

# Dobór oprogramowania do modelowania i symulacji procesów wytwarzania

Ryszard Zdanowicz\*

Ogromne zapotrzebowanie na programy symulacyjne powoduje ich ciągły rozwój, ale także coraz większą różnorodność oprogramowania i zwiększanie dostępnych na rynku systemów symulacyjnych. Jest to bardzo pozytywne zjawisko, lecz niestety także „niebezpieczne” dla użytkownika, który w trakcie poszukiwania odpowiedniego systemu może się w tym gąszczu zagubić. Powstaje pytanie, który z dostępnych programów jest dla danego zakładu najlepszy, aby poniesione nakłady zostały należycie wykorzystane. Artykuł jest próbą odpowiedzi na to pytanie?

Coraz większe wymagania jakości produkcji oraz obniżania kosztów wprowadzania nowych technologii wymuszają stosowanie przez przedsiębiorstwa coraz bardziej zaawansowanych narzędzi pozwalających na wirtualne planowanie produkcji. Symulacja jest techniką służącą do rozwiązywania problemów, polegającą na użyciu odpowiednich modeli wirtualnych. Model symulacyjny jest logiczno-matematycznym przedstawieniem pojęcia, systemu lub działań, zaprogramowanym w celu rozwiązania za pomocą komputera. System definiujemy jako zbiór obiektów powiązanych określonymi wzajemnymi zależnościami lub oddziaływaniami. Modelowanie symulacyjne jest jedną z ważniejszych technik wspomagających projektowanie zrobotyzowanych elastycznych systemów produkcyjnych, jak i zarządzanie produkcją. Wiąże się to z tym, że w warunkach gospodarki rynkowej przedsiębiorstwa muszą rozwiązywać złożone problemy w coraz to krótszym czasie. Na drodze do sukcesów ekonomicznych decydującą rolę odgrywa zdolność do wdrożenia konkurencyjnej koncepcji. Dla urzeczywistnienia tak silnie ukierunkowanej na odbiorcę produkcji muszą istnieć odpowiednie narzędzia do projektowania i modyfikacji systemów produkcyjnych. Narzędziami tymi mogą być układy modelowe, które umożliwiają prowadzenie badań pozwalających na szeroką ocenę jakościową i ilościową zjawisk występujących w systemach. Można powiedzieć, że modelowanie symulacyjne jest jedną z metod przyczyniających się do skrócenia czasu pojawienia się produktów na rynku.

Możliwe jest prowadzenie badań na rzeczywistym systemie produkcyjnym, poprzez poddawanie go nowym warunkom. Nie zawsze jest to uzasadnione ze względów ekonomicznych. W przypadku złożonych elastycznych systemów produkcyjnych modyfikacje wyposażenia technologicznego i projektowanie algorytmów sterowania wygodniej jest przeprowadzić na podstawie modeli symulacyjnych.

Do niedawna symulacja komputerowa była domeną dużych przedsiębiorstw, które mogły sobie pozwolić na zakup zarówno sprzętu komputerowego, jak i drogiego oprogramowania. Burzliwy rozwój technik informatycznych spowodował, że stały się one bardziej dostępne. Nastąpił też znaczący rozwój oprogramowania do modelowania i symulacji. Pakiety symulacyjne stały się bardziej kompleksowe i uniwersalne. Obecnie na rynku są dostępne różne programy symulacyjne od najprostszych stworzonych na bazie matematycznych modeli do najbardziej rozbudowanych, z aparatem stochastycznym umożliwiającym np. dopasowywanie danych wejściowych do właściwego rozkładu czy symulowanie awarii robota lub obrabiarki własnymi rozbudowanymi bibliotekami, ze środowiskiem do tworzenia animacji, grafiki 3D, bogatymi narzędziami do prezentacji danych wyjściowych z symulacji. Są one zróżnicowane cenowo od kilkudziesięciu do kilkuset tysięcy euro. Wybór narzędzia do eksperymentu symulacyjnego jest bardzo istotny i zależy od wielu czynników: doświadczenia modelującego, specyfiki problemu, wymaganego poziomu dokładności itp.

## Oprogramowanie symulacyjne

Rzeczywisty rozwój modelowania i symulacji przebiegał równolegle z ewolucją kolejnych generacji komputerów. Podstawowe elementy modelowania i symulacji komputerowej pozostały niezmiennione, np. w każdym procesie symulacji występują obliczenia matematyczne, schematy logiczne, elementy deterministyczne lub losowe albo elementy obu tych rodzajów. Do nowych zastosowań wykorzystano nowe techniki, opracowano nowe bardziej dogodne języki symulacyjne, jednakże podstawowe cechy pozostawały niezmiennione [1, 7]. W okresie powstawania aplikacji symulacyjnych większość modeli symulacyjnych była tworzona za pomocą podstawowych języków programowania lub języków symulacyjnych ogólnego przeznaczenia. Pierwsza symulacja komputerowa miała miejsce w latach 50. XX wieku. W latach 60. powstały pierwsze profesjonalne języki programowania do symulacji: GPSS, SIMSCRIPT, SIMPAC, CLS, ESP,

\* dr inż. Ryszard Zdanowicz  
– Instytut Automatyzacji Procesów  
Technologicznych i Zintegrowanych Systemów  
Wytwarzania, Politechnika Śląska

