

**VI Seminarium**  
**„Wyzwania modelowania inżynierskiego i biznesowego”**  
**Centrum Studiów Zaawansowanych oraz Centrum Informatyzacji**  
**Politechnika Warszawska**  
**8.11.2016**

**Damian Krenczyk**

[damian.krenczyk@polsl.pl](mailto:damian.krenczyk@polsl.pl)

**Bożena Skołod**

[bozena.skolud@polsl.pl](mailto:bozena.skolud@polsl.pl)

**Planowanie wieloasortymentowej produkcji rytmicznej**  
**Zastosowanie symulacji jako narzędzia weryfikacyjnego**



Damian Krenczyk

Politechnika Śląska  
Instytut Automatykacji Procesów Technologicznych  
i Zintegrowanych Systemów Wytwarzania



# Problem

W praktyce przemysłowej wykorzystywanie systemów symulacyjnych jest ograniczone głównie do narzędzi służących do analizy i weryfikacji: koncepcji rozmieszczenia zasobów produkcyjnych, sposobu przepływu produkcji, reguł sterowania czy zagadnień związanych z ergonomią w nowo projektowanych zakładach produkcyjnych, a więc problemów należących do poziomów taktycznego oraz strategicznego.

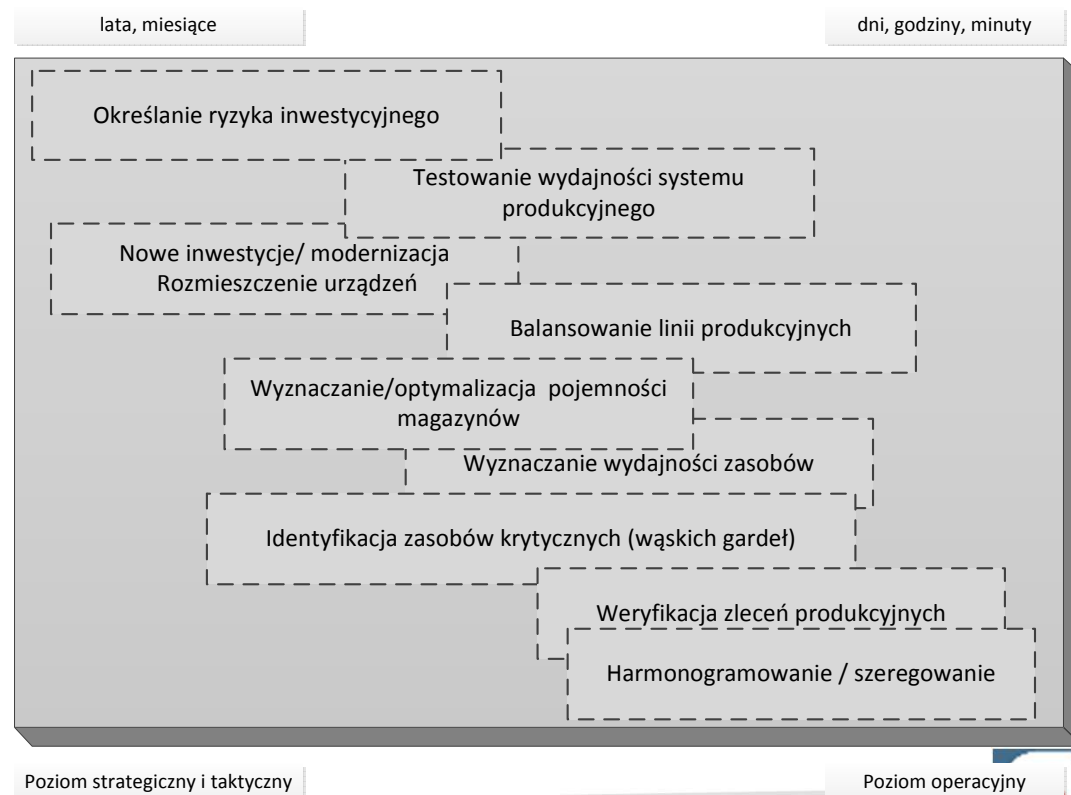
Wykorzystanie narzędzi symulacyjnych do wspomagania podejmowania decyzji na poziomie operacyjnym nie jest natomiast częste w praktyce. Taki stan rzeczy wynika przede wszystkim z następujących problemów:

- wymaganych dużych nakładów czasu pracy oraz kosztów związanych z przygotowaniem modelu symulacyjnego,
- konieczności zatrudniania wysoko wykwalifikowanego personelu, znającego zarówno metody tworzenia modeli symulacyjnych, jak i analizy systemów produkcyjnych,
- konieczności opracowywania nowych modeli symulacyjnych w przypadku zmian w strukturze systemu produkcyjnego, zmian zleceń produkcyjnych czy parametrów pracy,
- trudności z dostępem do danych wymaganych w trakcie budowy modelu symulacyjnego, wynikające z heterogeniczności stosowanych obecnie systemów wspomagania komputerowego w przedsiębiorstwie (PPC, CAx, MES, ERP).



# Dyskretne systemy symulacyjne we wspomaganiu podejmowania decyzji

30-40% czasu związanego z przeprowadzeniem projektu symulacyjnego zajmują czynności dotyczące gromadzenia niezbędnych danych, a 25-35% czasu poświęca się na przygotowanie samego modelu. Przeprowadzenie eksperymentu oraz analiza danych wyjściowych stanowią tylko 20% z całego czasu realizacji projektu symulacyjnego.



# Rozwiązanie?

Prowadzone obserwacje i badania wskazywały na duży potencjał stosowania narzędzi symulacyjnych do wspomagania procesu podejmowania decyzji na operacyjnym poziomie planowania. Możliwość uzyskania dużych korzyści w związku z wykorzystaniem symulacji komputerowej w planowaniu produkcji na poziomie operacyjnym, takich jak m.in.:

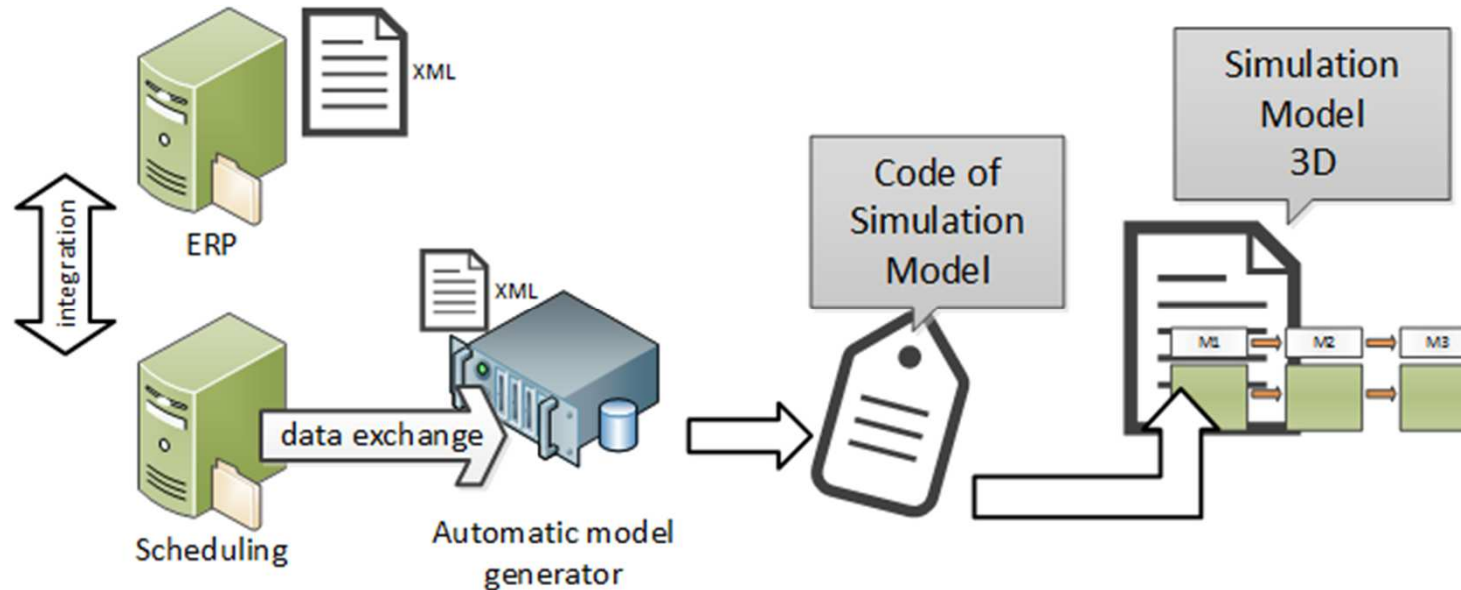
- weryfikacja możliwości realizacji produkcji dla opracowanych planów dotyczących kolejności realizacji zleceń oraz informacji dotyczących specyfikacji zlecenia produkcyjnego,
- zmniejszenie nakładów pracy nad wyborem harmonogramu,
- zwiększenie dokładności określenia terminów realizacji zleceń produkcyjnych,
- weryfikacja poprawności doboru środków transportu międzystanowiskowego i magazynowania,
- zapewnienie jakościowo dopuszczalnej realizacji produkcji w systemie – weryfikacja procedur sterujących pod kątem zapewnienia globalnej żywotności systemu,
- możliwość obserwacji przebiegu procesów w systemie dzięki wykorzystaniu wyników symulacji w procesach wizualizacyjnych.

