

Dr inż. Sebastian RZYDZIK, e-mail: [sebastian.rzydzik@polsl.pl](mailto:sebastian.rzydzik@polsl.pl)  
Politechnika Śląska

## SYSTEM GROMADZENIA WIEDZY O KOMPONENTACH

**Streszczenie:** Przedstawiono prototyp systemu przeznaczonego do gromadzenia wiedzy o komponentach. W części poświęconej opisowi teoretycznemu przedstawiono różnice pomiędzy danymi, informacjami a wiedzą. Ponadto, wyszczególniono najważniejsze zagadnienia związane z projektowaniem i konstruowaniem maszyn i urządzeń opartym na wiedzy. Część teoretyczną uzupełniono o wykaz najważniejszych sposobów zapisu i reprezentacji wiedzy. Część związana z opisem utworzonego systemu informacyjnego zawiera wykaz założeń projektowych, charakterystykę wybranych narzędzi oraz opis jego struktury. Całość uzupełniono przykładami zastosowania opracowanego prototypu systemu.

**Słowa kluczowe:** projektowanie oparte na wiedzy, zarządzanie wiedzą

## THE SYSTEM TO COLLECTION OF KNOWLEDGE ON COMPONENTS

**Abstract:** The paper presents a prototype of a system to collection of knowledge on components. In the section on the theoretical description, was shown the differences between the concepts data, information and knowledge. In addition, was described the key issues related to the design process of machinery based on engineering knowledge. The theoretical part was supplemented by a list of the most important methods of recording and representation of knowledge. Part of the paper which is related to the description of the information system contains a list of project assumptions, a description of its structure and characteristics of the used tools. The paper was supplemented by examples of the application of developed system.

**Keywords:** Knowledge-Based Engineering, Knowledge Management

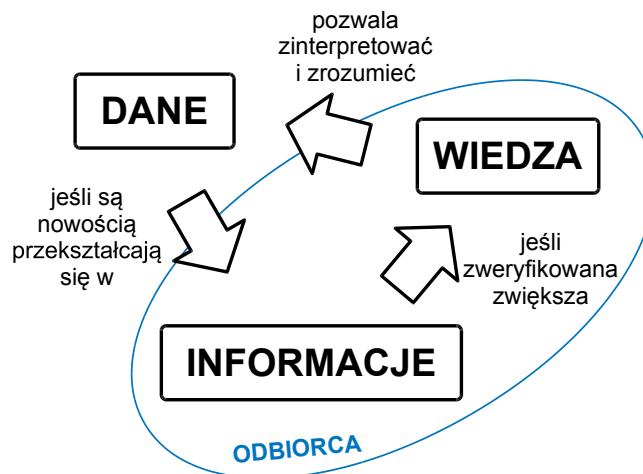
### 1. WSTĘP

Wiedza to rozumienie praw ogólnych i procedury postępowania oraz ogół wiarygodnych informacji o rzeczywistości wraz z umiejętnościami ich wykorzystywania [5]. Wyróżnia się wiedzę:

- *a priori*, która jest niezależna od zmysłów i dotyczy prawd „absolutnych” lub uniwersalnych, jakimi są prawa logiki, prawa matematyki;
- *a posteriori*, która jest wiedzą nabytą poprzez zmysły i jej prawdziwość może być obalona poprzez następne obserwacje.

Wiedza wynika z przetwarzania informacji, a te z przetwarzania danych [4][8]. Dane odzwierciedlają relacje pomiędzy symbolami lub znakami. Dane mogą być rejestrowane, przetwarzane i przesyłane w postaci *komunikatu*. Gdy dane są zrozumiałe dla odbiorcy, wtedy stają się *wiadomością*. Przykładem danych może być zapis  $d=40$  [mm]. Patrząc na ten ciąg symboli i odwołując się do nabytej wiedzy, odbiorca próbuje zinterpretować jego znaczenie. A zatem odbiorca analizuje otrzymaną wiadomość pod kątem znaczeniowym. Jeżeli treść wiadomości zawiera elementy nowe dla odbiorcy, to dane stają się *informacją*.

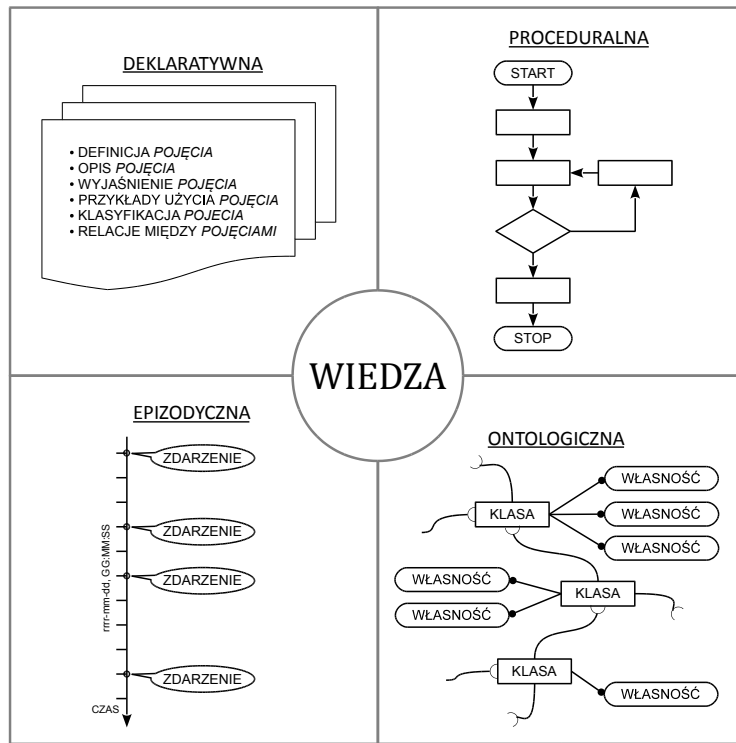
Inaczej, *informacje* są to dane zinterpretowane przez odbiorcę w określonym znaczeniu (w odniesieniu do sytuacji, zdarzenia, obiektu, czasu itp.), mające dla niego znaczenie i wnoszące do jego świadomości element nowości [11]. Przykładem informacji (np. danych rozpatrywanych w odniesieniu do zadania projektowo-konstrukcyjnego) może być zapis **średnica jest równa 40 milimetrów**. Powiązane ze sobą w sposób spójny i logiczny informacje, rozpatrywane w wybranej dziedzinie (ale niekoniecznie w tym samym znaczeniu) stają się *wiedzą*. Przykładem wiedzy (np. informacji rozpatrywanych w kontekście projektowania i eksploatacji maszyn) może być zapis **średnica wewnętrzna łożyska tocznego wynosi 40 milimetrów**. Zależności pomiędzy danymi, informacjami i wiedzą pokazano na Rysunku 1.