SIMON, SIMULA, ESP, MONTECODE, GPS [3, 6, 8], opracowane dla komputera IBM 7090. Były one wykorzystywane jako narzędzia symulacji – większość miała na celu eliminację wielu żmudnych czynności i uproszczenie działań przygotowawczych związanych z realizacją symulacji komputerowej. Burzliwy rozwój technologii komputerowej spowodował powstanie w 1979 roku wizualno-interaktywnego języka programowania symulacyjnego VIS (*Visual Interactive Simulation*). Dalszy rozwój – w latach 80. i 90. – komputeryzacji, systemów operacyjnych, zwłaszcza systemu Windows zaowocował powstaniem wielu programów symulacyjnych. Programy te miały rozbudowany aparat do wizualizacji procesów, w tym zdolność do generowania obrazów 3D, lepszą kompatybilność z innymi pakietami oprogramowania takimi jak: arkusze kalkulacyjne czy bazy danych, symulacja przez sieć, rozbudowane własne biblioteki czy narzędzia symulacyjne. Przykładami tego typu oprogramowania są: Enterprise Dynamics, Arena, ShowFlow czy ProModel.

Do głównych zalet symulacji wizualno-interaktywnej można zaliczyć:

- lepsze zrozumienie modelu
- łatwiejsza weryfikacja i walidacja modelu
- możliwość interaktywnego eksperymentowania
- ulepszony sposób prezentacji i interpretacji wyników
- udoskonalona komunikacja z modelem
- przystosowanie do potencjalnego wykorzystania w grupach rozwiązujących problemy.

Wymagane cechy współczesnych programów symulacyjnych to:

- generowanie losowych liczb z jednorodnego rozkładu prawdopodobieństwa
- generowanie losowych wartości ze sprecyzowanego rozkładu prawdopodobieństwa
- postępujący czas symulacji
- określenie kolejnych rezultatów oraz bieżąca kontrola kodu symulacji
- możliwość dodawania oraz usuwania rekordów
- gromadzenie i analiza danych
- raportowanie wyników
- wykrywanie błędów otoczenia.

Generalnie oprogramowania do symulacji i modelowania można podzielić na trzy zasadnicze grupy:

- arkusze kalkulacyjne
- języki oprogramowania ogólnego przeznaczenia
- oprogramowania specjalistyczne.

## Arkusze kalkulacyjne

Pakiety arkuszy kalkulacyjnych takie jak Excel, zapewniają podstawowe możliwości symulacyjne. Stworzenie prostego modelu przy użyciu dostępnych funkcji arkusza kalkulacyjnego jest stosunkowo proste. W Excelu można generować liczby losowe. Niektóre funkcje Excela umożliwiają realizację statystycznych funkcji rozkładu i modelowania postępu czasu [14]. Mimo

prostego poziomu funkcji arkuszy kalkulacyjnych w Excelu istnieje możliwość wykorzystania makr i aplikacji VBA. Model obiektowy Excela oferuje kilka bardzo wydajnych obiektów analizy danych, takich jak: arkusze, tabele przestawne, scenariusze oraz sporą liczbę funkcji matematycznych, finansowych, inżynierskich i powszechnie stosowanych w biznesie. Narzędzia te sprawiają, że Excel nadaje się do wykonywania analiz statystycznych. Język VBA umożliwia używanie obiektów oraz projektowanie zautomatyzowanych procedur. Ma on także wszystkie elementy nowoczesnych języków programowania, takie jak tablice, pętle itp. Jednak stworzenie animowanych modeli przy użyciu tych funkcji jest stosunkowo trudne.

Zaprezentowanie wyników zarówno w formie graficznej, jak i w postaci tabel daje możliwość szybkiej analizy problemu. Arkusze kalkulacyjne są lepsze od języków programowania po względem szybkości budowania modeli i wygody użycia, ale przegrywają pod tymi względami z oprogramowaniem specjalistycznym. Jediną cechą arkuszy kalkulacyjnych przewyższającą pozostałe aplikacje jest krótki czas zdobywania umiejętności ich obsługi. Należy również mieć na uwadze, że zastosowanie arkuszy kalkulacyjnych ogranicza się jedynie do symulacji systemów statycznych. Ceny arkuszy kalkulacyjnych i języków programowania kształtują się na podobnym poziomie.

## Języki programowania

W okresie powstawania pierwszych aplikacji symulacyjnych większość modeli była tworzona za pomocą podstawowych języków programowania takich jak Pascal czy Fortran. Istnieje możliwość stworzenia modeli symulacyjnych, korzystając z ogólnych języków programowania takich jak Visual Basic, C++ i Java. Użycie tych języków pozwala na duży stopień elastyczności w czasie tworzenia modeli. Jednak modelowanie i symulacja za pomocą języków programowania są bardzo czasochłonne, ze względu na ich zbyt ogólne możliwości symulacyjne.

Nowoczesny język programowania C++ oferujący programowanie obiektowe, może być przydatny w modelowaniu. Natomiast język Java pozwala na tworzenie symulacji i uruchomienie jej w sieci.

