

Bożena Mielczarek

**Modelowanie symulacyjne
w zarządzaniu
Symulacja dyskretna**



Oficyna Wydawnicza Politechniki Wrocławskiej
Wrocław 2009

Recenzent
Edward RADOSIŃSKI
Jerzy ŻYŻYŃSKI

Opracowanie redakcyjne

Maria IZBICKA

Korekta
Alina KACZAK

Projekt okładki

Dominika OSADCÓW

Wszelkie prawa zastrzeżone. Żadna część niniejszej książki, zarówno w całości, jak i we fragmentach, nie może być reprodukowana w sposób elektroniczny, fotograficzny i inny bez zgody Wydawcy i właściciela praw autorskich.

© Copyright by Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2009

OFICyna WYDAWNICZA POLITECHNIKI WROCLAWSKIEJ
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

ISBN 978-83-7493-471-8

SPIS TREŚCI

Wstęp.....	7
1. System, model i symulacja	11
1.1. Wprowadzenie do zagadnienia symulacji	11
1.2. Istota symulacji	14
1.3. Pojęcie systemu.....	16
1.4. Modele systemów	18
1.5. Cele badań symulacyjnych.....	22
1.6. Zalety i wady symulacji	23
1.7. Warianty symulacji	24
1.8. Przebieg analizy symulacyjnej	25
1.8.1. Ogólny schemat badania symulacyjnego.....	25
1.8.2. Sformułowanie problemu	26
1.8.3. Zebranie i analiza danych wejściowych	27
1.8.4. Budowa modelu matematycznego.....	28
1.8.5. Opracowanie programu komputerowego.....	30
1.8.6. Weryfikacja i walidacja modelu	30
1.8.7. Projektowanie układu eksperymentów	31
1.8.8. Analiza wyników.....	33
1.9. Programy symulacyjne.....	34
2. Zbieranie i analiza danych wejściowych.....	36
2.1. Sztuka zbierania danych.....	36
2.2. Metoda reprezentacyjna	37
2.2.1. Podstawowe schematy losowania.....	37
2.2.2. Etapy stosowania metody reprezentacyjnej	38
2.3. Wykorzystanie danych wejściowych w modelu symulacyjnym.....	44
2.4. Ciągłe i dyskretne rozkłady zmiennej losowej.....	45
2.5. Parametryzacja podstawowych rozkładów ciągłych	45
2.5.1. Rozkład jednostajny (równomierny, prostokątny).....	45
2.5.2. Rozkład wykładniczy	46
2.5.3. Rozkład normalny	47
2.5.4. Rozkład trójkątny	48
2.5.5. Inne rozkłady ciągłe	49
2.6. Parametryzacja podstawowych rozkładów dyskretnych	50
2.6.1. Rozkład dwupunktowy.....	50
2.6.2. Rozkład dyskretny jednostajny.....	51
2.6.3. Rozkład dwumianowy	52
2.6.4. Rozkład geometryczny	53
2.6.5. Rozkład Poissona	53
2.7. Rozkłady empiryczne.....	54
2.8. Dopasowywanie rozkładów teoretycznych	56
2.8.1. Niezależność obserwacji	56
2.8.2. Stawianie hipotezy odnośnie do rodziny rozkładów. Estymacja parametrów rozkładu.....	56
2.8.3. Badanie poprawności dopasowania rozkładu	57