## Wykorzystanie metod automatycznego generowania modeli symulacyjnych



# Podejście „Data-driven”

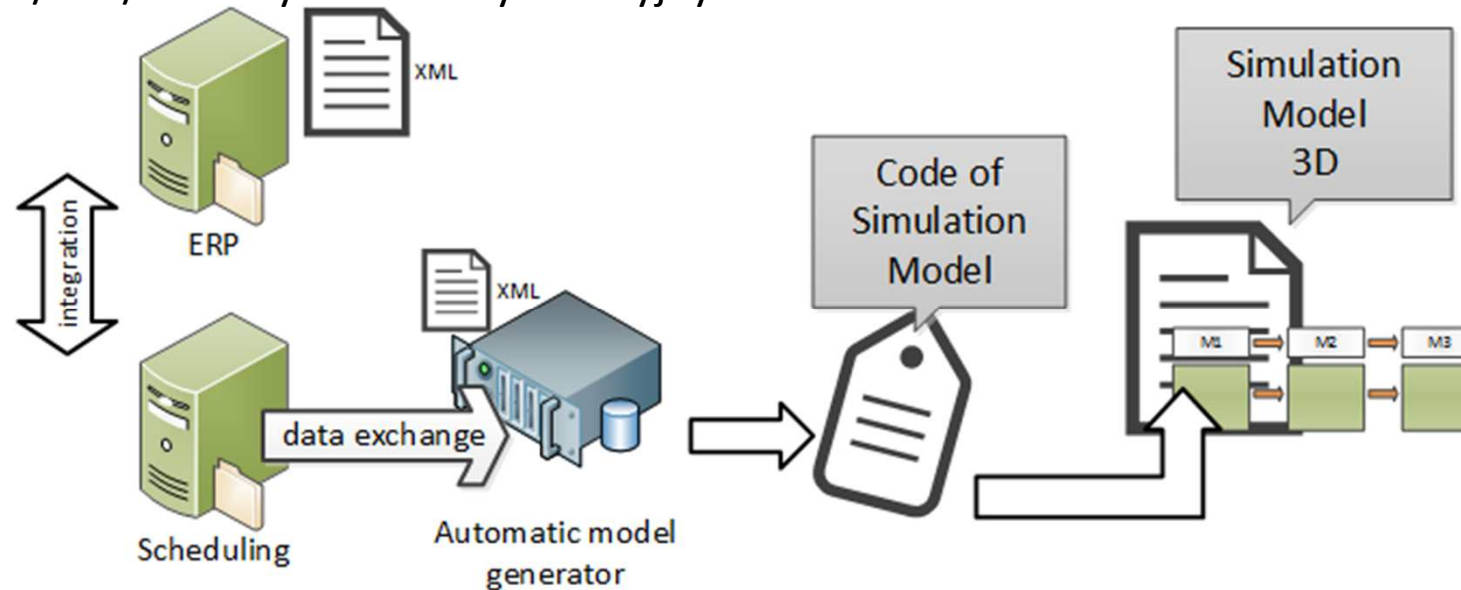
## Automatyczne generowanie modeli symulacyjnych



Koncepcja ta zakłada możliwość: pozyskiwania danych z różnych systemów wspomagających zarządzanie produkcją w przedsiębiorstwie, zapisywania/transformacji pozyskanych danych z wykorzystaniem języków modelowania, reprezentacji danych, a następnie automatycznego generowania modeli symulacyjnych dla dedykowanych systemów symulacyjnych.

# Podójście „Data-driven”

Rozwijany generator modeli symulacyjnych jest także narzędziem integracji systemów APS/ERP/PPC z systemami symulacyjnymi



model symulacyjny nie jest tworzony „ręcznie” przy użyciu narzędzia modelowania w systemie, lecz jest on generowany na podstawie zewnętrznego źródła danych przy użyciu interfejsów symulatora oraz algorytmy do tworzenia modelu.

Pozwala zmniejszyć ilość czasu potrzebnego do stworzenia modelu symulacyjnego, a także wiedzę niezbędną do tworzenia i prowadzenia symulacji.

# Integracja systemów planowania produkcji z systemami symulacyjnymi

Osiągnięcie postawionego celu zostało zrealizowane w czterech zasadniczych etapach:



Pozyskanie kompletnych danych wymaganych do przygotowania funkcjonalnego modelu symulacyjnego. Wynikiem realizacji tego etapu są definicje struktur danych dla systemów źródłowych.



Wymiana danych (Data Exchange) pomiędzy źródłową reprezentacją danych pozyskanych w etapie pierwszym a neutralnym pośrednim modelem danych. Wynikiem jest definicji pośredniego modelu danych.



Dane z modelu pośredniego są transformowane z wykorzystaniem autorskiej metody transformacji danych na kod programowy tworzony w wewnętrznych językach programowania systemów DES.



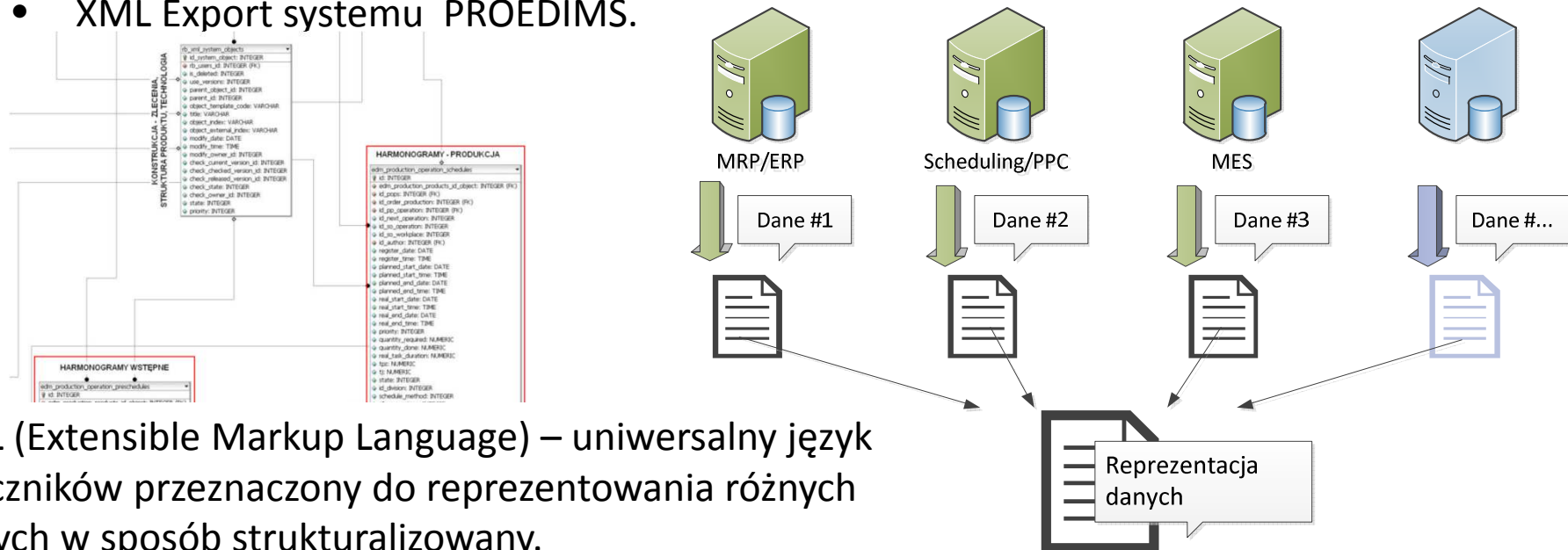
Połączenie kodu skryptów (z etapu trzeciego) w jeden dokument, zawierający dane o zasobach systemu produkcyjnego wraz z definicją sparametryzowanych informacji o realizowanych w systemie procesach produkcyjnych, procedurach sterujących przepływem tych procesów na zasobach systemów, danych o magazynach oraz środkach transportu i zaimportowanie go do systemu DES w celu przeprowadzenia eksperymentów symulacyjnych.



# Model danych

Przykładowe moduły systemów klasy PPC/ERP, pozwalające na pozyskiwanie wymaganych danych w postaci dokumentów XML:

- NetWeaver systemu SAP,
- IFS Connect systemu IFS Application,
- QXtend Data Import/Export firmy QAD,
- XML Export systemu PROEDIMS.



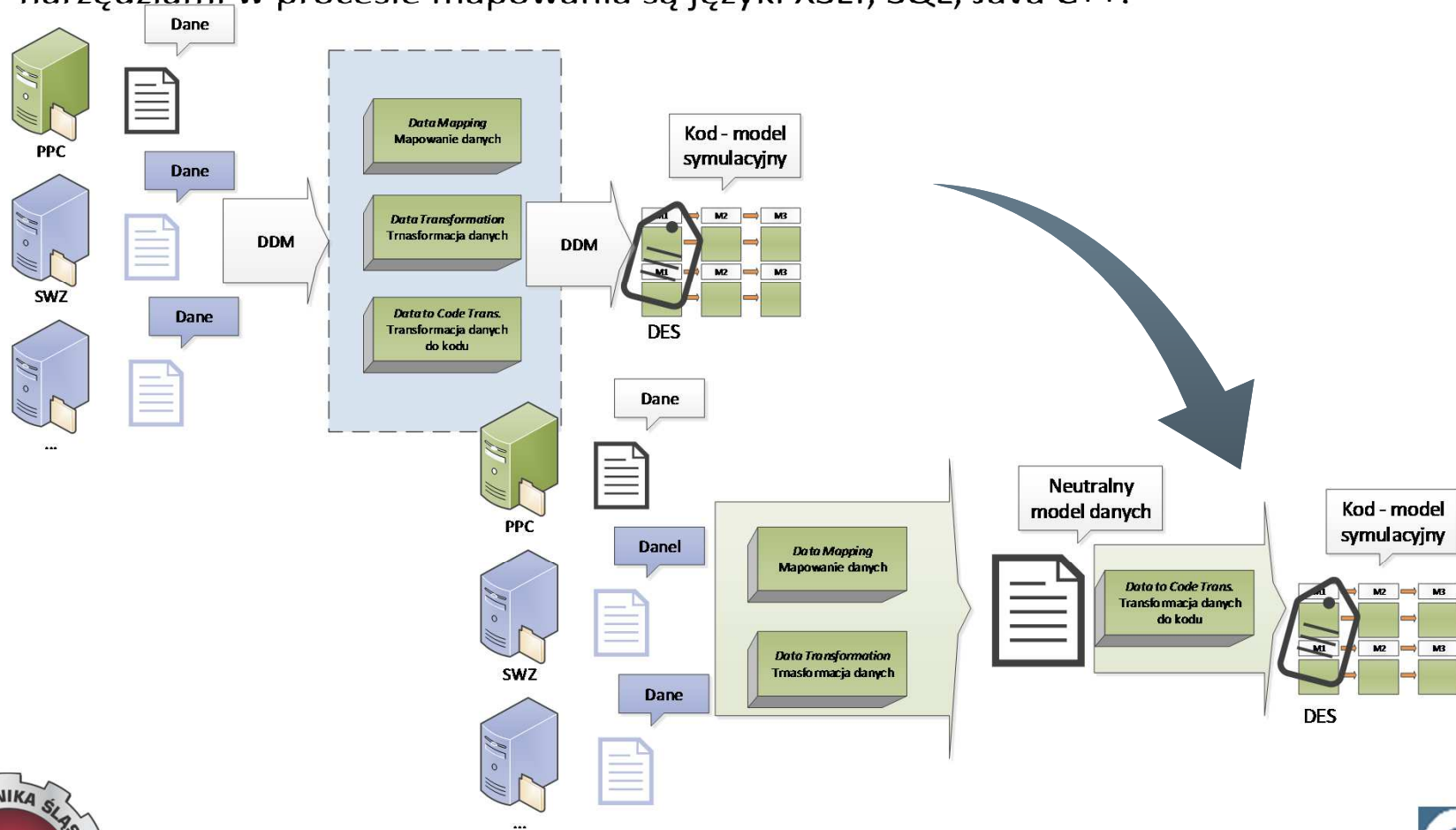
XML (Extensible Markup Language) – uniwersalny język znaczników przeznaczony do reprezentowania różnych danych w sposób strukturalizowany.