Języki programowania generalnie zapewniają największy zakres zastosowań aplikacyjnych i elastyczności modelowania. Modele stworzone za pomocą języków programowania „działają” szybciej niż modele stworzone w pozostałych aplikacjach. Jednak pod względem szybkości i wygody budowania modeli zwycięża oprogramowanie specjalistyczne [2, 4].

## Specjalistyczne oprogramowanie symulacyjne

Pakiety specjalistycznego oprogramowania do symulacji można podzielić na dwie grupy:

- pakiety symulacyjne ogólnego przeznaczenia do aplikacji różnych rodzajów

- pakiety symulacyjne dla aplikacji specjalistycznych do określonej dziedziny zastosowań, np. określenia wskaźników systemu wytwarzania, harmonogramowania produkcji.

Pakiety symulacyjne dla specjalistycznych zastosowań cechują się łatwością użycia oraz wąskim zakresem funkcji adekwatnych dla danej dziedziny.

Większość pakietów specjalistycznych można określić jako wizualno-interaktywne systemy modelowania VIS. Technika VIMS (*Visual Interaktive Modelling Systems*) pozwala na budowanie modelu symulacyjnego w sposób wizualny i interaktywny. Oprogramowanie pozwala na definiowanie zestawu obiektów symulacji. Użytkownik wybiera potrzebne obiekty i definiuje w logiczny sposób model za pomocą funkcji menu. Wizualizacja także jest definiowana za pomocą funkcji menu. W rezultacie użytkownik nie musi znać języka programowania, gdyż wszystkie parametry i ustawienia są dokonywane za pomocą funkcji menu. Istnieje oczywiście możliwość „przejścia” do kodu języka programowania danej aplikacji symulacyjnej.

Pojęcia VIS oraz VIMS nie są tożsame. Termin VIS odnosi się do natury modelowania, natomiast VIMS odnosi się do sposobu tworzenia modeli i symulacji.

Specjalistyczne oprogramowanie symulacyjne przewyższa zarówno języki programowania ogólnego przeznaczenia, jak i arkusze kalkulacyjne pod względem wygody i szybkości budowania modeli. Cena tego typu oprogramowań jest jednak zdecydowanie wyższa [9, 10].

Oferta programów symulacyjnych jest bardzo bogata. Do najbardziej znanych programów możemy zaliczyć: AutoMod, ExpertFit, MODSIM III, MAST Simulation Environment 7.0, Arena, ROBCAD, Simple++, Simul8, Quest, FactoryFlow, ShowFlow Simulation Software, Enterprise Dynamics, ReThink, IThink Analyst/Strategy 5.1, Extend V4.1, SimEngine, Witness 9.3, Micro Saint, GPSS/PC, @Risk, Powersim Enterprise Kit, ProModel, SDI Industry, Factor/Aim, Supply Chain Performance Series, OptQuest, Proof Animation, FaktorySim, StatFit, Extend+BPR v4.1, RiskOptimizer, SDI Industry, SLX. Przedstawione programy nie wyczerpują na pewno pełnej listy oprogramowania symulacyjnego.

## Dobór oprogramowania symulacyjnego

Wybór odpowiedniego oprogramowania symulacyjnego można zrealizować w pięciu następujących krokach [2, 9]:

- 1) ustalenie wymagań modelowania
- 2) przegląd oprogramowania
- 3) sporządzenie listy kryteriów oceny
- 4) ocena oprogramowania pod względem spełniania kryteriów
- 5) najlepszy wybór.

## Ustalenie wymagań modelowania

Na początku należy zdefiniować rodzaj systemu, który ma być zamodelowany. Następnie należy odpowiedzieć na pytanie, czy oprogramowanie ma być przeznaczone dla pojedynczej aplikacji czy do ogólnego użytku? Dla oprogramowania ogólnego zastosowania należy określić zakres aplikacji oraz rozległość różnorodności systemów.

Sposób przeprowadzenia modelowania jest także bardzo istotny. Należy określić, czy będzie to modelowanie szybkie i powierzchowne, czy złożone i szczegółowe. Kompleksowość realizacji modelowania wpływa, także w pośredni sposób na złożoność obsługi oprogramowania, co wiąże się z poziomem jego funkcjonalności.

Należy również wziąć pod uwagę ograniczenia wewnętrzne organizacyjne takie jak: poziom finansów przedsiębiorstwa, umiejętność obsługi oprogramowania przez pracowników oraz programowo-sprzętową taktykę przedsiębiorstwa.

## Przegląd oprogramowania

Mając zdefiniowane ogólne wymagania modelowania, można sporządzić krótką listę dostępnego oprogramowania symulacyjnego. Sporządzenie listy oprogramowania z krótkimi charakterystykami ułatwia skuteczny wybór zgodnie z ustalonymi wcześniej wymaganiami.

Informacje na temat dostępnego oprogramowania można uzyskać ze stron internetowych producentów lub sprzedawców tego typu oprogramowania. Można zasięgnąć opinii ekspertów w dziedzinie symulacji i modelowania.

Celem sporządzenia listy oprogramowania wraz z krótkimi charakterystykami jest szybkie wyeliminowanie produktów, które nie spełniają wymagań symulacyjnych danej organizacji. Po wybraniu kilku pakietów oprogramowania można przejść do ich bardziej szczegółowej analizy.