3. Generowanie liczb i zmiennych losowych.....	61
3.1. Wprowadzenie	61
3.2. Liczby losowe	62
3.3. Tablice liczb losowych.....	65
3.4. Generowanie liczb pseudolosowych	65
3.5. Przykład działania generatora LCG	67
3.6. Generowanie zmiennych losowych.....	68
3.6.1. Generowanie ciągłych zmiennych losowych – metoda transformacji odwrotnej	68
3.6.2. Generowanie ciągłych zmiennych losowych – przykładowe formuły.....	72
3.6.3. Generowanie dyskretnych zmiennych losowych.....	73
3.7. Generowanie strumieni zgłoszeń.....	74
3.7.1. Stacjonarny rozkład Poissona.....	74
3.7.2. Niestacjonarny proces Poissona	75
4. Symulacja zdarzeń dyskretnych.....	78
4.1. Przykład wprowadzający	78
4.2. Podstawowe elementy dyskretnego modelu symulacyjnego.....	79
4.3. Operacje na zgłoszeniach.....	81
4.4. Realizacja procesu symulacji	82
4.4.1. Zegar symulacji	82
4.4.2. Zdarzenia warunkowe i bezwarunkowe, zawieszone i równoczesne.....	83
4.4.3. Kalendarz zdarzeń i obsługa procesu symulacji	84
4.5. Techniki modelowania zdarzeń dyskretnych	86
4.5.1. Metoda planowania zdarzeń	86
4.5.2. Metoda przeglądu i wyboru działań	87
4.5.3. Metoda interakcji procesów.....	88
4.5.4. Realizacja procesu symulacji wg metody kolejnych zdarzeń – przykład nr 1	89
4.5.5. Realizacja procesu symulacji wg metody kolejnych zdarzeń – przykład nr 2	91
4.5.6. Realizacja procesu symulacji wg metody przeglądu działań – przykład nr 3	96
4.6. Analiza wyników w dyskretnym modelu symulacyjnym.....	99
4.7. Narzędzia do prowadzenia symulacji dyskretniej	99
4.8. Przykładowy model dyskretny opracowany za pomocą pakietu Arena.....	103
4.8.1. Opis systemu	103
4.8.2. Założenia do modelu	104
4.8.3. Przebieg symulacji	104
4.8.4. Wyniki symulacji	105
5. Weryfikacja i walidacja modelu	106
5.1. Podstawowe definicje.....	106
5.2. Wielopłaszczyznowa weryfikacja i walidacja	107
5.3. Trójstopniowa analiza zasadności Naylora	110
5.4. Zasady procesu walidacji i weryfikacji modelu	111
5.5. Techniki walidacji i weryfikacji.....	113
5.5.1. Taksonomia technik walidacji i weryfikacji.....	113
5.5.2. Wybrane techniki walidacji i weryfikacji.....	114
6. Planowanie eksperymentów symulacyjnych i analiza wyników.....	117
6.1. Aspekty planowania eksperymentów symulacyjnych.....	117
6.2. Strategie eksperymentów symulacyjnych dla modeli stochastycznych.....	118
6.3. Metody redukcji wariancji	121
6.4. Merytoryczne projektowanie układu eksperymentów.....	127
6.5. Przykład zastosowania analizy 2^k -czynnikowej	130

6.6. Analiza statystyczna wyników	132
7. Metoda Monte Carlo	137
7.1. Wprowadzenie	137
7.2. Etapy metody Monte Carlo	139
7.3. Określenie rozkładów losowych wybranych zmiennych wejściowych	140
7.3.1. Przykłady zmiennych o charakterze dyskretnym	141
7.3.2. Przykłady zmiennych o charakterze ciągłym	142
7.3.3. Określenie dystrybuanty dla rozkładów dyskretnych	143
7.3.4. Określenie postaci funkcji gęstości i dystrybuanty dla rozkładów ciągłych	144
7.4. Wybór zmiennych wyjściowych	144
7.5. Przeprowadzenie jednego pełnego eksperymentu symulacyjnego	145
7.5.1. Generowanie liczb losowych	145
7.5.2. Generowanie wartości zmiennej losowej	145
7.5.3. Wykonanie dużej liczby powtórzeń	147
7.6. Wykonanie pełnego badania symulacyjnego	148
7.7. Analiza wyników	148
7.7.1. Ocena precyzji oszacowania	148
7.7.2. Szacowanie wartości średniej i obliczanie prawdopodobieństwa	149
7.7.3. Badanie typu <i>What-If</i>	152
7.8. Przykład modelu symulacyjnego Monte Carlo	154
7.9. Dynamiczne modele symulacyjne Monte Carlo	161
7.9.1. Modele kolejkowe w konwencji symulacji Monte Carlo	161
7.9.2. Model kolejkowy Monte Carlo – metoda przeglądu działań	162
7.9.3. Model kolejkowy Monte Carlo – metoda kolejnych zdarzeń	164
7.9.4. Model kolejkowy Monte Carlo – metoda interakcji procesów	168
8. Arena – środowisko symulacji dyskretniej	169
8.1. Praca z programem – informacje ogólne	169
8.2. Przykładowy model nr 1. Proces produkcji z jednym stanowiskiem	171
8.2.1. Opis przykładu	171
8.2.2. Blok schematu <i>Create</i>	171
8.2.3. Blok danych <i>Entity</i>	173
8.2.4. Blok schematu <i>Process</i>	174
8.2.5. Blok danych <i>Resource</i>	176
8.2.6. Blok danych <i>Queue</i>	177
8.2.7. Blok schematu <i>Dispose</i>	177
8.2.8. Łączenie bloków schematu graficznego	178
8.2.9. Definiowanie parametrów eksperymentu symulacyjnego	178
8.2.10. Uruchamianie symulacji	179
8.2.11. Odczytywanie raportów symulacji	180
8.3. Przykładowy model nr 2. Proces produkcji z trzema stanowiskami	181
8.3.1. Opis przykładu	181
8.3.2. Blok schematu <i>Decide</i>	183
8.3.3. Blok schematu <i>Record</i>	184
8.3.4. Blok schematu <i>Assign</i>	185
8.3.5. Wyniki symulacji dla modelu nr 2	187
8.3.6. Harmonogram pracy na stanowisku <i>Kontroli Jakości</i>	188
8.4. Przykładowy model nr 3. Punkt kserograficzny	190
8.4.1. Opis przykładu	190
8.4.2. Ogólne założenia	194
8.4.3. Budowa modelu, część pierwsza. Blok <i>Separate</i>	194