Rys. 1. Zależności między danymi, informacją i wiedzą [11].

Rozróżnia się następujące rodzaje wiedzy: deklaracyjną, proceduralną, epizodyczną i ontologiczną (Rys. 2.):

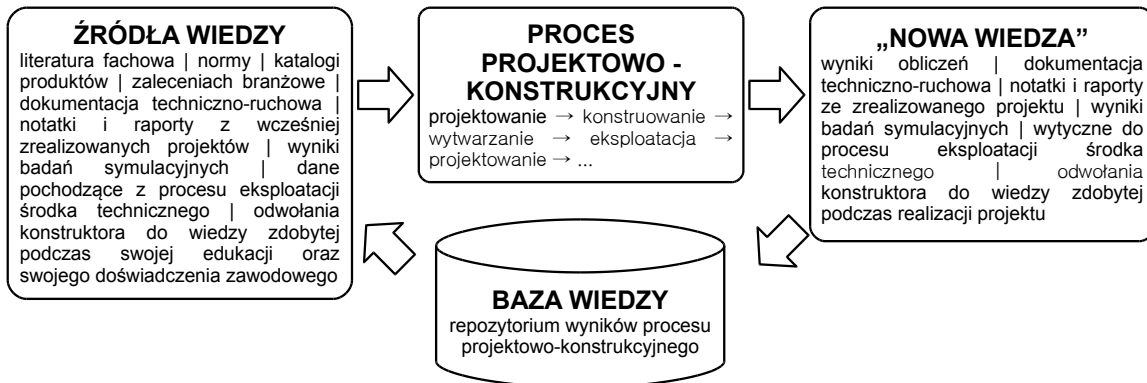
- *Wiedza deklaracyjna* dotyczy faktów, obiektów, relacji zachodzących pomiędzy obiektami itp. Wiedzę deklaracyjną można zapisać w postaci słownika pojęć. Z punktu widzenia systemów informatycznych, najlepszym sposobem gromadzenia wiedzy deklaracyjnej są encyklopedie, słowniki i leksykony, udostępniane na nośnikach elektronicznych lub w sieci Internet.
- *Wiedza proceduralna* jest najczęściej reprezentowana za pomocą procedur. Wiedza tego rodzaju jest przydatna podczas realizowania pewnego procesu, wymagającego określonych działań w określonych sytuacjach. Wiedza proceduralna jest zapisywana w postaci algorytmów.
- *Wiedza ontologiczna* wiąże się z podziałem danej dziedziny problemu na klasy (typy), własności egzemplarza danej klasy i relacje pomiędzy klasami oraz działaniami odwrotnymi, polegającymi na przypisaniu istniejącego obiektu do wcześniej zdefiniowanej klasy.
- *Wiedza epizodyczna* jest wiedzą zależną od czasu i pozwala opisać ciąg zdarzeń wstępujących w określonym czasie i miejscu.



Rys. 2. Rodzaje wiedzy [8].

### 1.1. Wiedza w procesie projektowania i konstruowania

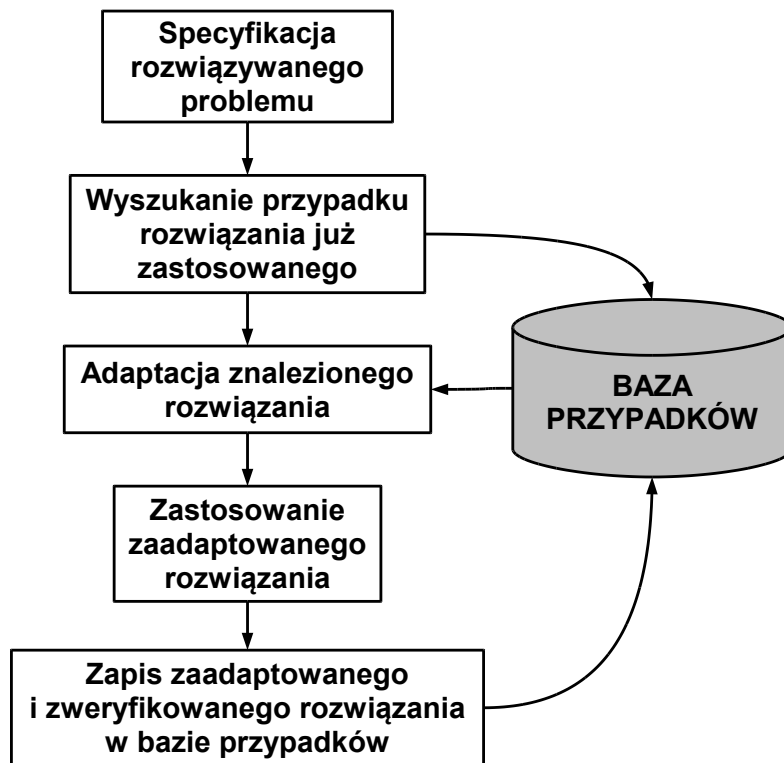
Generalnie, podczas realizacji procesu projektowo-konstrukcyjnego poszukiwanie rozwiązania postawionego zadania konstrukcyjnego polega na znalezieniu optymalnego stanu pomiędzy rozpatrywanymi cechami konstrukcyjnymi [8]. Przy czym na przyjęty zbiór postaci i wartości wybranych cech konstrukcyjnych mają wpływ także parametry związane z etapem produkcji i eksploatacji przyszłego środka technicznego, np. ograniczenia technologiczne, problem z dostępnością gotowych elementów, zakładana trwałość, poziom bezpieczeństwa użytkowania itp. Dlatego wiedza projektanta-konstruktora ma wiele źródeł, jak np. [1][2]: literatura fachowa, normy, katalogi gotowych produktów, zalecenia branżowe, dokumentacja techniczno-ruchowa, notatki i raporty z wcześniej zrealizowanych projektów, wyniki badań symulacyjnych, dane pochodzące z procesu eksploatacji środka technicznego, odwołania konstruktora do wiedzy zdobytej podczas swojej edukacji oraz swojego doświadczenia zawodowego. „Obieg” wiedzy w procesie projektowo-konstrukcyjnym zilustrowano na Rysunku 3.