## Sporządzenie listy kryteriów

Dokonanie skutecznego porównania pakietów oprogramowania symulacyjnego wymaga stworzenia listy kryteriów oceny. Nie wszystkie przedstawione kryteria muszą zostać wykorzystane do oceny oprogramowania. Należy je dobrać pod względem potrzeb organizacji i wymagań procesu modelowania.

### Wybrane zostały kryteria:

- K1 – gwarancja i wsparcie producenta (dostępność pomocy technicznej, możliwość konsultacji, technicznych, rodzaj oferowanego szkolenia, częstotliwość aktualizacji oprogramowania, wersje obcojęzyczne, jakość dokumentacji)
- K2 – wymagania sprzętowe i programowe (współpraca z systemami operacyjnymi, platforma

- sprzętowa, oprogramowanie zabezpieczające, dostęp do Internetu/stron www)
- K3 – tworzenie i testowanie modelu (łatwość budowania modelu, występowanie opcji testującej i sprawdzającej błędy, maksymalny rozmiar modelu, maksymalny wymiar obiektów, możliwość tworzenia dokumentacji, występowanie opcji „Pomoc”, występowanie opcji „Asystent”
- K4 – wprowadzanie danych i możliwość ich analizy (możliwość dystrybucji danych, możliwość pobierania danych z empirycznych źródeł, możliwość importu danych z innych programów)
- K5 – wewnętrzny język programowania i ewentualny dostęp do kodu źródłowego

- K6 – raportowanie i analiza danych na wyjściu (możliwość sporządzania raportów dla obiektów modelu, możliwość raportów graficznych, możliwość redagowania raportów przez użytkownika, możliwość eksportowania raportów do innych programów, statystyczna analiza wyników)
- K7 – planowanie eksperymentu (prawdopodobna szybkość działania, kontrola działania, możliwości interaktywne, ilość potoków losowych, kontrola potoków, możliwość dystrybucji przez Internet)
- K8 – możliwości wizualne (czy obraz jest w czasie rzeczywistym czy odtwarzany, szybkość wyświetlania obrazu, możliwość tworzenia ikon przez użytkownika, występowanie biblioteki

Tabela 1. Porównanie programów symulacyjnych

NAZWA PROGRAMU		AutoMod	ExpertFit	Quest	FactoryFlow	Taylor	ED	IThink	Witness	@Risk	ProModel	Arena
CENA (zł)		15000	435-1475	15000	10000	12500	33000	1099	13000-17000	395	16900	995
Kryteria ogólne	Wsparcie	•	•	•	•	•	•	•	•		•	•
	Możliwość obliczania kosztów	•		•	•	•	•	•	•	•	•	•
	Współpraca z systemami operacyjnymi											
	Windows 3.x		•		•	•			•	•	•	
	Windows 98x	•	•	•	•	•	•	•	•	•	•	•
	Windows NT/2000/XP	•	•	•	•	•	•	•	•	•	•	•
	Unix/Linux			•								
	HP			•								
Macintosh							•					
Kryteria dotyczące modelowania	Graficzne budowanie modelu	•		•	•	•	•	•	•		•	•
	Programowe budowanie modelu	•		•		•	•				•	•
	Nieograniczona wielkość modelu	•		•	•	•	•			•	•	•
	Język programowania	•		•		•	•					•
	Planowanie procesu	•		•	•	•	•	•	•	•	•	•
	Planowanie dostaw	•		•	•	•	•	•	•	•	•	•
	Możliwość importu plików CAD	•		•	•	•	•		•		•	•
	Zautomatyzowana optymalizacja modelu	•		•	•	•	•		•		•	•
	Kreatory tworzenia modelu i symulacji	•		•	•	•	•	•	•	•	•	•
	Interaktywne poszukiwanie błędów	•		•	•	•	•		•		•	•
Kryteria analizy i symulacji	Planowanie eksperymentu	•	•	•		•	•		•			•
	Animacja	•		•	•	•	•	•	•		•	•
	Animacja w czasie rzeczywistym	•		•			•					•
	Symulacja ogólnego przeznaczenia		•	•		•	•	•	•	•	•	•
	Wytwarzanie	•	•	•	•	•	•	•	•	•	•	•
	Zaopatrzenie		•	•		•	•	•	•	•	•	•
	Bezpieczeństwo produkcji		•	•		•	•	•	•	•	•	•
	Gospodarka materiałowa	•	•	•	•	•	•	•	•	•	•	•
	Magazynowanie/dystrybucja	•	•	•		•	•	•	•	•	•	•

ikon, możliwość pomniejszania i powiększania obrazu)

K9 - koszt (cena sprzedaży oraz koszty: utrzymania, wsparcia technicznego, szkolenia, czas opanowania umiejętności obsługiwanego oprogramowania, możliwość uzyskania licencji użytkownika).

W tabeli 1 podano przykładowe dane dla jedenastu wybranych programów symulacyjnych [2, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. Programy te zostaną poddane ocenie wielokryterialnej.

## Ocena oprogramowania pod względem spełniania kryteriów

Każdy pakiet oprogramowania należy ocenić pod względem spełniania przez niego ustalonych kryteriów. Oceny pakietów oprogramowania można także dokonać na podstawie następujących czynników:

- rozmowy z dostawcą oprogramowania
- wersji demonstracyjnej oprogramowania
- demonstracji modeli
- rozmowy z użytkownikami tego oprogramowania
- darmowej testowej kopii oprogramowania
- dokumentacji oprogramowania
- literatury
- opinii ekspertów.

