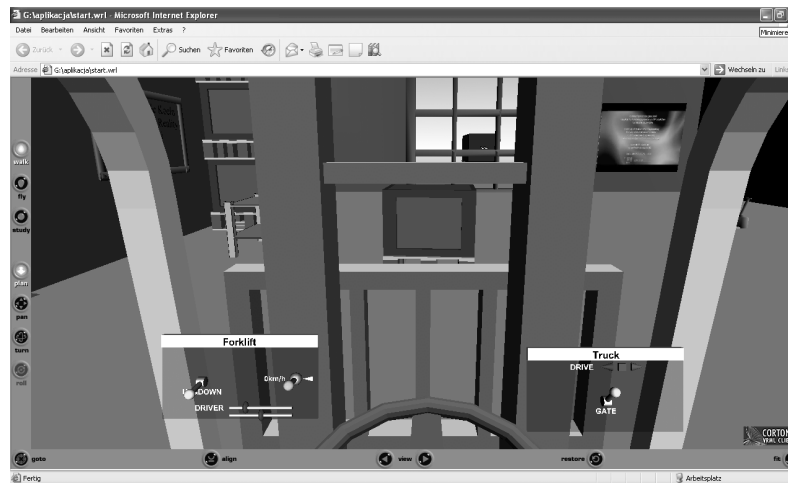
Rysunek 3.4. Możliwość sterowania aplikacją

ROZDZIAŁ 3

Kokpit „Forklift” umożliwia sterowanie jazdą samego wózka widłowego (DRIVE), sterowanie widłami wózka (UP&DOWN) oraz umożliwia obracanie głową operatora wózka zmieniając jednocześnie widok kamery podczas jazdy wózkiem.

Kokpit „Truck” umożliwia sterowanie jazdą ciężarówki, na którą (lub z której) ładowane są palety za pomocą wózka widłowego (DRIVE) oraz może służyć do otwierania i zamykania bramy wjazdowej do hali produkcyjnej (GATE).

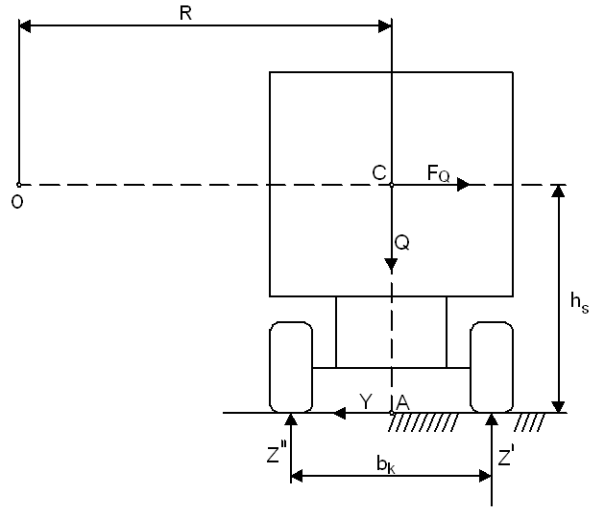
Na rysunku 3.5 przedstawiono kopię ekranu ilustrującą możliwość sterowania DRIVER, czyli możliwość obracania głową operatora wózka podczas jazdy.



Rysunek 3.5. „Świat” 3D widziany oczami operatora wózka widłowego

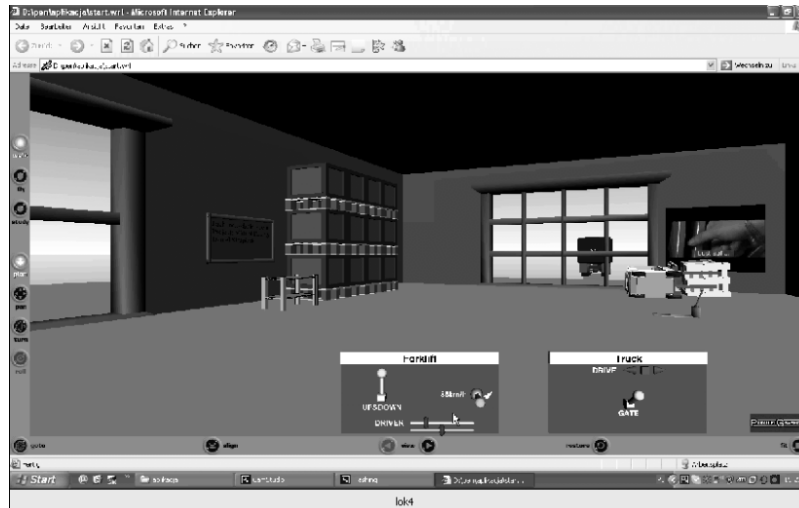
Symulacja jazdy po łuku

Do symulacji jazdy po łuku przyjęto bardzo uproszczony model dynamiczny wózka widłowego, który przedstawiono na rysunku 3.6.