8.4.4. <i>Expression Builder</i>	198
8.4.5. Budowa modelu, część druga. Blok <i>Batch</i>	201
8.4.6. Uruchomienie modelu i analiza wyników	205
Pytania sprawdzające i zadania.....	207
Literatura	214

WSTĘP

Metody symulacyjne są stosowane do opisu, badania i projektowania systemów gospodarczych i społecznych od kilkudziesięciu lat. Współczesne zastosowania symulacji datują się od czasów drugiej wojny światowej, kiedy narodziło się podejście badawcze zwane metodą Monte Carlo. Umożliwiało ono matematyczne modelowanie procesów rzeczywistych zbyt złożonych, aby można było przewidzieć ich wyniki za pomocą rozwiązań analitycznych. Początkowo metoda symulacji stosowana była głównie w fizyce i matematyce, jednak obecnie przykłady zastosowania można bez najmniejszego trudu odnaleźć nie tylko w naukach fizyczno-matematycznych, ale również wszędzie pomiędzy chemią, naukami przyrodniczymi a ekonomią, zarządzaniem i praktyką finansową. To właśnie na polu zarządzania zauważyć można coraz silniejszą potrzebę wykorzystania metod symulacyjnych, które są w stanie dostarczyć decydentom narzędzi umożliwiających ocenę sytuacji obarczonych niepewnością. Główną zaletą modeli symulacyjnych jest bowiem brak ograniczeń dotyczących struktury i stopnia skomplikowania badanego systemu oraz możliwość uwzględnienia procesów stochastycznych, co pozwala na modelowanie systemów rzeczywistych o wysokim stopniu złożoności i dużym udziale czynników losowych.

Symulacja jest dość szczególnym podejściem badawczym. Jest to w rzeczywistości nie jedno podejście, ale zbiór metod i technik, które *naśladują* działanie systemu rzeczywistego, używając komputer i różnego rodzaju oprogramowanie. Ta wielość podejść identyfikowanych jako metody symulacyjne wprowadza pewien zamęt, szczególnie wśród początkujących adeptów symulacji. Pod wspólnym szyldem *symulacji* występują: symulacja dyskretna, symulacja ciągła (w tym metoda dynamiki systemów), metoda Monte Carlo (w tym symulacje statyczne w arkuszu kalkulacyjnym), gry kierownicze, symulacja jakościowa, agentowa i inne. Odmienność takich podejść, jak np. symulacja Monte Carlo i dynamika systemów sprawia, że osoby posługujące się tymi metodami mogą mieć trudności z wzajemną komunikacją.

Spoiłem łączącym treści prezentowane w tej książce jest *losowość*. To właśnie umiejętność naśladowania probabilistycznej natury zjawisk rzeczywistych zdecydowała pierwotnie o sukcesie metody symulacyjnej Monte Carlo i sprawiła, że symulacja stochastyczna znajduje obecnie zastosowanie w analizie różnorodnych problemów decyzyjnych. Książka ta przeznaczona jest dla studentów kierunków zarządzania, ale

również dla menedżerów, decydentów i wszystkich tych, którzy chcieliby dowiedzieć się, w jaki sposób – za pomocą symulacyjnego modelu komputerowego – badać zjawiska ekonomiczne, których przebieg ma charakter *losowy*. Losowość sprawia, że dokładny opis jest możliwy dopiero *post factum*, w momencie, kiedy badane zjawisko już się dokonało. W trakcie podejmowania decyzji dostępna jest jedynie wiedza o możliwych, opisanych za pomocą funkcji prawdopodobieństwa, kierunkach ewolucji zjawiska.