Możliwość wykorzystania tych czynników zależy od czasu dysponowanego na ocenę pakietów. Zwykle stwierdzenie czy dany pakiet oprogramowania symulacyjnego spełnia kryteria, czy nie, jest niewystarczające.

Najlepszym rozwiązaniem jest ustalenie punktacji, przykładowo w skali od 1 do 3, gdzie 1 oznacza niski stopień zgodności z kryterium, 2 - średnią zgodność analizowanego pakietu z kryterium, 3 - bardzo dobry poziom zgodności. Jednak niektóre kryteria można rozpatrywać obiektywnie, a niektóre subiektywnie. Jeżeli to możliwe, należy znaleźć obiektywne miary oceny kryteriów.

## Wybór oprogramowania

Wybór odpowiedniego pakietu oprogramowania symulacyjnego na podstawie spełniania przez niego wybranych kryteriów jest bardzo subiektywny. Aby dokonywać w miarę obiektywnych wyborów, należy zsumować punkty uzyskane przez każdy pakiet. Każde z kryteriów cechuje się różnym stopniem ważności, któremu należy przypisać odpowiednią wagę.

Ustalenie wag dla poszczególnych kryteriów może być uzależnione od wymagań przedsiębiorstwa oraz wymagań modelowania. Jeśli kryteriów jest mało, dokonuje się procentowego przydziału ważności kryteriów. Im wyższy procent tym kryterium ważniejsze. Do ustalenia całkowitej sumy punktów, jaką uzyskuje każdy analizowany pakiet oprogramowania, można posłużyć się wzorem:

$$S_i = \sum W_j E_{ji} \quad (1)$$

gdzie:  $S_i$  - całkowita punktacja  $i$ -tego pakietu,  
 $W_j$  - waga dla  $j$ -tego kryterium,  
 $E_{ji}$  - punktowa ocena  $j$ -tego kryterium dla  $i$ -tego pakietu.

Wybieramy pakiet, który uzyskał największą sumaryczną liczbę punktów. Wyboru możemy dokonać również opierając się na analizie hierarchicznej AHP (*Analytic Hierarchy Process*).

## Wybór najlepszego programu oparty na AHP

Proces analitycznej hierarchizacji jest określany jako matematyczna technika podejmowania decyzji wielokryterialnych. Metoda ta pozwala na podejmowanie decyzji w szerokim zakresie obejmującym planowanie, wybór priorytetowy, alokację zasobów, wybór najlepszej alternatywy [11, 12, 13]. AHP przedstawiona w opracowaniu została przeprowadzona za pomocą algorytmu Saaty'ego. Ze względu na przejrzystość opracowania wszelkie macierze i wyznaczniki zostały przedstawione w formie tabel.

Realizacja AHP składa się z trzech podstawowych etapów:

- porównywanie elementów decyzyjnych parami
- ocena zgodności procesu porównywania parami
- wyznaczanie (obliczanie) wag.

Saaty przyjmuje następującą skalę ważności kryteriów: 1 - porównywalne, 3 - słaba preferencja, 5 - silna preferencja, 7 - bardzo silna preferencja, 9 - maksymalna preferencja, 2, 4, 6, 8 - wartości pośrednie.

Celem działań jest wybór programu do modelowania i symulacji. Przyjęto kryteria K1 - K9. Ocenie poddane zostaną programy: AutoMod, ExpertFit, Quest, FactoryFlow, Taylor, ED, IThink, Witness, @Risk, ProModel, Arena.

## Macierz ważności kryteriów

Porównania kryteriów dokonuje się parami, korzystając z ocen punktowych. Proces porównania kryteriów opiera się na skali ważności danego kryterium, dlatego w macierzy porównawczej są wpisywane także odwrotności oceny punktowej.

Macierz  $A = [a_{ij}]$  ma na przekątnej wartość 1 oraz własność  $a_{ji} = 1/a_{ij}$ .

W każdej kolumnie macierzy  $A$  sumujemy oceny  $a_{ij}$  wg zależności:

$$k_j = \sum_{i=1}^n a_{ij} \quad (2)$$

## Wagi kryteriów

Przeprowadzamy normowanie kryteriów. W pierwszym kroku wyznaczamy wartości macierzy  $B = [b_{ij}]$ :

$$b_{ij} = \frac{a_{ij}}{k_j} \quad (3)$$

gdzie:  $k_j$  - suma ważności kryteriów z macierzy  $A$ .

W kolejnym kroku oblicza się wektor sum częściowych

$$s_i = \sum_{j=1}^n b_{ij} \quad (4)$$

oraz wagi kryteriów

$$w_i = \frac{s_i}{n} \quad (5)$$

gdzie:  $i = 1, \dots, n$ .

Macierz wag kryteriów przedstawiono w tabeli 3.