Rysunek 3.6. Siły działające na wózek widłowy na łuku drogi

Na rysunku 3.6 pokazano układ sił działających na wózek widłowy na łuku drogi. Pominięto przy tym odkształcenia opon i zawieszenia oraz wynikający z nich przechył nadwozia pod działaniem siły bezwładności F_Q , przyjmując że wpływ tego przechyłu na obliczane wielkości jest niewielki. Program komputerowy na bieżąco analizuje, czy przy zadanym obciążeniu wózka i wysokości podnoszenia ładunku oraz przy zadanej prędkości liniowej i promieniu skrętu, nie nastąpi wywrócenie się wózka.

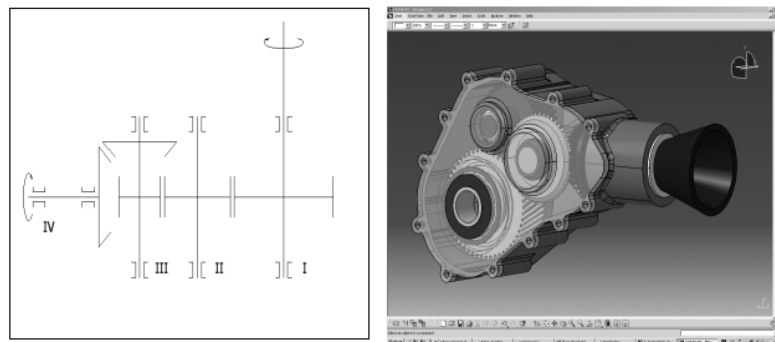


Rysunek 3.7. Kopia ekranu ilustrująca wywrócenie się wózka widłowego na łuku drogi

Na rysunku 3.7 przedstawiono kopię ekranu ilustrującą wywrócenie się wózka widłowego na łuku drogi.

Aplikacja do symulacji montażu przekładni zębatej

Drugi przykład wykorzystania wirtualnej rzeczywistości dotyczy symulacji i optymalizacji już istniejącej linii montażowej przekładni zębatej (do samochodu Mazda), przy aktywnym uczestnictwie decydentów – pracowników zakładu produkującego tę przekładnię.

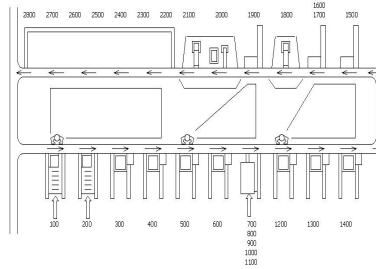


Rysunek 3.8. Schemat przekładni zębatej i jej model CAD

Na rysunku 3.8 przedstawiono schemat przekładni i jej model 3D powstały w systemie CATIA V5.

Na rysunku. 3.9 przedstawiono schemat linii montażowej tej przekładni (28 operacji).

Na rysunku 3.10 przedstawiono przykłady symulacji poszczególnych operacji montażowych.



Rysunek 3.9. Schemat linii montażowej przekładni



Rysunek 3.10. Symulacja poszczególnych operacji montażowych przekładni

W celu symulacji i optymalizacji procesu montażu zastosowano ten sam co w poprzednim przykładzie (rysunek 3.11) schemat blokowy do budowy aplikacji w środowisku wirtualnej rzeczywistości. Również wszystkie obiekty wykorzystywane w aplikacji, podobnie jak w poprzednim przykładzie, zostały zamodelowane w programach CAD (Catia V5 i 3ds Max) a następnie przekazane do „świata” wirtualnego

(do programu w języku VRML), za pomocą specjalnego narzędzia – jest nim program VrmlPad.

Następnie zaprojektowano system sterowania aplikacją, który powoduje że aplikacja staje się interaktywna. Dzięki odpowiedniej organizacji sterowania użytkownik może aktywnie ingerować w zawartość sceny i manipulować poszczególnymi obiektami, np. zabierać i przenosić różne przedmioty, w tym przykładzie, może wpływać na proces montażu przekładni.