W książce podjęto próbę całościowego przyjrzenia się metodologii budowy i stosowania stochastycznych modeli symulacyjnych, ze szczególnym uwzględnieniem metody symulacji zdarzeniowej, zwanej również *symulacją dyskretną*. Przyjmując za niepodważalny fakt, że otaczające nas zjawiska rzeczywiste mają naturę probabilistyczną, starano się zademonstrować, jak poprawnie wpleść niepewność do modelu symulacyjnego; jak upewnić się, że losowy charakter uzyskiwanych wyników oddaje istotę systemu, którego model zbudowaliśmy, a nie wynika z błędów, które popełniliśmy w trakcie prac nad modelem; w jaki sposób pozyskiwać wyniki symulacji i w jaki sposób je interpretować, aby pogłębić wiedzę o badanym systemie, w którym występują czynniki o charakterze losowym. Albert Einstein, jeden z największych fizyków naszych czasów, nie zgadzał się co prawda z probabilistyczną interpretacją świata oferowaną przez fizykę kwantową, czemu dał wyraz zdaniem *Nigdy nie uwierzę, że Bóg gra ze światem w kości*, jednak właśnie niepewność skutków planowanych działań sprawia, że menedżerowie poszukują narzędzi umożliwiających podejmowanie racjonalnych decyzji w warunkach ryzyka.

O niesłabnącym zainteresowaniu metodami symulacyjnymi na świecie świadczą pojawiające się każdego roku nowe książki oraz kolejne wznowienia cenionych podręczników akademickich. Rynek polski jest pod tym względem dość ubogi. Brakuje tłumaczeń uznanych wydawnictw zagranicznych, a nieliczne opracowania polskich autorów nie są w stanie wypełnić istniejącej luki. Być może książka ta ułatwi czytelnikowi wglębianie się w ciekawy świat symulacji komputerowej i zachęci do sięgnięcia po inne jeszcze źródła literaturowe.

W pierwszym rozdziale omówiono ogólną ideę symulacji i procesu modelowania symulacyjnego bez odwoływania się do konkretnej techniki. Na początku uporządkowano definicje dotyczące trzech ważnych pojęć: systemu, modelu i symulacji, a następnie wyszczególniono główne cele badań symulacyjnych i scharakteryzowano poszczególne etapy analizy symulacyjnej. Przybliżono niezorientowanemu w tematyce modelowania symulacyjnego czytelnikowi pojęcie symulacji, wyjaśniono kiedy i dlaczego warto sięgnąć po rozwiązania symulacyjne, jakie korzyści i zagrożenia niosą ze sobą metody symulacyjne.

Rozdział drugi dotyczy zagadnień związanych z przygotowaniem danych wejściowych o charakterze losowym. Dane należy najpierw poprawnie pozyskać z systemu rzeczywistego; w tym celu przypomniane zostały kluczowe kroki metody reprezentacyjnej. Po zebraniu, dane są wprowadzane do modelu pod postacią rozkładów losowych. Scharakteryzowano zatem najważniejsze dla modeli symulacyjnych rozkła-

dy ciągle i dyskretne, a następnie omówiono procedurę dopasowania rozkładu teoretycznego do zebranej próbki danych.

Podjęcie decyzji o postaci rozkładu losowego wprowadzanego do modelu nie spowoduje oczywiście automatycznego uzyskania zmienności o takim charakterze. W rozdziale trzecim omówiono kolejne etapy zamiany teoretycznej formuły rozkładu na losowe wartości odpowiadające wybranym zmiennym losowym. Przypomniano kluczowe dla tego zagadnienia pojęcie liczb losowych i wyjaśniono, w jaki sposób w modelach matematycznych uzyskuje się takie liczby, a następnie omówiono metodę transformacji odwrotnej – najczęściej stosowaną technikę zamiany liczb losowych na zmienne losowe. W symulacjach stochastycznych szczególną rolę odgrywa proces losowego napływu zgłoszeń do systemu i dlatego temu zagadnieniu poświęcono ostatni punkt rozdziału trzeciego.