Tabela 2. Macierz ważności kryteriów

A	K1	K2	K3	K4	K5	K6	K7	K8	K9
K1	1,00	1,00	0,33	0,50	0,3	0,33	0,50	0,50	0,33
K2	1,00	1,00	0,33	0,50	0,33	0,33	0,50	0,50	0,33
K3	3,00	3,00	1,00	2,00	1,00	1,00	2,00	2,00	1,00
K4	2,00	2,00	0,50	1,00	0,50	0,50	1,00	1,00	0,50
K5	3,00	3,00	1,00	2,00	1,00	1,00	2,00	2,00	1,00
K6	3,00	3,00	1,00	2,00	1,00	1,00	2,00	2,00	1,00
K7	2,00	2,00	0,50	1,00	0,50	0,50	1,00	1,00	0,50
K8	2,00	2,00	0,50	1,00	0,50	0,50	1,00	1,00	0,50
K9	3,00	3,00	1,00	2,00	1,00	1,00	2,00	2,00	1,00
Sumy ( $k_i$ )	20,00	20,00	6,16	12,00	6,16	6,16	12,00	12,00	6,16

Tabela 3. Macierz wag kryteriów

	K1	K2	K3	K4	K5	K6	K7	K8	K9	Waga ( $w_i$ )
K1	0,0500	0,0500	0,0536	0,0417	0,0536	0,0536	0,0417	0,0417	0,0536	0,0488
K2	0,0500	0,0500	0,0536	0,0417	0,0536	0,0536	0,0417	0,0417	0,0536	0,0488
K3	0,1500	0,1500	0,1623	0,1667	0,1623	0,1623	0,1667	0,1667	0,1623	0,1610
K4	0,1000	0,1000	0,0812	0,0833	0,0812	0,0812	0,0833	0,0833	0,0812	0,0861
K6	0,1500	0,1500	0,1623	0,1667	0,1623	0,1623	0,1667	0,1667	0,1623	0,1610
K7	0,1500	0,1500	0,1623	0,1667	0,1623	0,1623	0,1667	0,1667	0,1623	0,1610
K8	0,1000	0,1000	0,0812	0,0833	0,0812	0,0812	0,0833	0,0833	0,0812	0,0861
K9	0,1000	0,1000	0,0812	0,0833	0,0812	0,0812	0,0833	0,0833	0,0812	0,0861
K10	0,1500	0,1500	0,1623	0,1667	0,1623	0,1623	0,1667	0,1667	0,1623	0,1610

Tabela 4. Współczynnik niespójności

	K1	K2	K3	K4	K5	K6	K7	K8	K9	Suma	$\lambda_i$
K1	0,0488	0,0488	0,0531	0,0430	0,0531	0,0531	0,0430	0,0430	0,0531	0,4393	9,0004
K2	0,0488	0,0488	0,0531	0,0430	0,0531	0,0531	0,0430	0,0430	0,0531	0,4393	9,0004
K3	0,1464	0,1464	0,1610	0,1722	0,1610	0,1610	0,1722	0,1722	0,1610	1,4535	9,0255
K4	0,0976	0,0976	0,0805	0,0861	0,0805	0,0805	0,0861	0,0861	0,0805	0,7755	9,0101
K5	0,1464	0,1464	0,1610	0,1722	0,1610	0,1610	0,1722	0,1722	0,1610	1,4535	9,0255
K6	0,1464	0,1464	0,1610	0,1722	0,1610	0,1610	0,1722	0,1722	0,1610	1,4535	9,0255
K7	0,0976	0,0976	0,0805	0,0861	0,0805	0,0805	0,0861	0,0861	0,0805	0,7755	9,0101
K8	0,0976	0,0976	0,0805	0,0861	0,0805	0,0805	0,0861	0,0861	0,0805	0,7755	9,0101
K9	0,1464	0,1464	0,1610	0,1722	0,1610	0,1610	0,1722	0,1722	0,1610	1,4535	9,0255

## Analiza zgodności wag

Następnym krokiem w AHP jest sprawdzenie, czy otrzymane wyniki są wiarygodne. Przy ocenie spójności ocen jest spełniony wymóg przechodności: jeżeli element  $i$  jest preferowany względem  $j$ , a element  $j$  względem  $k$ , to element  $i$  jest preferowany względem  $k$ .

W pierwszym kroku należy obliczyć tzw. współczynnik spójności  $\lambda_{sr}$ .

$$\lambda_{sr} = \frac{1}{n} \sum_{i=1}^n \lambda_i \quad (6)$$

gdzie:

$$\lambda_i = \frac{\sum_{j=1}^n A_{ij} * w_j}{w_i} \quad (7)$$

Wartości współczynników spójności zestawiono w tabeli 4.

Z zależności (6) wyznaczono  $\lambda_{sr} = 9,0148$ .

Współczynnik rozbieżności CI wyznaczamy z zależności:

$$CI = \frac{|n - \lambda_{sr}|}{n - 1} \quad (8)$$

Wyznaczony z (8)  $CI = 0,002$ .

Kolejnym krokiem jest obliczenie wskaźnika zgodności CR.

$$CR = \frac{CI}{R} \quad (9)$$

Współczynnik losowych zgodności R dobieramy z tabeli 5.

Dla macierzy 9 x 9 przyjęto R = 1,45.

CR wyznaczony z (9) wynosi 0,0013.