Rys. 3. „Obieg” wiedzy w procesie projektowo-konstrukcyjnym.

Bardzo ważnym elementem pokazanym na Rysunku 3. jest *baza wiedzy*, która zawiera formalnie zapisaną wiedzę, uzyskaną w wyniku przeprowadzenia procesu pozyskiwania wiedzy. Według definicji może to być zbiór reguł, definicji i opisów faktów z określonej dziedziny, umożliwiający logiczny wywód wniosków lub podejmowanie decyzji. W przypadku realizacji procesu projektowo-konstrukcyjnego posiadana przez projektanta-konstruktora wiedza ma szczególnie duże znaczenie podczas rozwiązywania problemów *jak?* i *dlaczego?*. Problem *jak?* dotyczy zdolności użycia dostępnych narzędzi. Stopień opanowania narzędzia i umiejętność operowania nim oraz zakres wiedzy podręcznikowej zmniejsza istotność problemu *jak?* – *Jak użyć danej funkcji programu CAD? Jak użyć zależności pokazanej w podręczniku?*. Natomiast problem *dlaczego?* związany jest z doświadczeniem – *Dlaczego zastosowano tę funkcję programu CAD? Dlaczego użyto tej zależności? Dlaczego jest taka postać elementu?*. Problem *dlaczego?* jest trudniejszy do przedstawienia w sformalizowany sposób.

„Obieg” wiedzy stał się podstawą do opracowania metody wspomagającej rozwiązywanie nowych problemów na podstawie porównania i dopasowania odszukanych wyników rozwiązanych wcześniej problemów (*Case-Based Reasoning, CBR*) [6][7]. Schemat tej metody pokazano na Rysunku 4. Widoczna baza przypadków jest bazą wiedzy przechowującą przedstawione w sposób sformalizowany otrzymane rozwiązania.



Rys. 4. Schemat metody CBR [7].

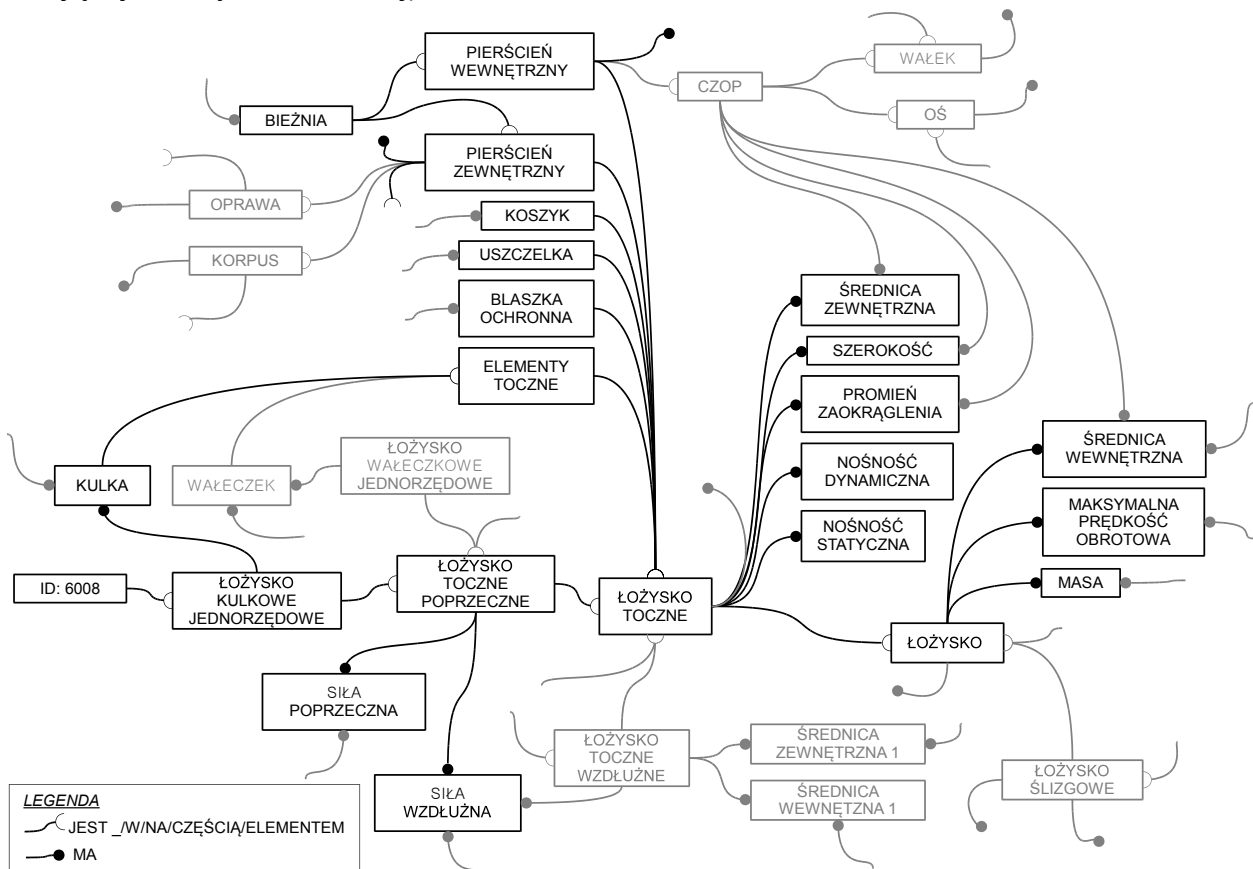
Stosowane w różnych formach bazy wiedzy oraz komputerowo zapisana wiedza stanowią podstawę do budowy systemów opartych na wiedzy (*Knowledge-Based Systems, KBS*). Szczególnym przypadkiem są systemy wspomagające procesy inżynierskie (*Knowledge-Based Engineering, KBE*), które wspomagają projektanta-konstruktora w realizowanych przez niego zadaniach [9][3]. Celem stosowania takich systemów jest automatyzacja

procesu tworzenia rozwiązań projektowo-konstrukcyjnych. Systemy KBE powinny realizować następujące cele [11]:

- Zapewniać zgodność konstrukcji z przyjętymi ograniczeniami projektowymi.
- Automatyzować dobór cech konstrukcyjnych (np. przez zastosowanie szablonów).
- Optymalizować konstrukcję w świetle przyjętych kryteriów.