XML jest niezależny od platformy, co umożliwia łatwą wymianę dokumentów pomiędzy heterogenicznymi systemami



# Mapowanie i wymiana danych

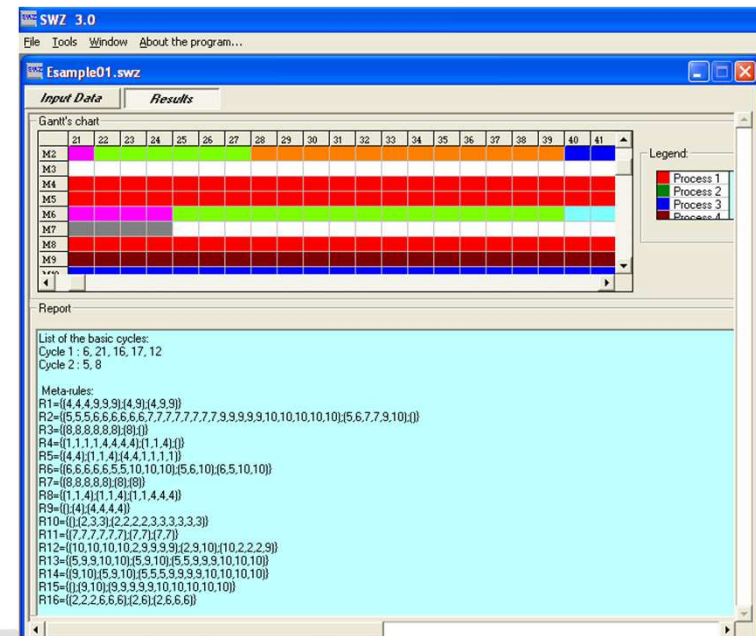
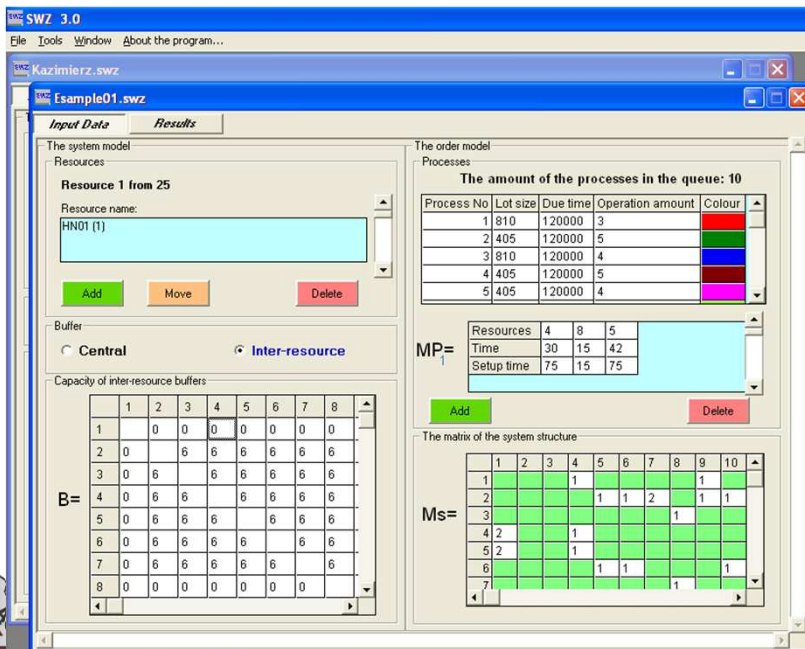
Metody wykorzystywane w celu znalezienia i zdefiniowanie relacji występujących pomiędzy źródłowym a wynikowym modelem danych oraz automatycznej realizacji integracji różnych systemów informacyjnych. Najczęściej wykorzystywanymi narzędziami w procesie mapowania są języki XSLT, SQL, Java C++.



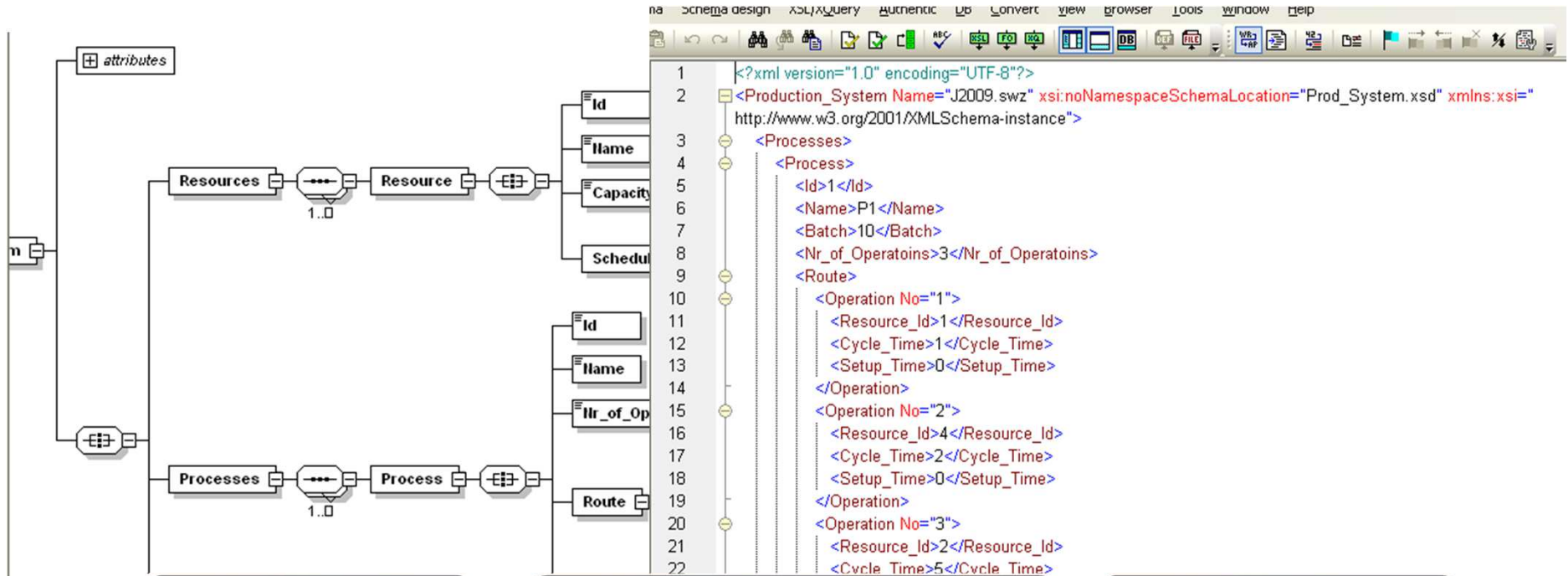
# SWZ

SWZ to komputerowy system weryfikacji zleceń produkcyjnych, wspomagający w obszarze planowania produkcji procesy podejmowania decyzji o przyjęciu bądź odrzuceniu do realizacji zleceń produkcyjnych w systemach jednoczesnej wieloasortymentowej produkcji rytmicznej. Program SWZ umożliwia wyznaczanie procedur sterujących pracą systemu w postaci lokalnych reguł rozstrzygnięcia konfliktów zasobowych dla przebiegu ustalonego oraz reguł dla fazy rozbiegu (rozruch) i wybiegu (wygaszanie) produkcji.

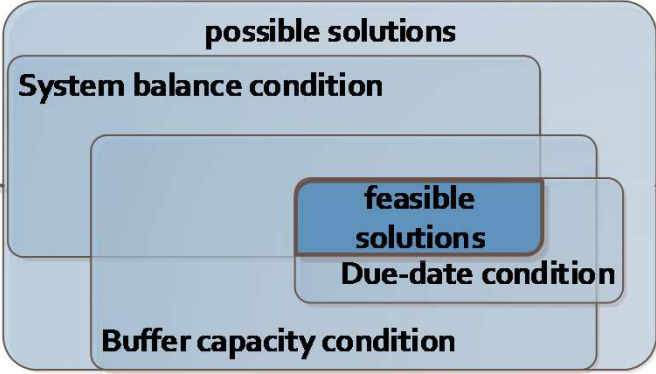
Wyznaczone reguły mają zagwarantować jakościowo dopuszczalną realizację produkcji (bez blokad i zagłódzeń) oraz terminową realizację zleceń.



# SWZ XML Schema



Cyclic concurrent processes system specification: (production resources, structure of the production processes, production processes matrix).