Warunek spójności określony zależnością (10) jest spełniony.

$$CR \leq 0,1 \quad (10)$$

### Ranking wariantów według kryteriów

Ranking wariantów polega na subiektywnej ocenie danego programu symulacyjnego i porównaniu go z innymi produktami.

Po przeprowadzeniu rankingu wariantów wykonujemy kolejne normowanie według wzoru:

$$d_{ij} = \frac{c_{ij}}{k_j} \quad (11)$$

gdzie:  $c_{ij}$  - wartości oceny produktu,  $d_{ij}$  - suma wartości ocen w danej kolumnie,  $k_j$  - suma ważności kryteriów z macierzy A.

W kolejnym kroku obliczamy wektor sum częściowych  $S_i$  oraz wagi kryteriów wk:

$$S_i = \sum_{j=1}^n d_{ij} \quad (12)$$

$$wk_i = \frac{S_i}{n} \quad (13)$$

gdzie:  $i = 1, \dots, n$ .

Tabela 5. Współczynnik losowych zgodności [28]

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49	1,51	1,48	1,56	1,57	1,59

Tabela 6. Wyniki oceny programów

	K1	K2	K3	K4	K5	K6	K7	K8	K9	Ranga suma (Ri)
P1	0,0024	0,0015	0,0323	0,0102	0,0311	0,0244	0,0160	0,0218	0,0084	0,1481
P2	0,0024	0,0078	0,0031	0,0025	0,0078	0,0061	0,0058	0,0019	0,0439	0,0814
P3	0,0024	0,0194	0,0323	0,0102	0,0311	0,0244	0,0160	0,0218	0,0115	0,1691
P4	0,0091	0,0050	0,0091	0,0102	0,0078	0,0061	0,0030	0,0061	0,0115	0,0679
P5	0,0091	0,0050	0,0323	0,0102	0,0311	0,0244	0,0160	0,0061	0,0115	0,1457
P6	0,0091	0,0014	0,0323	0,0102	0,0311	0,0244	0,0160	0,0218	0,0038	0,1501
P7	0,0091	0,0057	0,0091	0,0025	0,0078	0,0061	0,0030	0,0061	0,0440	0,0935
P8	0,0091	0,0057	0,0091	0,0102	0,0078	0,0244	0,0160	0,0061	0,0115	0,0999
P9	0,0008	0,0057	0,0031	0,0102	0,0078	0,0061	0,0030	0,0019	0,0440	0,0826
P10	0,0091	0,0057	0,0323	0,0102	0,0078	0,0244	0,0030	0,0061	0,0036	0,1021
P11	0,0024	0,0015	0,0098	0,0102	0,0311	0,0244	0,0160	0,0219	0,0409	0,1581

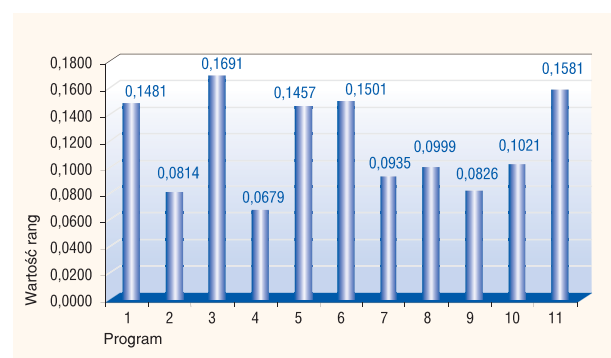
W kolejnym kroku przeprowadzono eliminacje niespójności. Jeżeli wartość CR jest mniejsza lub równa 0,1, to należy uznać, że można mówić o zgodności, natomiast gdy CR jest większy od 0,1, wówczas należy powtórzyć analizę ocen.

Powyższy ranking przeprowadzamy dla wszystkich dziewięciu kryteriów.

### Ocena oprogramowania

Wartość rangi przyporządkowana odpowiedniemu wariantowi według danego kryterium jest równa iloczynowi wagi danego kryterium oraz ocenie danego wariantu według danego kryterium.

Wyniki oceny programów zestawiono w tabeli 6.



Rys. 1. Wartości rang dla programów

Wyniki oceny programów w postaci graficznej pokazano na rys. 1. Liczby na osi poziomej odpowiadają odpowiednio programom: 1 - AutoMod, 2 - ExpertFit, 3 - Quest, 4 - FactoryFlow, 5 - Taylor, 6 - ED, 7 - IThink, 8 - Witness, 9 - @Risk, 10 - ProModel, 11 - Arena.

## Podsumowanie

Wzrastająca konkurencja będąca rezultatem wykorzystania automatyzacji produkcji zwiększającej produktywność, spowodowała zwiększenie złożoności systemów produkcyjnych, które można analizować tylko za pomocą symulacji.

Rozpowszechnienie komputeryzacji na światową skalę oraz obniżenie cen oprogramowania i sprzętu, przyczyniło się do rozwoju oprogramowania symulacyjnego.

Można powiedzieć, że symulacja jest pomocną technologią używaną coraz częściej do zwiększenia wydajności systemu poprzez dostarczenie informacji pozwalających podjąć najlepsze decyzje.

Rozwój technik multimedialnych pozwalających na animację systemów produkcyjnych, przyczynił się do lepszego zrozumienia teorii symulacji przez menedżerów i osoby niezwiązane z symulacją i modelowaniem.