## 1.2. Wybrane sposoby reprezentowania wiedzy

Reprezentacja wiedzy jest to sposób w jaki jest przedstawiana wiedza o świecie wraz z metodami przetwarzania i wnioskowania. Przy czym „świat” powinien być rozpatrywany w określonej dziedzinie i w określonym zakresie tej dziedziny. Reprezentacja wiedzy, to inaczej formalny język opisu wiedzy wraz z mechanizm przetwarzania wiedzy. Celem stosowania reprezentacji wiedzy jest gromadzenie wiedzy (na potrzeby jej archiwizacji i udostępniania) oraz umożliwienie łatwego przetwarzania wiedzy (na potrzeby odszukania istniejącej i odkrywania nowej).



Rys. 5. Przykład sieci semantycznej opisującej łożysko toczne [8].

Nie ma jednego sposobu reprezentowania wiedzy, który byłby najlepszy w zastosowaniu w każdej dziedzinie i dla każdego rodzaju wiedzy [8]. Dlatego istnieje wiele sposobów formalnego reprezentowania wiedzy: logika matematyczna (klasyczna, niestandardowa), zbiory przybliżone (rough sets), sieci semantyczne (semantic networks), grafy koncepcji (concept graphs), ramy, scenariusze (frames, scripts), sieci neuronowe (neural nets),

algorytmy genetyczne (genetic algorithms), diagramy UML (UML diagrams) oraz ontologie (np. standard OWL).

Na Rysunku 5. pokazano przykład zastosowania sieci semantycznej. Węzły w takiej sieci reprezentują obiekty, a gałęzie relacje pomiędzy tymi obiektami, np.: **łożysko toczne ma elementy toczne, pierścień wewnętrzny jest na czopie** itp. Poruszając się po gałęziach od obiektu do obiektu można prowadzić proces wnioskowania.

W wielu z wymienionych sposobów reprezentowania wiedzy stosuje się odpowiednio dostosowany język znaczników XML (meta-język, Wydruk 1.). Dostosowanie języka XML do reprezentowania wiedzy według określonego sposobu sprowadza się do opracowania odpowiedniego zbioru znaczników i atrybutów, które są później interpretowane przez oprogramowanie służące do zarządzania tak zapisanymi kolekcjami wiedzy.

Wydruk 1. Przykład formalnej struktury języka znaczników XML.

```
<?xml version="1.0"?>
  <znacznik1>dane1</znacznik1>
  <znacznik2>
    <znacznik2a atrybut="atrybut2a">dane2a</znacznik2a>
    <znacznik2b>
      <znacznik2b1>dane2b1</znacznik2b1>
      <znacznik2b2>dane2b2</znacznik2b2>
    </znacznik2b>
  </znacznik2>
</xml>
```

## 2. OPIS SYSTEMU

Zaproponowany system gromadzenia wiedzy o komponentach docelowo ma spełniać rolę bazy wiedzy zarówno dla użytkowników, jak i dla innych systemów komputerowych. Zaletą bazy wiedzy jest, to że można ją zdegradować do roli bazy danych. Dlatego też projektowany system może pełnić również rolę magazynu danych (np. danych katalogowych, wymiarów typoszeregów itp.). Należy zaznaczyć, że tutaj komponentem, w sensie technicznym, nazywany będzie zarówno element, jak i złożenie – podzespół, zespół, podukład, układ, czy megaukład.

### 2.1. Główne założenia

Przystępując do prac projektowych nad systemem gromadzenia wiedzy o komponentach przyjęto następujące założenia ogólne:

1. Wspomaganie procesu udostępniania wiedzy użytkownikom i innym systemom.
2. Wspomaganie procesu zapisu wiedzy (m.in. związków i relacji pomiędzy komponentami), w tym zapisu automatycznego.
3. Organizowanie procesu aktualizacji zapisanej wiedzy, w tym historii zmian.
4. Możliwość przetwarzania zgromadzonej wiedzy.
5. Możliwość rozszerzania funkcji systemu przez dodawanie własnych metod i funkcji.



## 2.2. Systemy zarządzania treścią

Analizując pole możliwych rozwiązań w zakresie istniejącego oprogramowania skupiono się przede wszystkim na programach z otwartym kodem źródłowym, udostępnianych na licencjach darmowych. Szczególną uwagę zwrócono na tzw. systemy zarządzania treścią (*Content Management System, CMS*), które zwykle mają postać oprogramowania, pozwalającego na tworzenie serwisów stron internetowych WWW. Podstawową cechą systemów klasy CMS jest to, że gromadzone w takim systemie treści są oddzielone od formy jej prezentacji. Zakres stosowania CMS obejmuje:

- zarządzanie dokumentacją (*Document Management*),
- zarządzanie danymi cyfrowymi (*Digital Asset Management*),
- zarządzanie zasobami informacyjnymi przedsiębiorstwa lub instytucji (*Enterprise Content Management*),
- zarządzanie treścią publikowaną na stronie internetowej (*Web Content Management*),
- zarządzanie wiedzą (*Knowledge Management*).

Wśród dostępnych rozwiązań systemów CMS, dostępne są systemy klasy Wiki. Systemy klasy Wiki są używane do pracy współbieżnej nad projektami, takimi jak repozytoria wiedzy lub projekty różnych grup społecznych. Przykładowym reprezentantem tej klasy systemów jest encyklopedia internetowa Wikipedia. W systemach Wiki treść można tworzyć i zmieniać z poziomu przeglądarki internetowej, za pomocą specjalnego języka znaczników (tzw. *wikikod*) lub dedykowanego edytora. Utworzona strona może być dowolnie modyfikowana przez wszystkie osoby uczestniczące w projekcie i posiadające odpowiednie uprawnienia. Do innych cech tych systemów można zaliczyć:

- Dowolne kształtowanie struktury treści i zasobów.
- Rozszerzalność wewnętrznego języka znaczników.
- Zapis dowolnie sformalizowanych treści.

W systemach Wiki treść można reprezentować i organizować jako:

- Stronę (*page*) – jest przeznaczona do przechowywania treści.
- Grupę stron (*group*) – jest przeznaczona do łączenia powiązanych (np. wybraną dziedziną wiedzy) ze sobą stron. Każdą grupę można dowolnie konfigurować niezależnie od pozostałych.

Przykład: `Main.HomePage`, gdzie `Main` – grupa stron; `HomePage` – strona.