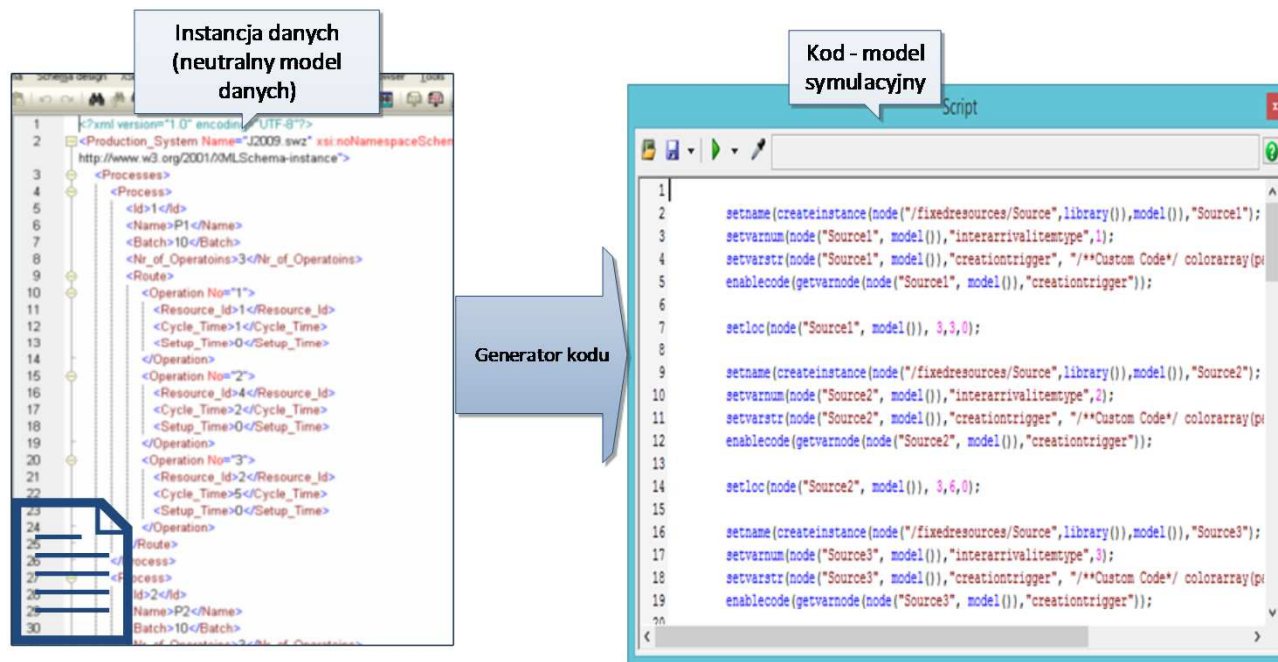


Production system parameters:  
 - dispatching rules,  
 - the cycle time,  
 - number of critical resources,  
 - coefficient of resource utilisation,  
 - capacity of storage system.



# Transformacja danych do kodu

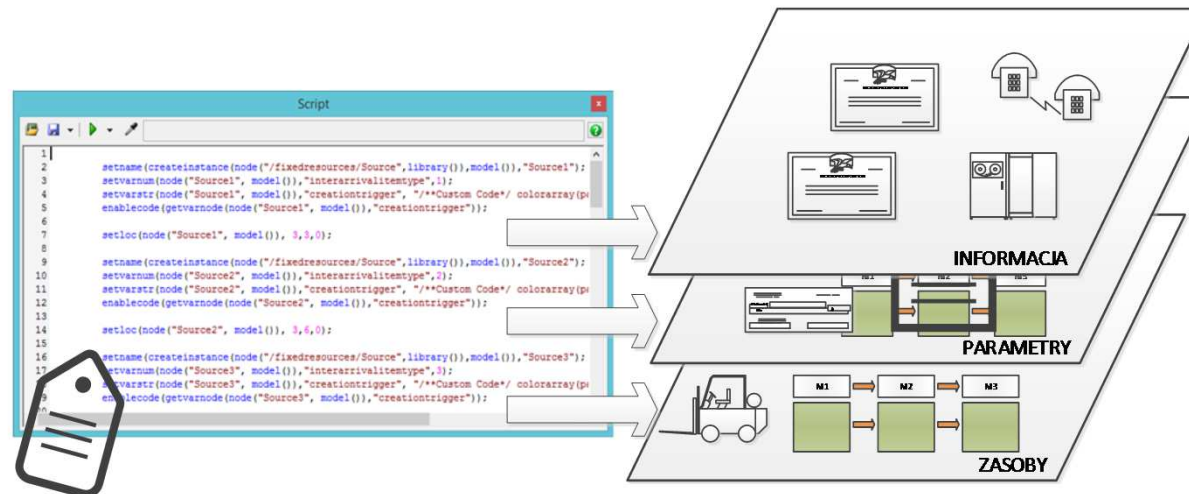
Automatyczne generowanie kodu w wewnętrznym języku programowania systemów symulacyjnych, które zawierają instrukcje tworzące kompletny model symulacyjny, są realizowane z wykorzystaniem metod transformacji danych. Jako informacje wejściowe w tym procesie wykorzystywane są jedynie instancje pośredniego neutralnego modelu danych.



# Transformacja danych do kodu

Na potrzeby opracowywanej metody wyróżniono następujące części kodu tworzącego poszczególne warstwy modelu symulacyjnego:

- warstwa zasobów systemu wytwórczego – zawiera linie kodu, tworzące zasoby składające się na system wytwórczy, tj.: maszyny, magazyny międzyoperacyjne,
- warstwa parametrów zasobów systemu – zawiera linie kodu, które parametryzują wcześniej zdefiniowane zasoby, np.: funkcje ustalające czasy realizacji operacji technologicznych, pojemności magazynów, ...
- warstwa zasobów informacyjnych – zawiera linie kodu tworzące zasoby informacyjne, tj.: tabele zawierające dane o czasach realizacji operacji na zasobach, tabele zawierające harmonogram pracy zasobów, kalendarze, ...



## Transformacja danych do kodu

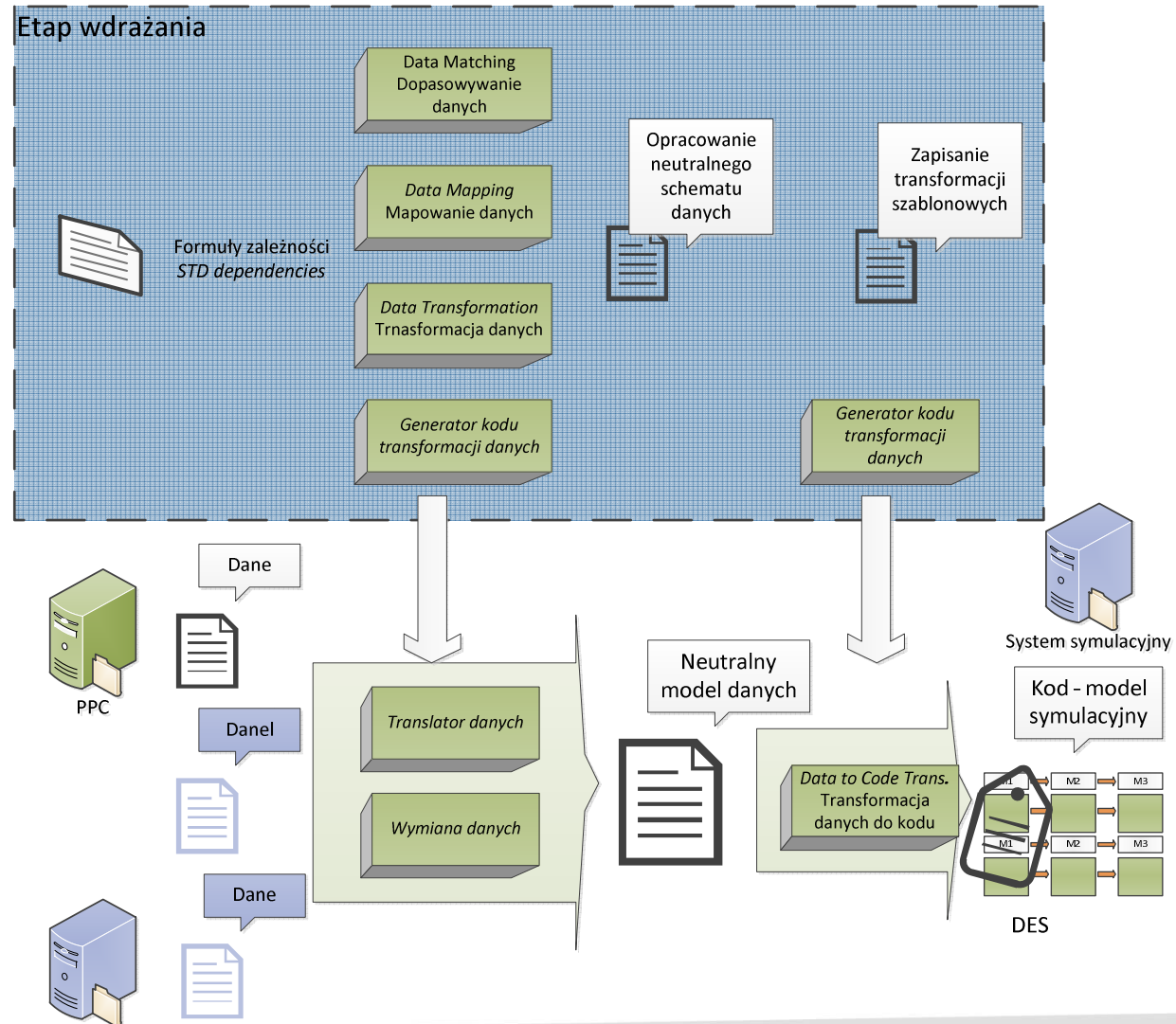
Dla przygotowanych transformacji szablonowych może następnie zostać wygenerowany kod wynikowy w wybranym przez użytkownika języku transformacji danych: XSLT 1.0/2.0.

```
T10S <[setname(createinstance(node("/fixedresources/Processor",library()), model()),"
M#1 "); setloc(node(" M#1 ", model()), (_(4+7*(#1 – round((#1-1)/#2))*#2)_ ,
_(round((#1-1)/#2+1)*3)_ ,0); Id, rzad, 1>
```

```
<xsl:variable name="rzad,, select="round(count(/Production_System/Resources/
Resource) div 2)"/>
<xsl:for-each select="Production_System/Resources/Resource">
    setname(createinstance(node("/fixedresources/Processor",library()),model()),
"M<xsl:value-of select="Id"/>");
    setloc(node("M<xsl:value-of select="Id"/>", model()), <xsl:value-of select="4 +
7 * ((Id - 1) mod $rzad +1)"/>,<xsl:value-of select="(ceiling(Id div $rzad)) * 3"/>,0);
</xsl:for-each>
```



# Integracja systemu SWZ z systemami symulacyjnymi



# Implementacja metodyki integracji systemów planowania z systemami symulacyjnymi

Implementacja metodyki wymagała określenia sposobów transformacji danych dla konkretnych, wybranych systemów planowania i symulacji produkcji. Określenia praktycznych rozwiązań w postaci definicji struktur danych, definicji schematów transformacji oraz narzędzi symulacyjnych bazujących na zaproponowanych metodach:

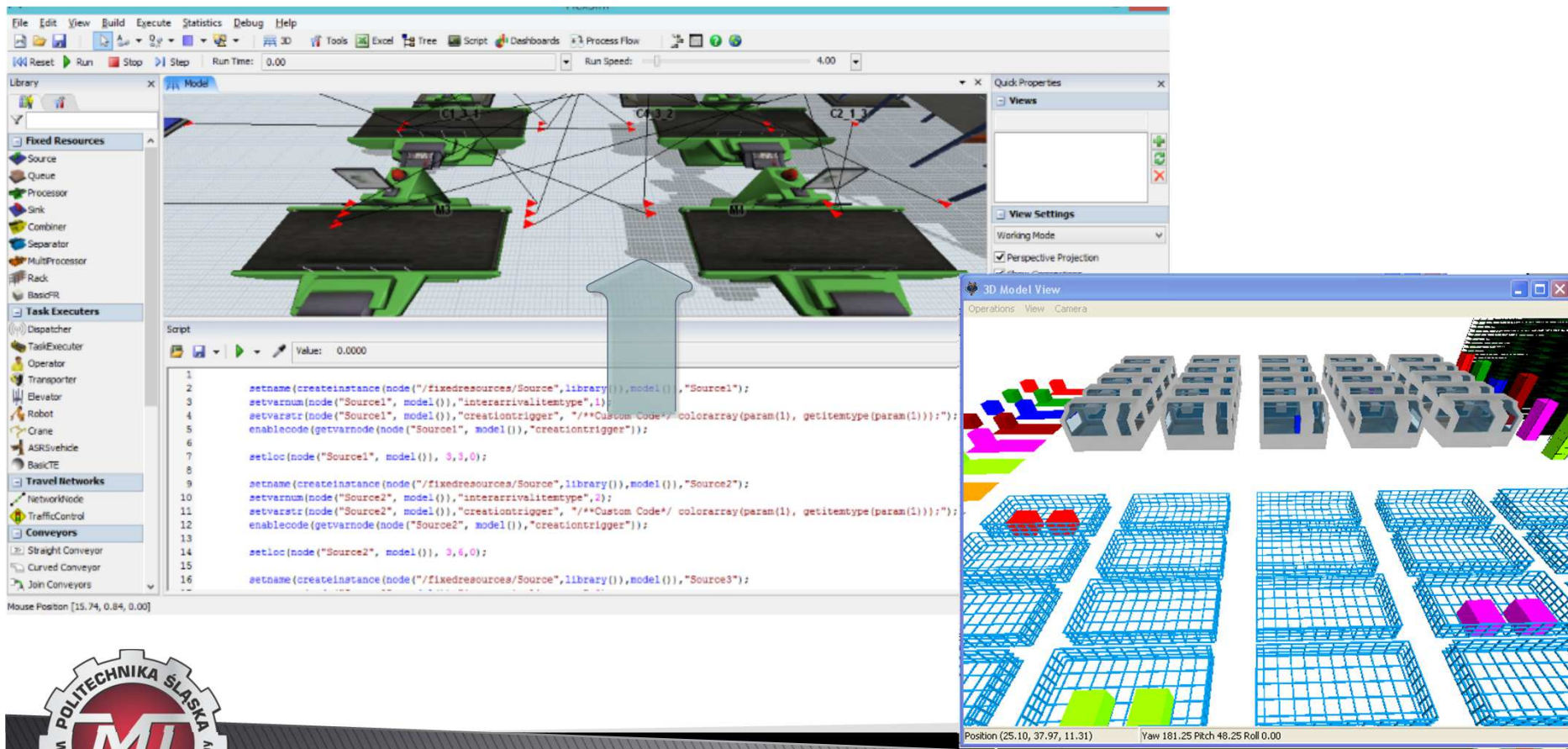
- Definicje struktur danych (SWZ) - Rozszerzalny język znaczników XML:
  - definiowanie struktury dokumentu XML przeprowadzono z wykorzystaniem standardu XML Schema (XSD),
  - możliwość zapisu informacji w sposób sformalizowany,
  - możliwość zapisywania informacji bez względu na ich typ,
  - duża elastyczność zapisu – ponieważ nie istnieje predefiniowany zbiór znaczników, nie ma on ograniczenia semantyki,
  - prostota i czytelność zapisu danych – umożliwiającą łatwe zrozumienie zapisanych informacji i tworzenie dokumentów XML dzięki stosowaniu dowolnych znaczników opisujących dane,
  - niezależność od platformy sprzętowej i programowej, co daje możliwość wymiany dokumentów pomiędzy heterogenicznymi systemami komputerowymi,
  - jest przystosowany do obsługi w sieciach internetowych i intranetowych przedsiębiorstw.





# System symulacyjny – ED, FlexSim

Na potrzeby implementacji dla wybranego systemu symulacyjnego opracowano skrypty w języku FlexScript oraz 4DScript, tworzące obiekty składowe modelu symulacyjnego dla poszczególnych warstw.

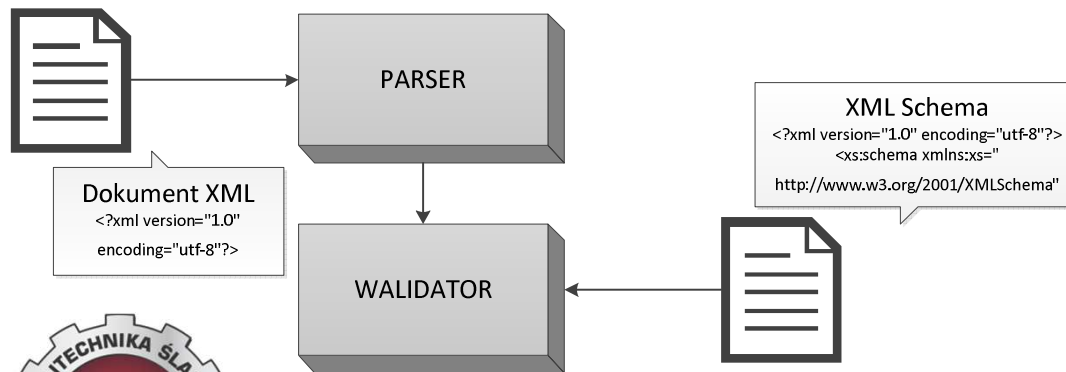


# Pośredni model danych

Opracowano definicje danych w postaci schematu XML (XSD). Struktura dokumentu definicji została podzielona na trzy części:

- zasoby – lista zasobów produkcyjnych składających się na system wytwórczy wraz z harmonogramem ich pracy,
- produkcja – dane o procesach produkcyjnych dotyczące zleceń oraz powiązanych z nimi operacji,
- magazyny – dane dotyczące typu i pojemności magazynów produkcji w toku.

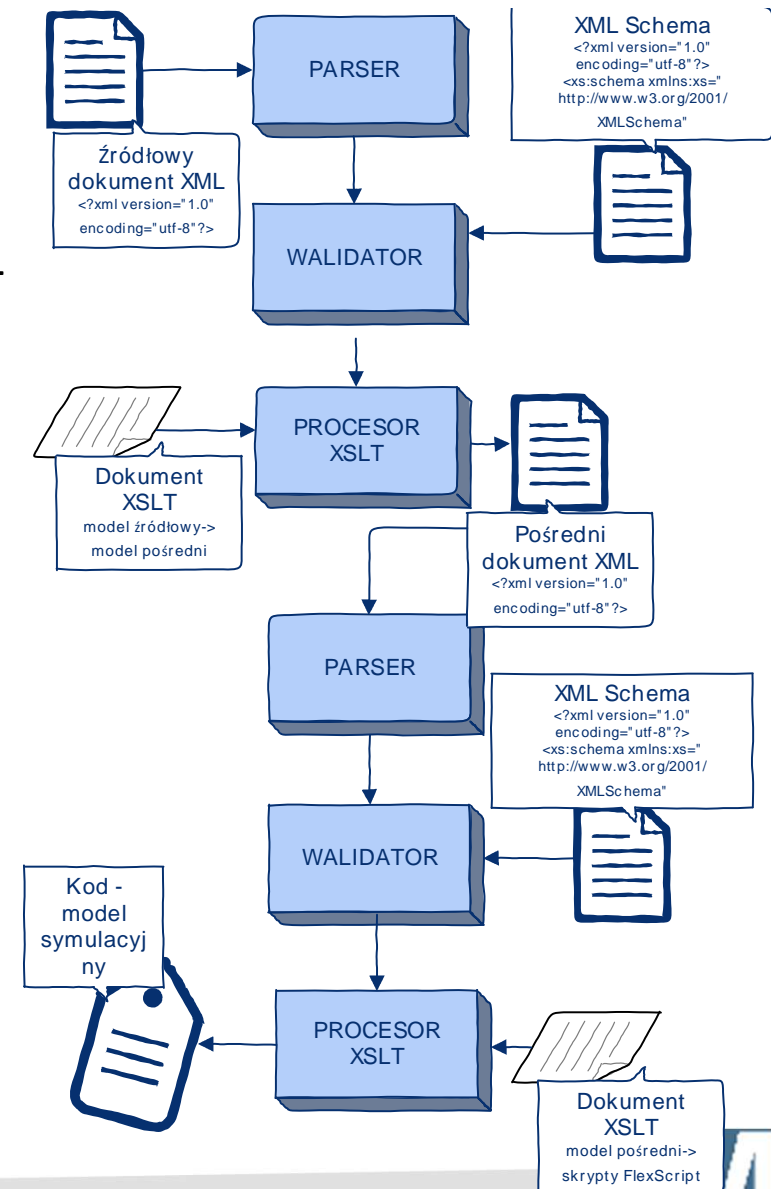
```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
version="1.00" id="Production_System_Data">
<xs:element name="Production_System">
<xs:complexType>
<xs:all>
<xs:element name="Resources">
<xs:complexType>
<xs:sequence maxOccurs="unbounded">
<xs:element name="Resource">
<xs:complexType>
<xs:all>
<xs:element name="Id" type="xs:integer"/>
<xs:element name="Name" type="xs:string"/>
<xs:element name="Capacity" type="xs:decimal"/>
<xs:element name="Schedule">
<xs:complexType>
<xs:sequence maxOccurs="unbounded">
<xs:element name="Sequence">
<xs:complexType>
<xs:all>
<xs:element name="Id" type="xs:positiveInteger"/>
<xs:element name="Process_Id"/>
</xs:all>
</xs:complexType>
</xs:element>
</xs:sequence>
```