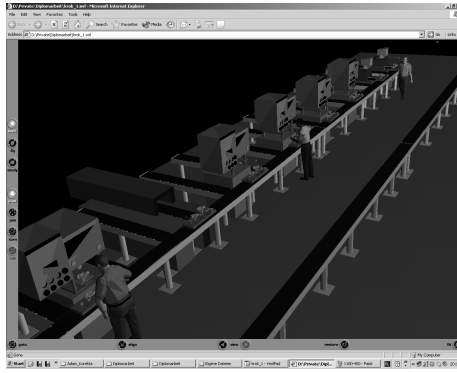
Opracowano również system wizualizacji wirtualnego procesu montażu. Na rysunku 3.11 przedstawiono kopię ekranu ilustrującą ogólny widok wirtualnej hali montażowej a na rysunku 3.12 kopię ekranu ilustrującą jedną z operacji montażowych.

W pierwszym etapie projektu zbudowano wirtualny model istniejącej hali montażowej i umożliwiający symulację montażu całej przekładni. Wyniki tego etapu zostały przedstawione do pozytywnej weryfikacji przedstawicielom zakładu produkującego przekładnię.

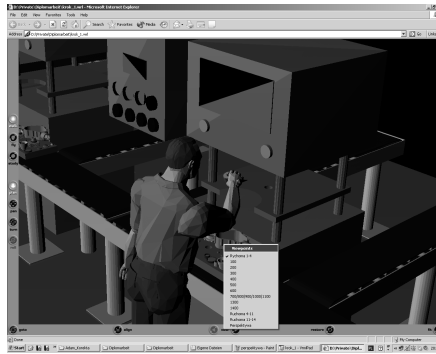
W drugim etapie, poświęconym optymalizacji montażu zmieniano w umiejętny sposób organizację procesu montażu, dążąc do jego poprawy. Dzięki wirtualnemu modelowi procesu montażu, zmiany te były łatwe do zrealizowania. Należy podkreślić, że w procesie takiej optymalizacji uczestniczyli pracownicy zatrudnieni w dotychczasowym procesie montażu a także eksperci od procesów montażu.

W projekcie uwzględniono różne kryteria optymalizacji w tym kryteria ekonomiczne (koszt i czas montażu przekładni, wymagana liczba pracowników i ich obciążenie pracą na poszczególnych stanowiskach), ale także kryteria związane z bezpieczeństwem, ergonomią i jakością. Niektóre z tych kryteriów były łatwe do obliczenia ale inne, głównie te związane z bezpieczeństwem i ergonomią, były szacowane dopiero podczas wizualizacji procesu montażu.

SYMULACJA SYTUACJI NIEBEZPIECZNYCH W UŻYTKOWANIU MASZYN

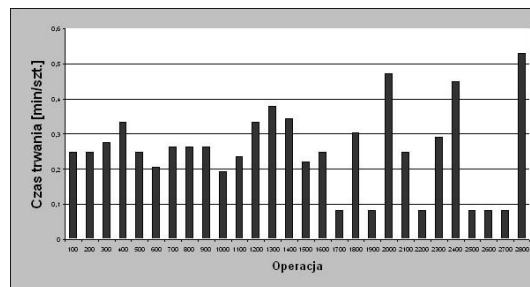


Rysunek 3.11. Wirtualna hala montażu przekładni



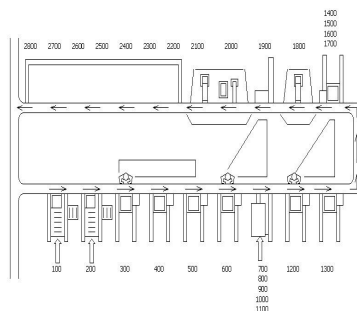
Rysunek 3.12. Wirtualne stanowisko montażowe

Na rysunku 3.13 przedstawiono czasy realizacji poszczególnych operacji montażowych dla jednego z analizowanych wariantów procesu montażu.



Rysunek 3.13. Wirtualne stanowisko montażowe

W wyniku optymalizacji linii montażowej (rysunek 3.14) uzyskano poprawę większości kryteriów.



Rysunek 3.14. Optymalny schemat procesu montażu

Takie nieklasyczne podejście do zadania optymalizacji z wykorzystaniem narzędzi wirtualnej rzeczywistości staje się skutecznym narzędziem szczególnie w zadaniach, gdzie w procesie decyzyjnym w aktywny sposób uczestniczy zespół ekspertów. Potwierdza to przykład optymalizacji linii montażowej przekładni zębatej.



Zakończenie

W podręczniku „Techniki komputerowe w bezpieczeństwie pracy” przedstawiono wybrane problemy komputerowego wspomaganie prac związanych z bezpieczeństwem człowieka w środowisku pracy i ergonomią” w tej dziedzinie.