- Szlak (*trail*) – jest przeznaczony do tworzenia sekwencji powiązań pomiędzy kolejnymi stronami lub grupami stron.

Przykład: `<< PreviousPage | Trail Index Page | NextPage >>`

- Kategorię (category) – jest przeznaczona do gromadzenia stron tematycznie związanych ze sobą.
- Farmę (WikiFarms) – jest przeznaczona do organizowania wzajemnej współpracy dwóch lub więcej równolegle uruchomionych systemów Wiki.

Różnorodność dostępnych sposobów organizowania stron oraz możliwość dodawania klasycznych odsyłaczy internetowych pozwala organizować treść i gromadzone zasoby w różne struktury: grafy, drzewa, listy itp.

Ponadto, każdy system Wiki pozwala na elastyczne kształtowanie struktury zabezpieczeń, co umożliwia dowolne definiowanie zakresu uprawnień użytkowników i grup użytkowników do stron i grup stron, a w tym przede wszystkim nadawanie i odbieranie uprawnień użytkowników do: edycji użytkowników, edycji ustawień systemu, przeglądania (czytania) stron, edycji (zapisu) stron, przeglądania historii zmian treści stron, zapisu plików w systemie itd.

Podstawowe funkcje systemów Wiki, to:

- Dodawanie stron|grup stron|użytkowników|grup użytkowników|uprawnień;
- Organizowanie stron|grup stron;
- Edycja stron|grup stron|użytkowników|grup użytkowników|uprawnień;
- Wyszukiwanie stron|grup stron|użytkowników;
- Gromadzenie i odtwarzanie historii zmian treści stron;
- Odtwarzanie poprzednich treści stron;
- Gromadzenie plików (zasobów cyfrowych);
- Gromadzenie i odtwarzanie rejestru zdarzeń w obrębie całego systemu.
- Możliwość rozszerzania dostępnych funkcji i metod przez dodawanie własnych modułów programowych.

Ostatecznie do budowy systemu przeznaczonego do gromadzenia wiedzy o komponentach postanowiono użyć oprogramowania o nazwie PmWiki dystrybuowanego na licencji GNU General Public License, dostępnego w Internecie pod adresem: <http://www.pmwiki.org>.

### 2.3. Struktura gromadzonych zasobów

W proponowanym systemie dane, informacje i wiedza gromadzone są z uwzględnieniem struktury:

**Produkt → Złożenia → Elementy**

lub w prostszej formie:

**Produkt → Elementy**



## Wydruk 2. Przyjęty format zapisu danych, informacji i wiedzy.

```
<xmlCAx>
  <xmlCAxProj>
    <xmlCAxDsc></xmlCAxDsc>
    <xmlCAxProp></xmlCAxProp>
    <xmlCAxView></xmlCAxView>
    <xmlCAxRefe></xmlCAxRefe>
    <xmlCAxTree></xmlCAxTree>
    <xmlCAxCnts></xmlCAxCnts>
    <xmlCAxMtrl></xmlCAxMtrl>
    <xmlCAxTool></xmlCAxTool>
    ...
  </xmlCAxProj>
  <xmlCAxAsm id="">
    <xmlCAxDsc></xmlCAxDsc>
    <xmlCAxProp></xmlCAxProp>
    <xmlCAxView></xmlCAxView>
    <xmlCAxRefe></xmlCAxRefe>
    <xmlCAxTree></xmlCAxTree>
    <xmlCAxCnts></xmlCAxCnts>
    <xmlCAxMtrl></xmlCAxMtrl>
    <xmlCAxTool></xmlCAxTool>
    ...
  </xmlCAxAsm>
  <xmlCAxPrt id="">
    <xmlCAxDsc></xmlCAxDsc>
    <xmlCAxProp></xmlCAxProp>
    <xmlCAxView></xmlCAxView>
    <xmlCAxRefe></xmlCAxRefe>
    <xmlCAxMtrl></xmlCAxMtrl>
    <xmlCAxTool></xmlCAxTool>
    ...
  </xmlCAxPrt>
</xmlCAx>
```

Na Wydruku 2. pokazano ogólną postać przyjętej struktury XML, która umożliwia zapis danych, informacji i wiedzy związanych z następującymi zasobami:

1. Zasób projektu (produktu)
  - a. Zasób nadrzędny projektu, który grupuje dane, informacje i wiedzę zapisane w zasobach podrzędnych. Identyfikator zasobu nadrzędnego:

**ID\_PROJ.ID\_PROJ**

- b. Zasoby podrzędne projektu produktu, które przechowują dane, informacje i wiedzę, zależne od rozpatrywanego kontekstu. Identyfikator zasobu podrzędnego:

**ID\_PROJ.ID\_PROJ-{type},**

gdzie {type}, to m.in.:

- **dscr** – opis produktu,
- **prop** – właściwości projektowanego produktu,

- **view** – model geometryczny produktu,
- **refe** – definicja obiektów referencyjnych przypisanych do produktu,
- **tree** – drzewo budowy produktu uwzględniające złożenia i elementy składowe,
- **cnts** – graf wiązań pomiędzy złożeniami i elementami składowymi,
- **mtrl** – wykaz tworzyw elementów składowych,
- **tool** – wykaz narzędzi i środków potrzebnych do czynności wytwórczych i eksploatacyjnych.

## 2. Zasób złożenia i elementu

- a. Zasoby nadrzędne złożenia (**Asm** – assembly) i elementu (**Prt** – part), które grupują dane, informacje i wiedzę zasobów podrzędnych. Identyfikator zasobu nadrzędnego:

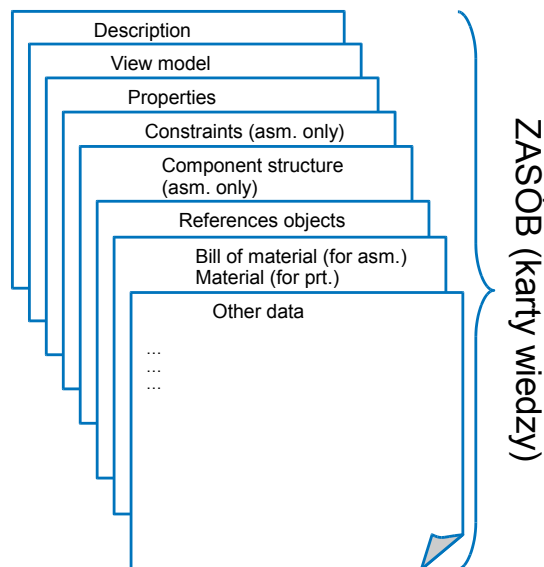
**ID\_PROJ.ID{Prt|Asm}**

- b. Zasoby podrzędne złożenia i elementu, które przechowują dane, informacje i wiedzę, zależne od rozpatrywanego kontekstu. Identyfikator zasobu podrzędnego:

**ID\_PROJ.ID{Prt|Asm}-{type}**

gdzie {type}, to m.in.:

- **dscr** – opis złożenia/elementu,
- **prop** – właściwości złożenia/elementu,
- **view** – model geometryczny złożenia/elementu,
- **refe** – definicja obiektów referencyjnych przypisanych do złożenia/elementu,
- **tree** – dotyczy tylko złożenia; drzewo budowy złożenia uwzględniające elementy i inne złożenia składowe,
- **cnts** – dotyczy tylko złożenia; graf wiązań pomiędzy elementami i innymi złożeniami składowymi,
- **mtrl** – tworzywo elementu lub wykaz tworzyw elementów składowych (dla złożeń),
- **tool** – wykaz narzędzi i środków potrzebnych do czynności wytwórczych i eksploatacyjnych.



Rys. 6. Karty wiedzy.

Zasoby gromadzone w systemie można porównać do formularzy papierowych (Rys. 6.), które przyjęto nazywać „kartami wiedzy”. Zbiór kart wiedzy stanowi zasób, a każda karta wiedzy zawiera dane, informacje i wiedzę na temat produktu, złożenia lub elementu zależne od rozpatrywanego kontekstu.

### 3. PRZYKŁADY UŻYCIA

Na Rysunku 7. pokazano fragment strony internetowej wygenerowanej przez system o roboczej nazwie wikiCAX. Opracowany prototyp systemu bazuje na oprogramowaniu PmWiki. Rozszerzenie istniejącego oprogramowania polegało na dodaniu funkcji, które mają za zadanie przetwarzanie danych, informacji i wiedzy zapisanej w przyjętym formacie zapisu danych (Wydruk 2.).

The screenshot shows the wikiCAX interface. At the top, there is a search bar and a language selector. The main content area displays the 'Description' and 'Properties' for a component named '05-00: Krążnik gładki'. The 'Description' section includes XML-like code for the component's name and manufacturer (IPKM). The 'Properties' section includes XML-like code for various dimensions and dynamic load capacity. Below the text is a technical drawing of a shaft with dimensions labeled: D=42mm, d=M16, b=40mm, LT=35mm, L1=210mm, L2=300mm, C0=4kN. The drawing shows a shaft with a central hole and a flange-like structure on the right side.

Rys. 7. Strona internetowa wikiCAX.

Rysunek 8. przedstawia przykład wikikodu dla zasobu nadrzędnego. W systemie wikiCAX do dołączania jednych stron do innych służy polecenie `include`, np. `(:include {IDZasobuPodrzedengo}:)`. Przedstawienie opisu zasobu w postaci `!$[ResourceName]` oznacza, że opis ma zostać wyróżniony (symbol `!`), i ewentualnie zastąpiony tłumaczeniem z innego języka niż język angielski (zapis `$(ResourceName)`). Dokładny opis stosowanego wikikodu można znaleźć w pomocy programu PmWiki.

Na Rysunkach 9-11 pokazano przykłady różnych zasobów podrzędnych. Zasób „Właściwości elementu” zawiera informacje o parametrach użytego lub skonstruowanego elementu. Można tutaj zapisać podstawowe cechy w różnej formie, np. jako wykaz kolejnych cech i ich wartości i/lub jako obraz graficzny. Odpowiednia funkcja systemu przetwarza zapisane informacje do postaci czytelnej dla użytkownika. Z drugiej strony, pierwotny zapis jest przechowywany i docelowo może zostać użyty w przypadku potrzeby udostępnienia tych danych innym systemom informacyjnym.

!\$[Description]  
(:include 05-00.05-00-dscr:)  
----

!\$[Properties]  
(:include 05-00.05-00-prop:)  
----

!\$[View] model  
(:include 05-00.05-00-view:)  
----

!\$[References objects]  
(:include 05-00.05-00-refe:)  
----

!\$[Component structure]  
(:include 05-00.05-00-tree:)  
----

!\$[Constraints]  
(:include 05-00.05-00-cntr:)  
----


!\$[Bill of material]  
(:include 05-00.05-00-mtrl:)

Description

```
<xmlCaxDscr><cmpld>05-03</cmpld><shortDscr>Single Row Deep Groove Ball Bearings, DIN 625, 5x16x5mm, open</shortDscr><dscr>Groove ball bearings are not only suitable for high speeds, but also to take up radial and axial forces. Groove ball bearings are supplied "open" (default) or "closed" on both sides (Z2 version) or "closed and sealed" on both sides (ZRS
```

View model


```
<xmlCaxView><fileName>05-03-view.dwt</fileName></xmlCaxView>
```



Properties

```
<xmlCaxProp><type>61804</type><design>2RS1</design><d unit="mm">20</d><D unit="mm">32</D><B unit="mm">7</B><weight unit="g">19</weight><dynamicLoadCapacity unit="N">4000</dynamicLoadCapacity><tolerance>DIN 620</tolerance><bearingPlay>CN-normal</bearingPlay><unitStandard>metric</unitStandard></xmlCaxProp>
```

D=32mm  
d=20mm  
B=7mm  
C0=4kN  
type=2RS1



Bill of material

Component id	Type of material
05-03-01	100Cr6
05-03-02	100Cr6
05-03-03	100Cr6
05-03-04	100Cr6
05-03-05	100Cr6

References objects

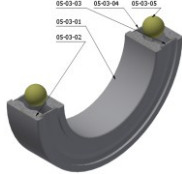
```
<xmlCaxRef><refObject type="axis">ref01</refObject><refObject type="plane">ref11</refObject><refObject type="plane">ref12</refObject><refObject type="plane">ref21</refObject><refObject type="plane">ref22</refObject></xmlCaxRef>
```

Component structure

```
<xmlCaxTree><Prt id="05-03-01">05-03-01</Prt><Prt id="05-03-02">05-03-02</Prt><Prt id="05-03-03">05-03-03</Prt><Prt id="05-03-04">05-03-04</Prt><Prt id="05-03-05">05-03-05</Prt></xmlCaxTree>
```

05-03

- 05-03-01 part
- 05-03-02 part
- 05-03-03 part
- 05-03-04 part
- 05-03-05 part



Rys. 8. Przykład zasobu nadrzędnego.