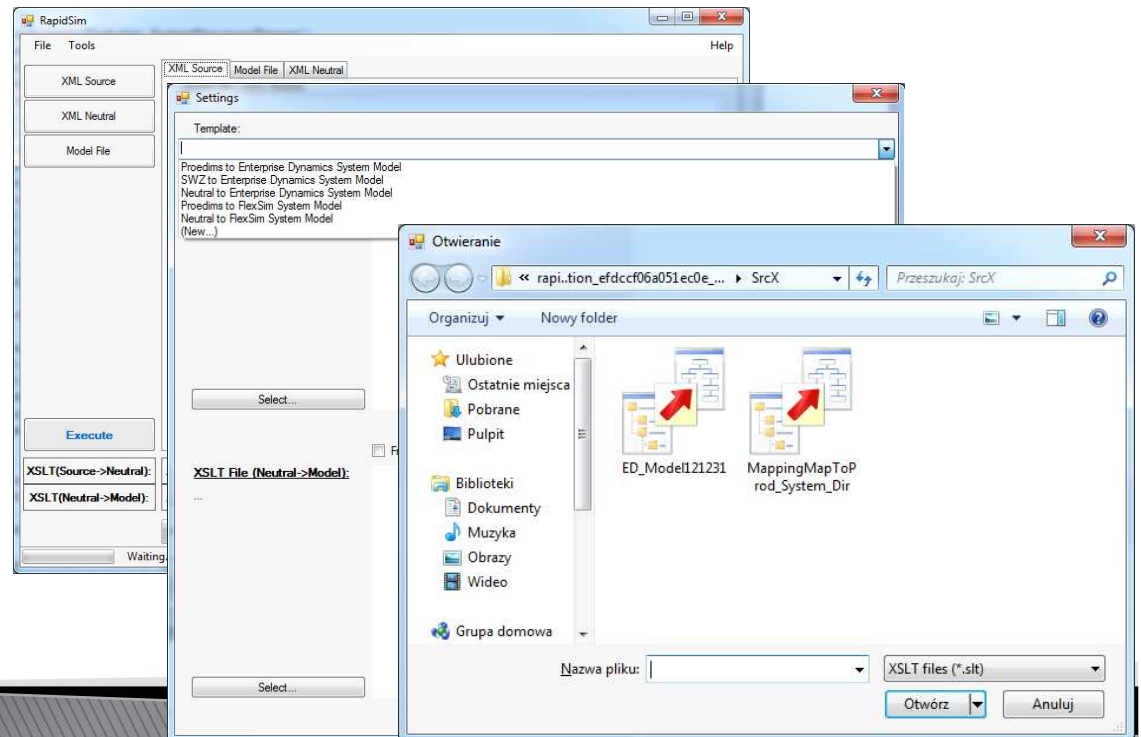
# Mapowanie i transformacja danych

- wczytanie dokumentu (dokumentów) XML z systemów PPC,
- walidacja wczytanego dokumentu XML,
- transformacja dokumentu (dokumentów) XML w procesorze XSLT pomiędzy danymi źródłowymi a modelem pośrednim danych,
- walidacja pośredniego dokumentu XML,
- transformacja dokumentu XML w procesorze XSLT pomiędzy danymi z modelu pośredniego a kodem skryptów FlexScript,
- wygenerowanie dokumentu (MOD) zawierającego skrypty FlexScript tworzące model symulacyjny systemu FlexSim,
- walidacja wygenerowanego dokumentu,
- utworzenie gotowego modelu symulacyjnego w systemie FlexSim przez wczytanie wygenerowanego dokumentu

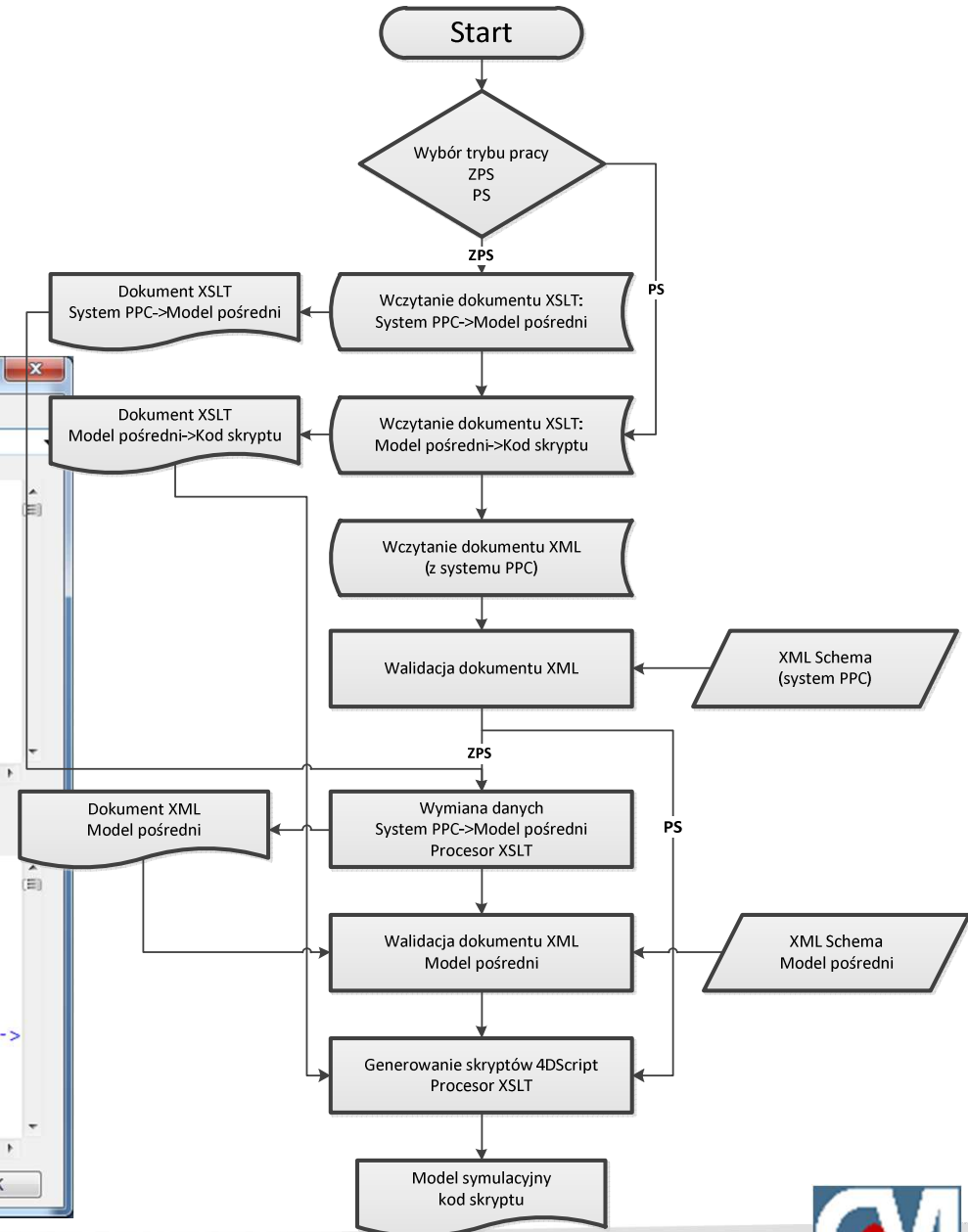
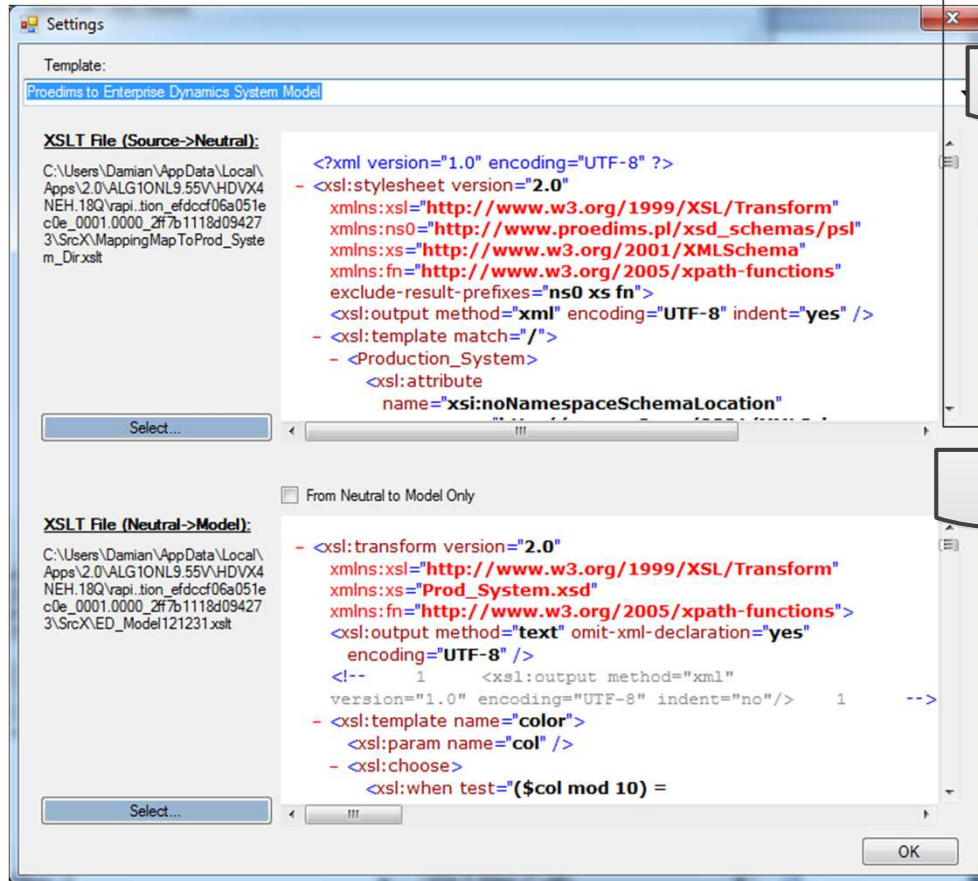


# Oprogramowanie RapidSim

- Implementacja w środowisku programowym:
  - graficzny interfejs użytkownika,
  - moduł walidacji i parsowania dokumentów XML (na podstawie definicji struktury danych XML Schema),
  - procesor XSLT,
  - generator kodu skryptów modelu symulacyjnego,
  - moduł tworzenia i edycji dokumentów XML oraz XSLT,
  - moduł automatyzacji parametrów wejściowych.