W pierwszej części, dotyczącej wprowadzenia do komputerowego wspomaganie prac związanych z bezpieczeństwem człowieka w środowisku pracy i ergonomią, przedstawiono podstawowe idee projektowania wspomaganego komputerowo. W procesie projektowania kształtowane są bowiem podstawowe charakterystyki bezpieczeństwa maszyn i stanowisk pracy. W rozdziale tym przedstawiono też wybrane zagadnienia z teorii informacji.

W drugiej części opisano najważniejsze pojęcia związane z komputerowymi bazami danych. Technologie bazodanowe mogą być z powodzeniem stosowane w zagadnieniach związanych z bezpieczeństwem człowieka w środowisku pracy i ergonomią.

Część trzecia dotyczy symulacji sytuacji niebezpiecznych w użytkowaniu maszyn. Po krótkim wprowadzeniu do symulacji komputerowej opisano podstawowe możliwości technologii wirtualnej rzeczywistości. Przedstawiono też przykłady wykorzystania wirtualnej rzeczywistości do symulacji sytuacji niebezpiecznych. Pierwszy przykład dotyczy symulacji zagrożeń przy użytkowaniu wózków widłowych w transporcie wewnątrzzakładowym. Symulacja dotyczyła typowego i popularnego wózka widłowego firmy Komatsu FB30H-2R. Zaproponowano prosty model, w środowisku VRML, do symulacji pracy wózka, a szczególnie jazdy po łuku obciążonego wózka widłowego. Przedstawiono i przeanalizowano przykładowe wyniki symulacji. Zbudowana aplikacja może być z wykorzystana zarówno przez projektantów wózków widłowych, inżynierów zajmujących się organizacją transportu wewnątrzzakładowego jak i przez przyszłych operatorów wózków widłowych w procesie ich szkolenia. Drugi przykład dotyczy wykorzystania wirtualnej rzeczywistości do

optymalizacji linii montażowej przekładni zębatej w wykorzystaniu kryteriów bezpieczeństwa. Symulacja komputerowa podobnie jak technologie bazodanowe może być z powodzeniem stosowana w wielu zagadnieniach związanych z bezpieczeństwem człowieka w środowisku pracy i ergonomią.



Bibliografia

1. Beynon-Davies P.: Systemy baz danych, WNT, Warszawa 1998
2. Bezpieczeństwo pracy i ergonomia, praca zbiorowa pod red. D. Koradeckiej, Wydawnictwo CIOP, Warszawa, 1997
3. Celko J.: SQL – zaawansowane techniki programowania, Wydawnictwo Mikom, 1999.
4. Grieves, Michael: Product Lifecycle Management: Driving the Next Generation of Lean Thinking (Hardcover), McGraw-Hill, 2006
5. Gruber M.: SQL, Wyd. Helion, 1966, Gliwice
6. ISO 1074:1991 Counterbalanced fork-lift trucks; stability tests
7. ISO 2328:1993 Fork-lift trucks; hook-on type fork arms and fork arm carriages; mounting dimensions
8. ISO 2330:2002 Fork-lift trucks - Fork arms - Technical characteristics and testing
9. ISO 3184:1998 Reach and straddle fork-lift trucks - Stability tests
10. Kernyskyy A.: Integration of Engineering Data Base for Small Machinery Building Enterprises, rozprawa doktorska, Politechnika Warszawska, 2003,
11. Laudański M., Wróbel J., Myśliwiec J.: Szacowanie kosztów w procesie projektowania pojazdów ratownictwa technicznego, XX Konferencja "Problemy rozwoju maszyn roboczych", Zakopane 2007, pp 1-6
12. Marowski W. (red.): Inżynierskie bazy danych w projektowaniu maszyn, WNT, Warszawa, 2002
13. Myrcha K.: Organizacja baz danych dotyczących wymagań z zakresu bezpieczeństwa w procesie komputerowo wspomaganego projektowania maszyn, rozprawa doktorska, Politechnika Warszawska, 1998