W rozdziale czwartym omówiono najważniejsze podejście w grupie symulacyjnych metod stochastycznych, czyli symulację dyskretną, nazywaną również symulacją zdarzeniową lub kolejkową. Przedstawiono podstawy teoretyczne metody i zaprezentowano przykładowe modele, skupiając się na najistotniejszych kwestiach metodologicznych. Metoda symulacji dyskretny została omówiona bez odwoływania się do konkretnego narzędzia, tak aby umożliwić czytelnikowi zapoznanie się z metodologicznymi aspektami modeli zdarzeniowych. Wyszczególniono kluczowe elementy dyskretnego modelu symulacyjnego i scharakteryzowano podstawowe podejścia teoretyczne wyróżniane w grupie modeli kolejkowych: przeglądu działań, planowania zdarzeń i interakcji procesów. W celu zilustrowania przebiegu symulacji zdarzeniowej przeprowadzono ręczną symulację na trzech wybranych przykładach.

W kolejnych rozdziałach powrócono do ogólnych rozważań odnoszących się do teorii symulacji: analizy zasadności modeli symulacyjnych i planowania eksperymentów symulacyjnych.

W rozdziale piątym zdefiniowano pojęcie weryfikacji, walidacji i wiarygodności modeli, zaprezentowano główne podejścia metodologiczne oraz wyszczególniono podstawowe zasady i techniki walidacji i weryfikacji modeli symulacyjnych.

W rozdziale szóstym omówiono zagadnienie planowania eksperymentów w dwóch aspektach. Po pierwsze, niezbędny jest taki sposób uruchomienia modelu, aby można było wychwycić *typowy* obraz działania systemu, a nie jednostkowe (czyli przypadkowe) jego zachowanie. Z kolei osiągnięcie merytorycznego celu badań symulacyjnych oznacza podjęcie decyzji *przed* uruchomieniem symulacji, odnośnie do wyboru takiej konfiguracji parametrów modelu, która pozwoli uzyskać odpowiedzi na postawione pytania przy najmniejszej możliwej liczbie przeprowadzonych eksperymentów.

W rozdziale siódmym przedstawiono metodę Monte Carlo, która należy do grupy stochastycznych metod symulacyjnych i jest najczęściej stosowana do badania zjawisk, w których czas nie odgrywa istotnej roli. Podejście Monte Carlo można jednak zastosować również do prowadzenia symulacji dynamicznej, w trakcie której obserwuje się zmiany systemu zachodzące w czasie i wykorzystuje się informacje zebrane podczas wcześniej symulowanych okresów. W takim ujęciu metodę symulacyjną

Monte Carlo można potraktować jako odmianę symulacji dyskretnej, akcentując istotną zaletę, jaką jest możliwość budowy modeli symulacyjnych w arkuszu kalkulacyjnym. Naturalnym środowiskiem budowy modeli Monte Carlo jest bowiem arkusz, chętnie używane narzędzie do rozwiązywania różnorodnych problemów decyzyjnych. W rozdziale tym omówiono etapy prowadzenia symulacji Monte Carlo, prezentując szczegółowo wiele przykładowych modeli. Zadbano o to, aby przywoływane były tylko standardowe funkcje programu Excel, umożliwiające prowadzenie symulacji bez konieczności instalowania dodatkowych nakładek.

W ostatnim, ósmym rozdziale zamieszczono przykładowe modele dyskretne opracowane za pomocą Areny – komputerowego środowiska graficznego do prowadzenia symulacji zdarzeniowej. Zaprezentowano podstawowe kwestie związane z budową i uruchomieniem prostych modeli symulacyjnych za pomocą Areny v.10. Rozdział ten zawiera informacje wprowadzające, które powinny ułatwić dalszą samodzielną naukę. Dydaktyczna wersja programu Arena dostępna jest jako dodatek do wielu angielskojęzycznych podręczników akademickich, a na kilku polskich uczelniach prowadzone są zajęcia z wykorzystaniem tego narzędzia. Pominięcie tego rozdziału nie pomniejszy ogólności wywodów prezentowanych w książce i Czytelnik niezainteresowany prezentowanym środowiskiem Areny może zakończyć lekturę na rozdziale siódmym.