# Oprogramowanie RapidSim



# Przykład praktyczny

The screenshot shows the SWZ 3.0 software interface with the following sections:

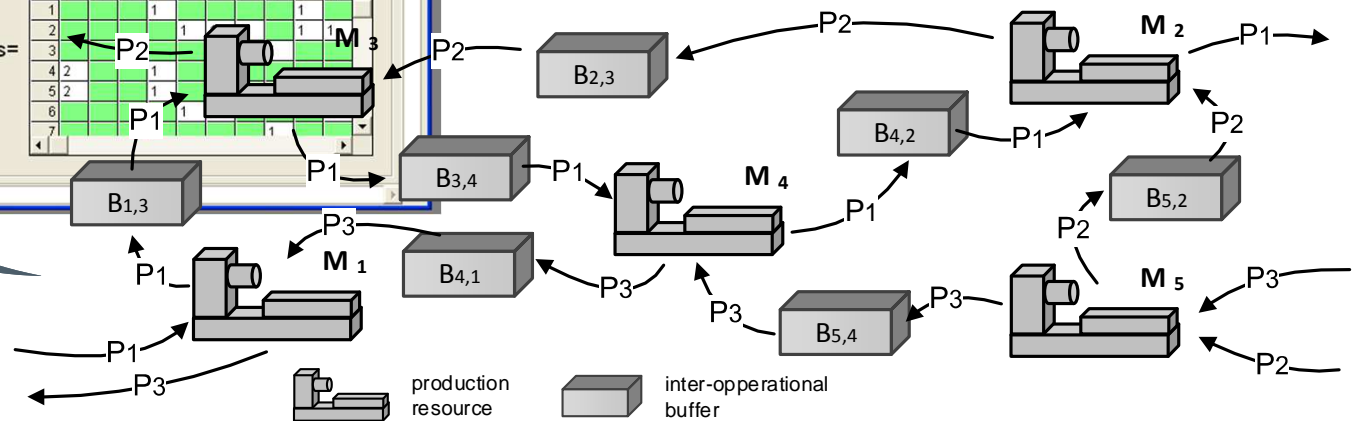
- Input Data:**
  - Resources:** Resource 1 from 25, Name: HNOT (1). Buttons: Add, Move, Delete.
  - Buffer:** Central (selected), Inter-resource.
  - Capacity of inter-resource buffers:** A grid labeled B= with values for 8 resources.
- Results:**
  - The order model:** The amount of the processes in the queue: 10. Table below:

| Process No | Lot size | Due time | Operation amount | Colour  |
|------------|----------|----------|------------------|---------|
| 1          | 810      | 120000   | 3                | Red     |
| 2          | 405      | 120000   | 5                | Green   |
| 3          | 810      | 120000   | 4                | Blue    |
| 4          | 405      | 120000   | 5                | Magenta |
| 5          | 405      | 120000   | 4                | Cyan    |

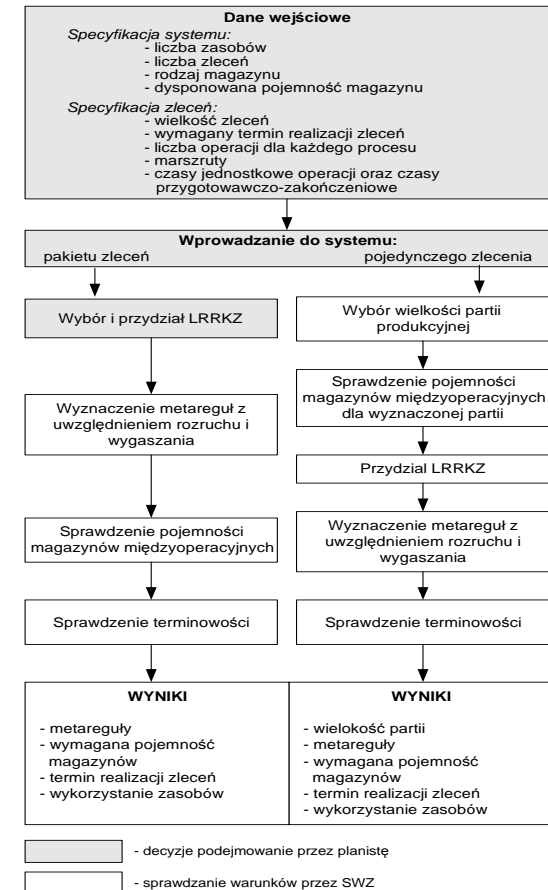
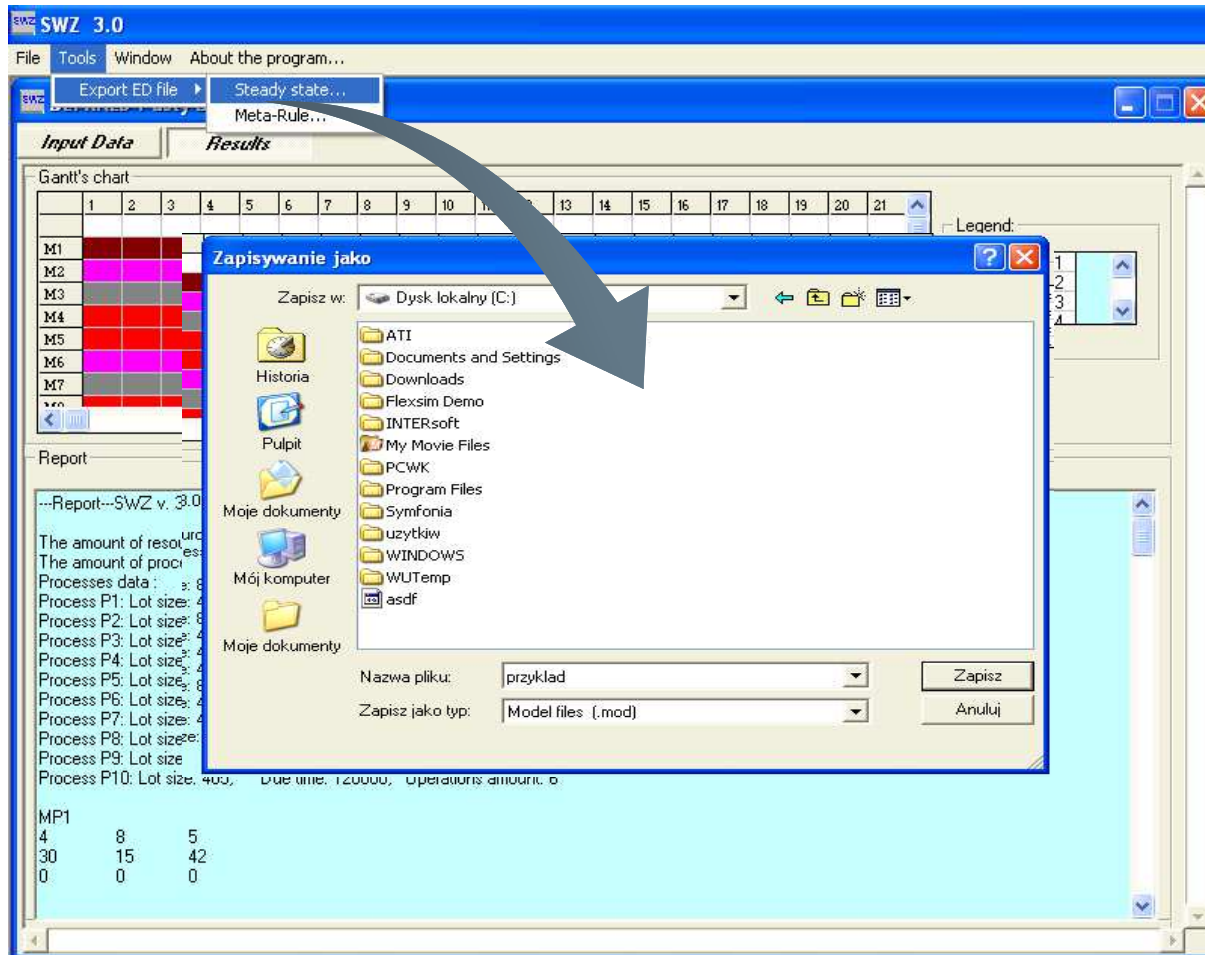
  - MP=** Table below:

| Resources  | 4  | 8  | 5  |
|------------|----|----|----|
| Time       | 30 | 15 | 42 |
| Setup time | 75 | 15 | 75 |

  - The matrix of the system structure:** A grid labeled Ms= with values for 10 processes.