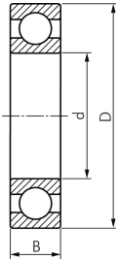
(:xmlCaxProp xml:

```
<xmlCaxProp>
  <D unit="mm">42</D><d unit="mm">16</d>
  <axisEnd>OT</axisEnd><b unit="mm">40</b>
  <LT unit="mm">35</LT><L1 unit="mm">210</L1>
  <L2 unit="mm">220</L2><L3 unit="mm">300</L3>
  <dynamicLoadCapacity unit="N">4000</dynamicLoadCapacity>
  <unitStandard>metric</unitStandard>
  <figureFileName>61804.png</figureFileName>
</xmlCaxProp>)
```

Properties

```
<xmlCaxProp><type>61804</type><design>2RS1</design><d unit="mm">20</d><D unit="mm">32</D><B unit="mm">7</B><weight unit="g">19</weight><dynamicLoadCapacity unit="N">4000</dynamicLoadCapacity><tolerance>DIN 620</tolerance><bearingPlay>CN-normal</bearingPlay><unitStandard>metric</unitStandard></xmlCaxProp>
```

D=32mm  
d=20mm  
B=7mm  
C0=4kN  
type=2RS1



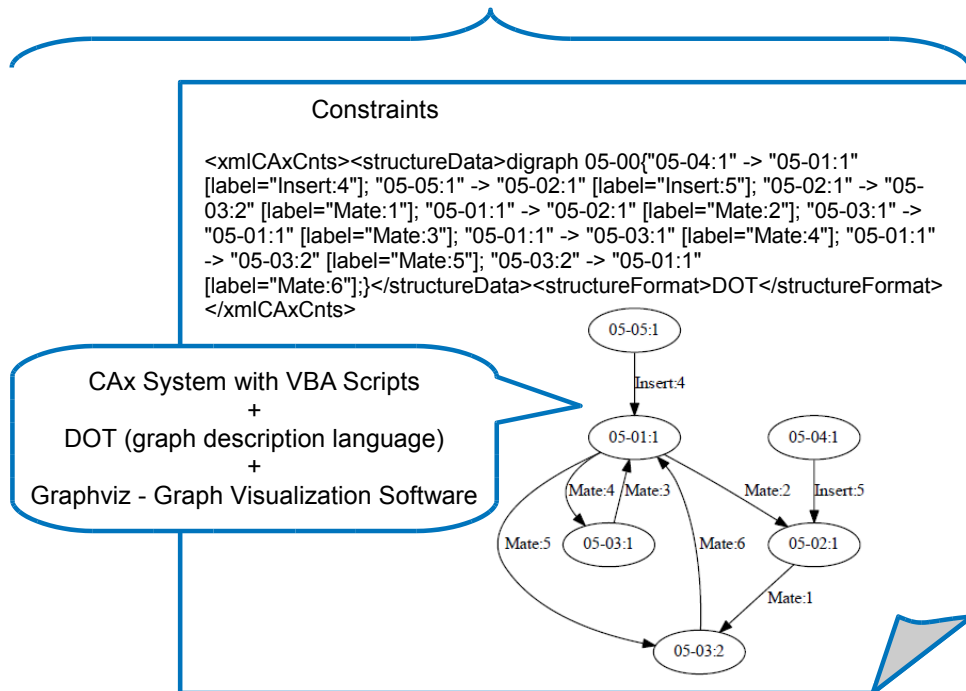
```
|| border=0
||D=42mm[<<<]
||d=M16[<<<]
||b=40mm[<<<]
||LT=35mm[<<<]
||L1=210mm[<<<]
||L2=300mm[<<<]
||C' 0' =4kN[<<<]
||Attach:05-00-draw.png ||
```

Rys. 9. Przykład zasobu podrzędnego. Właściwości elementu.

Zasób „Sieć wiązań” zawiera informacje o wiązaniach geometrycznych użytych w procesie tworzenia modelu geometrycznego złożenia. W tym celu, do zapisu grafu wiązań użyto języka DOT (*Graph Description Language*). Użycie oprogramowania do interpretacji języka DOT, jak np. Graphviz - Graph Visualization Software, umożliwia przetworzenie zapisu znakowego do formy graficznej, bardziej czytelnej dla użytkownika.