1. SYSTEM, MODEL I SYMULACJA

1.1. WPROWADZENIE DO ZAGADNIENIA SYMULACJI

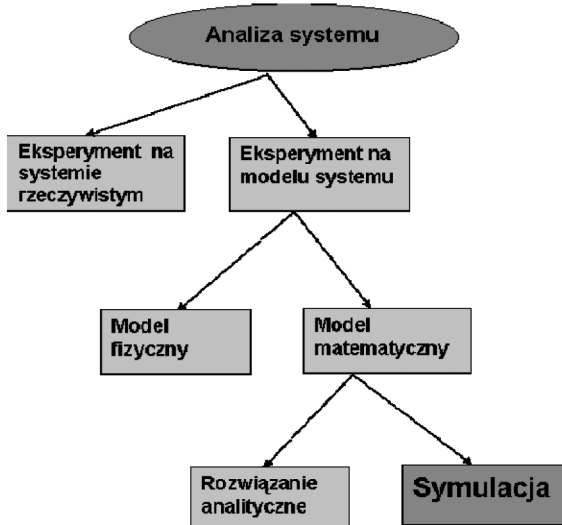
Naturalnym sposobem badania systemów rzeczywistych jest prowadzenie eksperymentów bezpośrednio na obiekcie (por. rys. 1.1). Można na przykład manipulować czasem pracy sygnalizacji świetlnej (na poszczególnych skrzyżowaniach miejskich) tak, aby doprowadzić do ich synchronizacji, zapewniającej najkrótszy czas przejazdu samochodów przez centrum miasta w różnych porach doby; kierownictwo hurtowni może oceniać skutki zastosowania różnych strategii kontroli zapasów w aspekcie płynności realizacji zamówień; dyrekcja supermarketu może badać wpływ poszczególnych wariantów harmonogramów pracy pracowników stanowisk kasowych na sprawność obsługi klientów; administrator systemu komputerowego może testować różne zasady korzystania z sieci przez użytkowników. Jeżeli możliwe jest przeprowadzenie eksperymentu bezpośrednio na systemie bez wywoływania niepożądanych skutków, to jest to z pewnością najwłaściwsza decyzja. Nie zawsze jednak możemy się na tego rodzaju próby zdecydować. W wielu przypadkach eksperyment na systemie rzeczywistym będzie niemożliwy, kosztowny lub możliwy, ale bardzo trudny do przeprowadzenia, np.:

- nie można testować alternatywnych strategii w firmie, którą dopiero zamierzamy założyć,
- nawet jeżeli firma już istnieje, zbyt kosztowne będzie np. wybudowanie nowej hali produkcyjnej po to, by przekonać się, że inwestycja taka nie poprawiła znacząco naszych możliwości produkcyjnych,
- trudno byłoby sprowadzić podwójną liczbę klientów do banku, aby przekonać się, czy oddział poradzi sobie z obsługą w przypadku zamknięcia innego, pobliskiego oddziału,
- testowanie nowych procedur celnych na lotnisku może spowodować chaos i liczne skargi podróżnych,
- eksperymentowanie z różną liczbą pracujących karettek pogotowia nie wyszłoby zapewne na zdrowie pacjentom.

W powyższych sytuacjach (i wielu innych) można zbudować model, który będzie zastępował system w eksperymentach, a następnie zadawać pytania, co mogłoby się zdarzyć, jeżeli podejmiemy taką czy inną decyzję lub wtedy, gdy wydarzy się coś, co trudno nam będzie kontrolować (poważna awaria w firmie, wypadek masowego zatrucia w mieście itp.). Badania na modelu nie przyniosą nikomu szkody, a my możemy formułować nawet bardzo zaskakujące pytania.

Możemy wyróżnić wiele typów modeli. Już w szkole podstawowej sklejamy modele figur przestrzennych, czyli budujemy modele fizyczne. Modelem fizycznym może być np.: makieta miasta, na której będziemy analizować możliwe skutki różnych rozwiązań komunikacyjnych; model restauracji w pełnej skali, gdzie prowadzone będą szkolenia pracowników i testowane różne procedury obsługi; symulator lotu przeznaczony do szkolenia pilotów; symulator pomieszczeń kontrolnych elektrowni nuklearnych.

Innym rodzajem modeli są modele matematyczne, które opisują system rzeczywisty za pomocą zbioru symboli i relacji. Model matematyczny może być przedstawiony w postaci programu komputerowego (uwaga: model matematyczny nie jest tożsamy z modelem komputerowym). Jeżeli model taki jest wiarygodny (czyli poprawnie opisuje badany system), chcielibyśmy na podstawie obserwacji modelu znaleźć odpowiedzi na pytania związane z zachowaniem się systemu. Manipulowanie modelem nie będzie miało żadnych nieoczekiwanych (czy wręcz niepożądanych) skutków dla systemu.