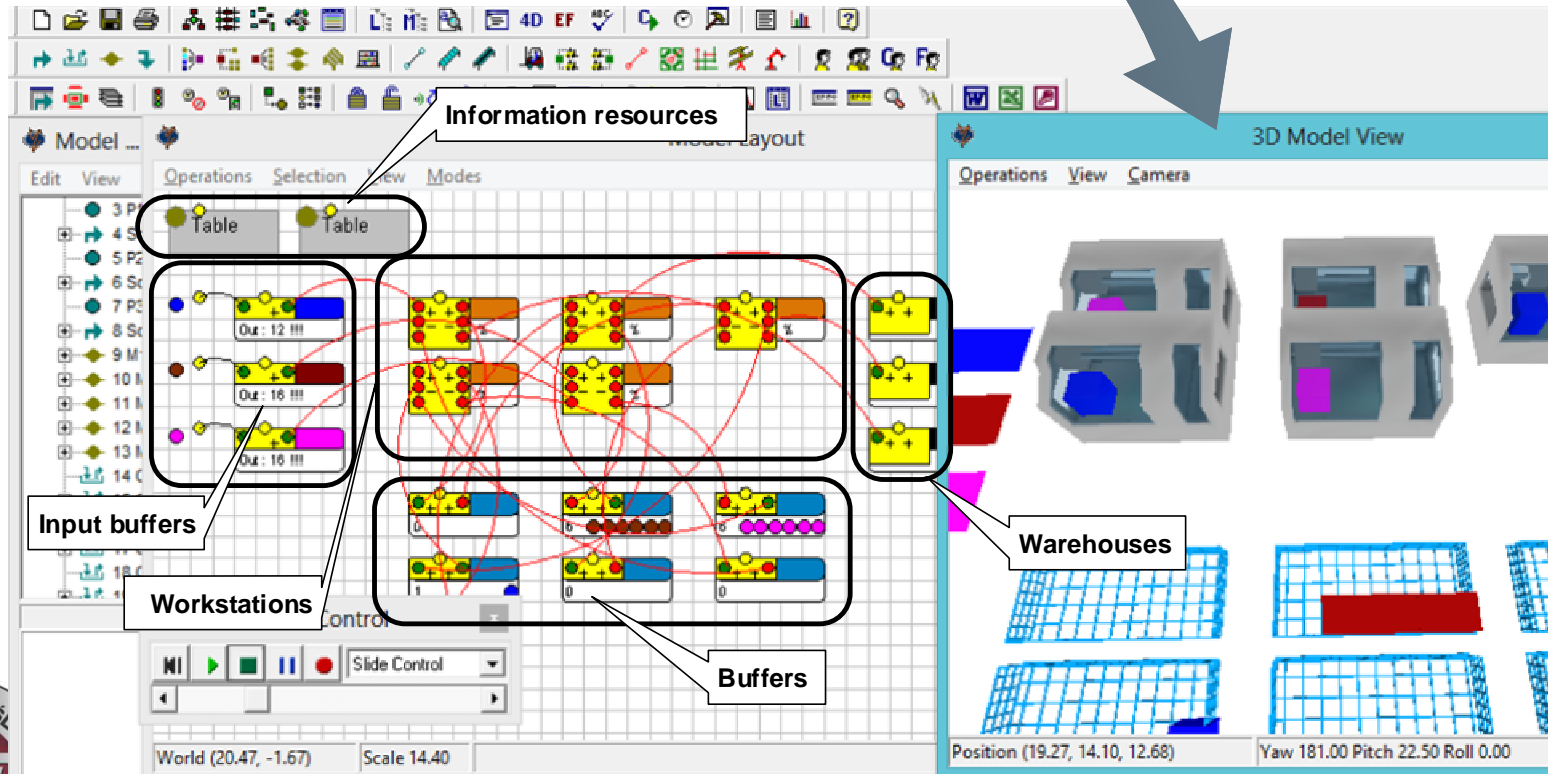
# Przykład praktyczny





# Przykład praktyczny

```
1  
Sets(CreateAtomAtomByName([Seiver], Library), Model, [M2]),  
setloc(18,3,0),  
SetChannels (3,3,s),  
SetExpAtt(1,[czaszy(Value(StripString(Name(First(c)),[P])),Value(StripString(Name(c),[M]))],s),  
SetExpAtt(2,[Value(StripString(Name(First(c)),[P]))],s),  
SetExpAtt(18,[openic(lrrkz(Value(StripString(Name(c),[M])),+(mod(input(c),lrrkz(Value(StripString(Name(c),[M])),299)),1)),c]],  
OnReset(s):=[Do(Inherit,CloseAllc(c),InStrategy)],
```



# Przykład praktyczny

The screenshot displays the RapidSim software interface, which is used for simulating production processes. The main window is divided into several panels:

- XML Source:** Contains the XML configuration for the simulation. The code includes instructions for creating atoms, setting their properties (color, icon, location), and connecting them to form a production line.
- Model Layout:** A 2D diagram showing the layout of the production line. It consists of several vertical columns of yellow rectangular blocks representing machines or workstations, connected by green lines representing the flow of materials or information.
- 3D Model View:** A 3D perspective view of the production line. The machines are rendered as grey rectangular blocks with blue and yellow details. The view is currently in a wireframe mode, showing the underlying structure of the 3D model.
- Run Control:** A control panel for running the simulation, featuring a play button, a stop button, and a slider for controlling the simulation speed.
- Execute:** A button to execute the simulation.
- XSLT:** A section for XSLT transformations, showing the paths for converting XML Source to XML Neutral and vice versa.

The interface also includes a clock, a progress bar, and various status indicators. The 3D view shows the current position and orientation of the camera, as well as the scale of the model.



Damian Krenczyk

Politechnika Śląska  
Instytut Automatyki Procesów Technologicznych  
i Zintegrowanych Systemów Wytwarzania

# Przykład praktyczny

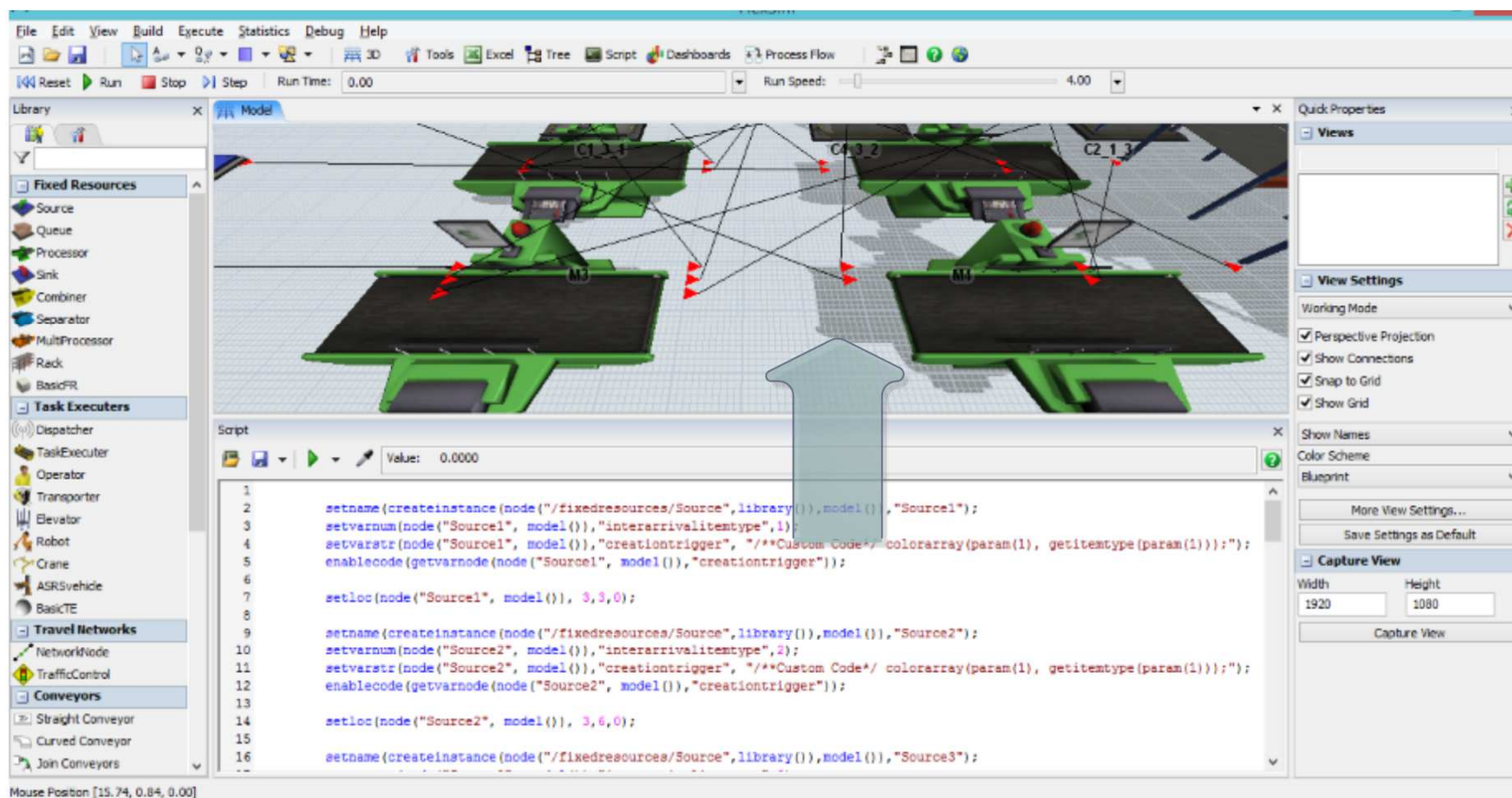
The screenshot displays the RapidSim software interface. The main window shows a 3D perspective view of a production line with components labeled Source1, Source2, Source3, M1, M2, M3, C1 2 1, C1 2 2, C1 3 1, C1 3 2, and C2 1 3. The interface includes a menu bar (File, Tools, Help), a toolbar, and a script editor on the left. The script editor contains the following code:

```
setname(createinstance(n  
setvamum(node("Source1",  
setvarstr(node("Source1",  
getitemtype(param(1))):  
enablecode(getvamode(n  
setloc(node("Source1", m  
setname(createinstance(n  
setvamum(node("Source2",  
setvarstr(node("Source2",  
getitemtype(param(1))):  
enablecode(getvamode(n  
setloc(node("Source2", m  
setname(createinstance(n  
setvamum(node("Source3",  
setvarstr(node("Source3",  
getitemtype(param(1))):  
enablecode(getvamode(n  
setloc(node("Source3", m  
setname(createinstance(n  
setvamum(node("Source3",  
setvarstr(node("Source3",  
getitemtype(param(1))):  
enablecode(getvamode(n  
setloc(node("Source3", m
```

The right side of the interface features a 'FlexSim' window with a 'Library' pane containing various components like Fixed Resources, Task Executors, Travel Networks, and AGV. A 'Quick Properties' pane is also visible on the right, showing 'View Settings' and 'Capture View' options.



## Przykład praktyczny



# Wnioski

- Metodyka umożliwiła bezpośrednio wygenerowanie modeli symulacyjnych systemów produkcyjnych dla danych pozyskiwanych z systemów PPC, niezależnie od struktury systemu produkcyjnego, topologii przepływu procesów w systemie produkcyjnym oraz ilości zasobów i zleceń produkcyjnych.
- Zgodne z koncepcją „Data-driven modelling”, umożliwia wykorzystanie narzędzi symulacyjnych przez planistów w celu wspomagania procesu podejmowania decyzji bez wymaganej wiedzy o sposobie tworzenia modeli symulacyjnych i pozwala na rezygnację z czaso- i pracochłonnego procesu budowy modelu symulacyjnego oraz jego parametryzacji w zależności od wygenerowanego planu produkcji.
- Wykonane oprogramowanie RapidSim jest narzędziem w uniwersalnym, które może być używane z dowolnym systemem PPC lub DES. Wymaga jedynie zapisania odpowiednich formuł transformacji, definicji wykorzystywanych modeli danych oraz zapisu obiektów składowych modelu symulacyjnego.