14. Okulicz K.: VR-based approach to manufacturing process planning, *International Journal of Production Research*, 2004, vol. 42, no 17, pp 3493-3504
15. Okulicz K., Sierpiński D., Wróbel J.: Symulacja sytuacji niebezpiecznych w użytkowaniu wózków widłowych w środowisku VRML, *Transport Przemysłowy*, no 2 (32), 2008, pp 56-58
16. Okulicz K., Wróbel J.: Some principles of an expert optimization of assembly, 19th International Conference on Production Research, Santiago de Chile, 2007, pp. 1-4
17. Okulicz K., Wróbel J. Non classical formulation and solution of design optimization problems in virtual reality environment, in "Design Methods for Industrial Design" edited by R. Rohatyński, 2008, pp 181-186
18. Osiński Z., Wróbel J.: Teoria konstrukcji, PWN, Warszawa, 1995.
19. Ruschitzka M., Suchodolski A., Wróbel J.: Ontologically-based approach for Knowledge Representation for Stamping Die Applications, *Computer-Aided Design & Applications*, 2009, no 6(3), pp 281-289
20. Saaksvuori, Antti: Product Lifecycle Management, Springer, 2008
21. Turski W.M., „Propedeutyka informatyki”. Wyd. III. PWN Warszawa 1989
22. Ullman J.D., Widom J.: Podstawowy wykład z systemów baz danych, Wydawnictwa Naukowo-Techniczne, Warszawa 2000
23. Weiss Z., Kasica M., Kowalski M.: Rzeczywistość wirtualna w projektowaniu wyrobów, Wyd. PP, Poznań, 2005
24. Wróbel J.: Technika komputerowa dla mechaników, PWN, Warszawa, 1994
25. Wróbel J.: Metody i środki projektowania wspomaganego komputerowo, *Przegląd Mechaniczny*, 2008, no 3, pp 13-16
26. 26. Wróbel J. (redaktor): Technika komputerowa dla mechaników. Laboratorium, Oficyna Wydaw. Pol. Warsz., Warszawa, 2004

27. Wróbel J., Fitowski M.: Adaptacja systemu CAD dla potrzeb kontroli wersji dokumentacji projektowej, *Przegląd Mechaniczny*, no 4, 2006, pp 27-30
28. Wróbel J., Fitowski M., Linkiewicz G.: Wykorzystanie Internetu do zarządzania dokumentacją projektową, *Przegląd Mechaniczny*, no 2, 2002 pp 35-39
29. Wrobel J., Laudański M.: Cost assessment in design of low volume manufacture machines, *Automation in Construction*, vol. 17 (2008), no 3, pp 265-270
30. Wróbel J., Mierzejewski R.: System jakości ISO 9000 w cyklu „życia” maszyny z wykorzystaniem systemów zarządzania dokumentacją inżynierską, *Przegląd Mechaniczny*, no 4, 2006, pp. 11-15
31. Wróbel J., Myśliwiec J.: Metody komputerowe szacowania ceny pojazdów specjalnych produkowanych jednostkowo, *Systemy Logistyczne Wojsk*, no 31, 2006, pp 417-426
32. Wróbel J., Raczko R.: Polyoptimal selection of rolling bearings, *Machine Dynamics Problems*, 2007, vol. 31, no 3, pp 105-108
33. Wróbel J., Raczko R.: Optymalizacja internetowego systemu do doboru elementów znormalizowanych, *Biuletyn Wojskowej Akademii Technicznej*, vol. LVI, no 1 s, 2007, pp 105-114
34. Wróbel J., Raczko R.: Zarządzanie dokumentacją elektroniczną w procesie eksploatacji maszyn z wykorzystaniem systemów klasy Workflow, XIII Szkoła komputerowego wspomaganie projektowania, wytwarzania i eksploatacji, Jurata, 2009, Wydawnictwa Wojskowej Akademii Technicznej, pp 311-316
35. Wróbel J., Sierpiński D., Okulicz K.: Simulation of loaded forklift overturn in virtual reality environment, in “Modelling and Simulation in Machine Design” edited by M. Dudziak, G. Domek, 2009, pp 33-41
36. Wróbel J., Szczepanik T.: Inżynierskie bazy danych w małych i średnich przedsiębiorstwach a problemy jakości, *Mechanik*, no 5-6, 2006, pp 532-534
37. Wróbel J., Szczepanik T.: Zastosowanie inżynierskich baz danych w celu redukcji kosztów produkcji w procesie nadzoru nad cyklem przezbrajania i kalibracji maszyn technologicznych,

- Biuletyn Wojskowej Akademii Technicznej, vol. LVI, no 1 s, 2007, pp 115-126
38. Wróbel J., Szczepanik T.: Zastosowanie inżynierskich baz danych w celu redukcji kosztów projektowania w małych i średnich przedsiębiorstwach, *Mechanik*, 2009, no 3, pp 234-236
 39. Wróbel J., Szlom M., Jaworski R.: Komputeryzacja obiegu dokumentów, *Przegląd Mechaniczny*, no 2, 2002 pp 5-6
 40. Wu Zhiyun: Application of Internet Tools to International Team Work in Machinery Design, rozprawa doktorska, Politechnika Warszawska, 2003
 41. Wu Zhiyun, Wróbel J.: Application of Internet Tools to International Team Work in Technical Documentation Exchange, *Journal of Transdisciplinary Systems Science*, vol.7, no 1, 2002, pp 589-594
 42. www.cdc.gov/niosh/