Rys. 1.1. Metody analizy systemów rzeczywistych

Źródło: [Law i Kelton 1991]

Po zbudowaniu wiarygodnego modelu matematycznego należy znaleźć sposób na manipulowanie i posługiwanie się nim oraz analizę jego zachowania. Można w tym celu zastosować metody analityczne, takie jak: teoria masowej obsługi, rachunek róż-

niczkowy, teoria programowania liniowego, algebra i inne. Dysponujemy wtedy jasno zdefiniowanymi formułami do udzielania odpowiedzi na zadane pytania, a co najważniejsze, odpowiedzi te są jednoznaczne i dokładne. Wiele systemów, które chcemy analizować jest jednak na tyle skomplikowana, że *modele* tych systemów są również skomplikowane, co często uniemożliwia zastosowanie metod analitycznych. Możemy oczywiście zbudować uproszczony model takiego systemu, ale najprawdopodobniej nie będzie on wiarygodny. Uzyskamy wtedy klarowne i proste odpowiedzi na źle sformułowane pytania. Alternatywą może być zastosowanie *symulacji*.

Symulacja jest jedną z technik rozwiązywania modeli matematycznych. Wywodzi się z teorii statystyki i z metod analizy złożonych stochastycznych systemów fizycznych. W obu podejściach używa się liczb i zmiennych losowych w celu oszacowania uzyskanych wyników i/lub proponowanego rozwiązania. W symulacji niezbędna jest ponadto umiejętność modelowania oraz znajomość metod analizy statystycznej i wnioskowania heurystycznego. Chociaż początki symulacji sięgają czasów, w których nie znano maszyn cyfrowych, trudno wyobrazić sobie obecnie prowadzenie badań symulacyjnych bez komputerów, stąd do nazwy *symulacja* przylgnęło na stałe określenie *komputerowa*.

Symulacja oznacza eksperyment prowadzony na modelu matematycznym systemu rzeczywistego, zapisanym w postaci programu komputerowego, w miejsce eksperymentu wykonywanego bezpośrednio na badanym obiekcie. W stwierdzeniu tym, kluczowe znaczenie dla zrozumienia istoty i natury symulacji mają słowa *model* i *eksperyment*. Podobnie jak inne techniki ilościowe wykorzystywane do wspomagania procesów decyzyjnych, symulacja wymaga zbudowania *modelu* matematycznego. Tym, co odróżnia symulację od innych podejść, jest właśnie *eksperymentowanie* na modelu poprzez wielokrotne uruchamianie go, obserwowanie działania i analizowanie wyników, które najczęściej są zbiorem wartości, a nie jedną liczbą.

Słowo symulacja wywodzi się z języka łacińskiego od słowa *similis*, mającego znaczenie: podobieństwo, podobny. Rdzeń tego słowa tkwi także w wyrazach łacińskich *similo* – podobny, *similare* – udawać, upodabniać się oraz *simulacrum* – udawać. Współczesne rozumienie pojęcia symulacji w niektórych językach ma także etymologiczne źródło w greckim słowie *mimeisthai*, znaczącym tyle, co naśladować, grać rolę oraz łacińskim *imitatio*, oznaczającym naśladowanie (np. rosyjskie *umuma-czia*). W sensie znaczeniowym słowo symulacja ma zatem także swoje znaczenia bliskoznaczne: imitacja, mimikra, mimetyzm.

Początek współczesnych zastosowań symulacji to okres II wojny światowej, kiedy to opracowano metodę, której nadano nazwę pochodzącą od znanej z hazardu dzielnicy księstwa Monako, aby podkreślić losowy charakter zjawisk, do badania których może być ona wykorzystana. Oficjalnie metoda Monte Carlo „narodziła się” w 1944 roku w trakcie badań nad teorią konstrukcji bomby atomowej. W historii jej powstania odnajdujemy polskie ślady. Wywodzący się z Lwowskiej Szkoły Matematycznej matematyk Stanisław Ulam współpracował z profesorem Johnem von Neumannem i wniósł znaczący wkład w rozwój pierwszych metod numerycznych. To właśnie fa-

scynacja Ulama procesami losowymi naprowadziła zespół kierowany przez Johna von Neumanna na pomysł wykorzystania liczb losowych do opisu możliwych trajektorii neutronów w procesie dyfuzji jądrowej. Ponieważ było to zadanie ściśle tajne, nadano mu kryptonim „Monte Carlo”, wiążąc losowość zdarzeń ze światowej sławy gniazdem hazardu. Obecnie nazwa Monte Carlo odnosi się do pewnej gałęzi eksperymentów matematycznych bazujących na liczbach losowych.