Większość modeli symulacyjnych jest przedstawiana w formie wizualno-interaktywnych symulacji (VIS), oferujących użytkownikowi animację obrazu i możliwość oddziaływania na działający model. Można wyróżnić trzy główne typy oprogramowania dla symulacji, tj.: arkusze kalkulacyjne, języki programowania oraz specjalistyczne oprogramowanie symulacyjne. Wybór odpowiedniego rodzaju oprogramowania zależy od natury przeprowadzonych studiów danej dziedziny.

Większość specjalistycznego oprogramowania symulacyjnego może być opisana jako systemy wizualno-interaktywnego modelowania, które pozwalają użytkownikowi na ich obsługę za pomocą zestawu zdefiniowanych funkcji menu. Większość pakietów ma dodatkowo interfejsy programowania, co rozszerza ich możliwości i zakres modelowania.

Przedstawione przykłady pakietów symulacyjnych stanowią reprezentację różnych kategorii oprogramowania symulacyjnego. Zdolność programów do modelowania systemów produkcyjnych zależy od poziomu szczegółowości analizowanego systemu.

Niektórych dostępnych pakietów symulacyjnych można użyć do modelowania systemów o zróżnicowanym poziomie szczegółów. Problem tego typu modelowania jest ściśle uzależniony od metodologii samego modelowania. Modelowanie systemów o zmiennym poziomie szczegółowości zależy także od sposobu klasyfikacji danych do modelowania.

Proces wyboru odpowiedniego pakietu oprogramowania specjalistycznego jest uzależniony od wymagań modelowania, liczby dostępnych pakietów oraz kryteriów oceny zdefiniowanych przez przedsiębiorstwo. Dla ułatwienia wyboru można posłużyć się metodą punktowo-wagową oceny pakietów symulacyjnych lub skorzystać z AHP, będącej matematyczną techniką podejmowania decyzji wielokryterialnych. ■

## Bibliografia

1. G. Gordon, *Symulacja systemów*, WNT Warszawa, 1974.
2. C. Harrell, K. Biman, R. Bowden, *Simulation Using Pro-Model*, Mc Graw Hill, 2004.
3. V. Hlupic, R.J. Paul, *A Critical Evaluation of Four Manufacturing Simulators*. International Journal of Production Research. 1995, vol. 33 (10), pp. 2757 - 2766.
4. V. Hlupic, R.J. Paul, *A Methodological Approach to Manufacturing Simulation Software Selection*. Computer Integrated Manufacturing Systems. 1996, vol. 9 (1), pp. 49 - 55.
5. M.E. Johnson, J. P. Poorte, *A Hierarchical Approach to Computer Animation in Simulation Modeling*. Simulation. 1998, vol. 50 (January), pp. 30 - 36.
6. A.M. Law, W.D. Kelton, *Simulation Modelling and Analysis*, 3rd edn. 2000, New York, McGrawHill.
7. F.F. Martin, *Wstęp do modelowania cyfrowego*. PWN, Warszawa 1976.
8. J. Nikoukaran, V. Hlupic, R.J. Paul, *A hierarchical framework for evaluating simulation software*. Simulation Practice and Theory, 1999, 7(3), 219 - 231.
9. S. Robinson, *Simulation, The Practice of Model Development and Use*. John Wiley & Sons, Ltd, England 2004.
10. S. Robinson, *Soft with a hard centre: discrete-event simulation in facilitation*. Journal of the Operational Research Society, 2001, 52(8), 905 - 915.
11. T. L. Saaty, *Multicriteria Decision Making: The Analytic Hierarchy Process*, RWS Publications, Pittsburgh, PA., 1990.
12. T. L. Saaty, *Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process*, RWS Publications, Pittsburgh, PA, 1994.
13. O. J. Toncich, *Multilevel Simulation for Advanced Manufacturing Systems*. Advanced Manufacturing Technology. 1992 vol. 7: pp. 178 - 185.
14. W.L. Winston, *Financial Models Using Simulation and Optimization: A Step-by-Step Guide with Excel and Palisade's Decisiontools Software*, Newfield, NY: Palisade Corp, 1998.
15. R. Zdanowicz, *Modelowanie i symulacja procesów wytwarzania*. Wydawnictwo Politechniki Śląskiej, Gliwice 2002.
16. <http://www.ath.bielsko.pl/~gnikiel/publik/optym.pdf>
17. <http://www.autosim.com>
18. <http://www.deneb.com>
19. <http://www.eai.com>
20. <http://www.enterprisedynamics.com>
21. <http://www.informs-cs.org/wsc98papers/007.PDF>
22. <http://www.informs-cs.org/wsc98papers/054.PDF>
23. <http://www.lionhrtpub.com/orms/orms-8-03/frsurvey.html>
24. <http://www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html>
25. <http://www.sosresearch.org/simulationeducation/ie99simsurvey.pdf>
26. <http://www.sosresearch.org/simulationeducation/simsoftware.html>
27. <http://www.topology.org/soft/sim.html>
28. [http://www.wszia.edu.pl/pracownicy/jandreasik/mop\\_2/metoda\\_ahp/](http://www.wszia.edu.pl/pracownicy/jandreasik/mop_2/metoda_ahp/)