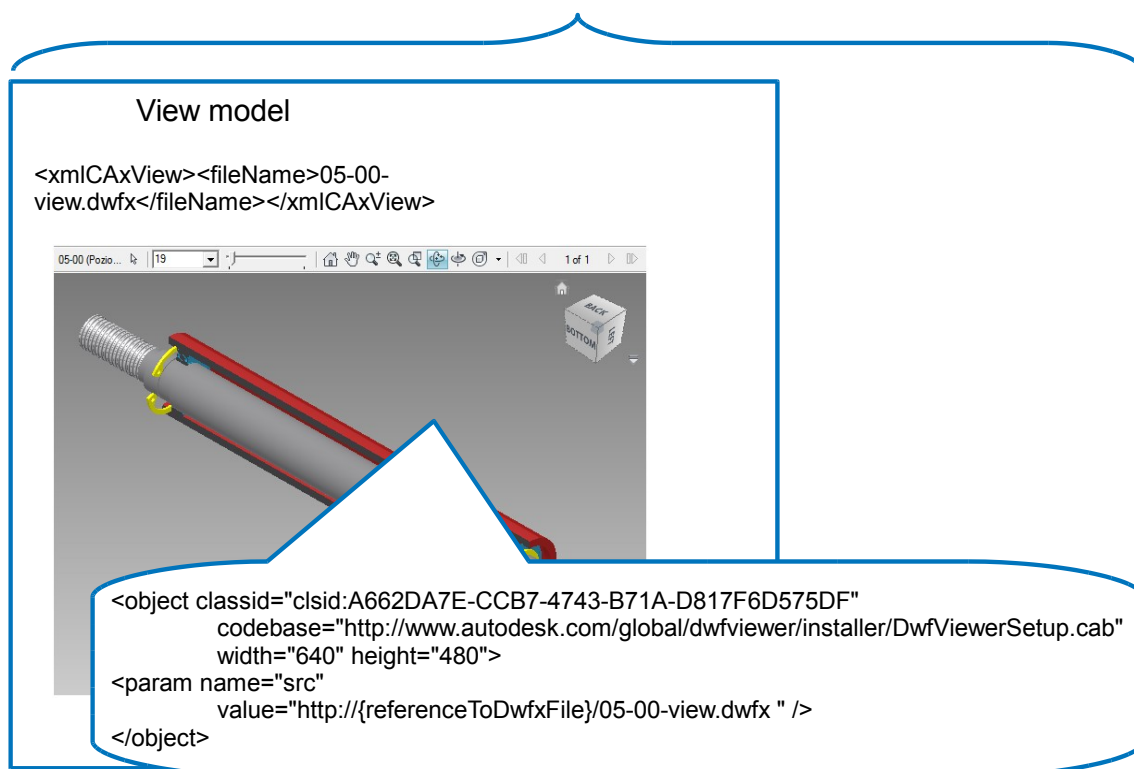
Zasób „Model geometryczny” jest przeznaczony do gromadzenia danych umożliwiających odwzorowanie modelu geometrycznego – płaskiego lub przestrzennego. Na pokazanym przykładzie model geometryczny został wygenerowany w programie Autodesk Inventor i zapisany formacie DWF (*Design Web Format*). Odpowiednia funkcja systemu wikiCAx po odzyskaniu wskazanego pliku z modelem geometrycznym oraz interpretacji jego formatu uruchamia dedykowaną przeglądarkę graficzną, która przetwarza dane zapisane w pliku.

```
(:xmlCAxProp xml:
  <xmlCAxCnts>
    <structureData>
      Digraph 05-00{
        "05-04:1" -> "05-01:1" [label="Insert:4"];
        "05-05:1" -> "05-02:1" [label="Insert:5"];
        "05-02:1" -> "05-03:2" [label="Mate:1"];
        "05-01:1" -> "05-02:1" [label="Mate:2"];
        "05-03:1" -> "05-01:1" [label="Mate:3"];
        "05-01:1" -> "05-03:1" [label="Mate:4"];
        "05-01:1" -> "05-03:2" [label="Mate:5"];
        "05-03:2" -> "05-01:1"[label="Mate:6"];}
    </structureData>
    <structureFormat>DOT</structureFormat>
  </xmlCAxCnts>:)
```



Rys. 10. Przykład zasobu podrzędnego. Sieć wiązań.

```
(:xmlCAxView xml:  
  <xmlCAxView>  
    <fileName>05-00-view.dwfx</fileName>  
  </xmlCAxView>:)
```



Rys. 11. Przykład zasobu podrzędnego. Model geometryczny.

#### 4. PODSUMOWANIE

Tworzony prototyp systemu informacyjnego przeznaczonego do gromadzenia wiedzy o komponentach jest oparty na oprogramowaniu klasy Wiki. Systemy klasy Wiki są przygotowane na zapis danych, informacji i wiedzy reprezentowanych w wielu, różnych formatach oraz pozwalają na dowolne kształtowanie struktur zapisanych treści. Ich rozbudowana struktura zabezpieczeń pozwala na elastyczną budowę zabezpieczeń i kontrolowanie poziomu dostępu do różnych fragmentów systemu. Tworzenie odwołań i odsyłaczy w ramach zgromadzonych danych, informacji i wiedzy pozwala na tworzenie związków semantycznych pomiędzy nimi.

Rozpoczęcie prac na tego typu systemem było spowodowane potrzebą zapisu nie tylko danych i informacji związanych z realizowanym procesem projektowo-konstrukcyjnym, ale również wiedzy konstrukcyjnej, wynikającej z tego procesu. Ważne jest, to żeby zapisana wiedza nie była wiedzą statyczną. Dlatego w chwili obecnej zaproponowany system wikiCAx wymaga opracowania metod wspomagających czynności zapisywania, przetwarzania i udostępniania danych, informacji i wiedzy, a w szczególności metod automatycznego zapisu wiedzy bezpośrednio z systemów CAx. Istnieje również potrzeba opracowania podsystemu obsługującego procesy zapytań, a w szczególności zapytania, których źródłem są inne systemy.



## LITERATURA

- [1] Cichocki P.: *Metodyka przechowywania wiedzy projektowej w budowie maszyn*. Instytut Podstaw Budowy Maszyn. Politechnika Warszawska, Warszawa 2001.
- [2] Dostatni E.: *Zarządzanie wiedzą o procesie projektowania*. Zeszyty naukowe Politechniki Poznańskiej. Budowa Maszyn i Zarządzanie Produkcją, Zeszyt 5, Poznań 2007.
- [3] Grana M., Toro C., Posada J., Howlett R.J., Jain L.C.: *Advances in Knowledge-Based and Intelligent Information and Engineering Systems*. Frontiers in Artificial Intelligence and Applications. Ios PressInc, 2012.
- [4] Niziński S., Żółtowski B.: *Informatyczne systemy zarządzania eksploatacją obiektów technicznych*. Uniwersytet Warmińsko-Mazurski w Olsztynie, Akademia Techniczno-Rolnicza w Bydgoszczy, Olsztyn - Bydgoszcz 2001.
- [5] Nowa Encyklopedia Powszechna, Tom 8. PWN, Warszawa 2004.
- [6] Pokojski J. (red.): *Zastosowanie metody case-based reasoning w projektowaniu maszyn*. WNT, Warszawa 2003.
- [7] Pokojski J.: *Systemy doradcze w projektowaniu maszyn*. WNT, Warszawa 2005.
- [8] Rzydzik S.: *Rozproszony system komputerowego wspomaganie procesu projektowo-konstrukcyjnego*. Monografia 475, Wydawnictwo Politechniki Śląskiej, 2013.
- [9] Sandberg M.: *Knowledge based engineering - in product development*. LTU-TR-0305-SE, Lulea University of Technology, Department of Applied Physics and Mechanical Engineering, Division of Computer Aided Design, February Sweden 2003.
- [10] Wełyczko A.: *Kierunki rozwoju systemów CAD: KBE (cz. I-IV)*. Design News Polska, maj-wrzesień 2007.
- [11] Zajac A., Grabowski M.: *Dane, informacja, wiedza - próba definicji*. Zeszyty Naukowe / Uniwersytet Ekonomiczny w Krakowie, nr 798, 2009, 99-116.