Analiza symulacyjna była stosowana początkowo głównie w sektorze produkcyjnym (planowanie, sterowanie zapasami, linie produkcyjne), ale stopniowo modele symulacyjne powstawały także w sektorach nieprodukcyjnych, np. usługowym i rządowym (analiza szpitalnego systemu przyjęć i wypisów, systemy transportowe i inne). Obszary zastosowań symulacji są bardzo rozległe i stale się powiększają. Znane przykłady wdrożeń modeli symulacyjnych obejmują systemy finansowe, transportowe, produkcyjne, dystrybucyjne, usługowe.

Modelami symulacyjnymi są również gry kierownicze, które mają niezwykle bogatą historię, sięgającą czasów prehistorycznych. O rozwoju symulacyjnych gier kierowniczych do XIX wieku można przeczytać w [Balcerak i Pelech 1999].

1.2. ISTOTA SYMULACJI

Definicji i określeń pojęcia *symulacja* jest wiele. Poniżej zamieszczono tylko kilka z nich.

Gordon [1974] zdefiniował symulację jako *technikę rozwiązywania problemów, polegającą na śledzeniu w czasie zmian zachodzących w dynamicznym modelu systemu*.

Według Naylora [1974] *symulacja to technika numeryczna służąca do dokonania eksperymentów na pewnych rodzajach modeli matematycznych, które opisują przy pomocy maszyny cyfrowej zachowanie się złożonego systemu w ciągu długiego czasu*.

Fishman [1981] stwierdził, że *symulacja to metoda badania zachowania się systemów*. Rozwijając następnie tę definicję wyjaśnił, że *przez symulację systemów rozumie czynność przedstawiania systemu za pomocą modelu symbolicznego, którym można łatwo operować i na podstawie którego otrzymujemy wyniki numeryczne*.

Dla Gajdy [2001] *symulacja oznacza wprawianie modelu w ruch*.

Evans i Olson [2002] podają, że *symulacja to proces projektowania modelu matematycznego lub logicznego systemu rzeczywistego lub problemu decyzyjnego, a następnie prowadzenie na tym modelu eksperymentów (najczęściej komputerowych) w celu uzyskania wiedzy na temat zachowań systemu rzeczywistego lub pomocy w rozwiązaniu problemu decyzyjnego*.

Autorzy jednej z najczęściej cytowanych książek na temat symulacji ([Law i Kelton 1991]) interpretują symulację jako *technikę numeryczną, która polega na prze-*

prowadzaniu eksperymentów na modelu w celu uzyskania wiedzy na temat wpływu parametrów wejściowych na wybrane mierniki wyjściowe.

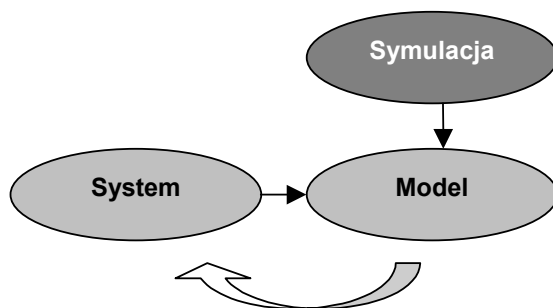
Na podstawie wprowadzonej przez siebie definicji symulacji Naylor sformułował kilka stwierdzeń, które wydają się być kluczowe dla poprawnej interpretacji tego pojęcia (por. [Naylor 1974]).

1. Symulacja jest techniką numeryczną.
2. Symulacja jest pewnym eksperymentem.
3. Komputer nie jest niezbędnym narzędziem do prowadzenia eksperymentu symulacji modelu matematycznego, na pewno jednak przyspiesza ten proces i zmniejsza ryzyko popełnienia błędu. Współcześnie symulacja prowadzona jest przeważnie za pomocą komputera.

4. Eksperyment symulacyjny prowadzony jest na *modelu systemu* rzeczywistego w *określonej chwili* lub *podczas pewnego czasu*. W pierwszym przypadku mówimy o symulacji *statycznej* lub *przekrojowej*, natomiast w drugim symulacja nosi nazwę *dynamicznej* lub symulacji *szeregów czasowych*.

5. Większość eksperymentów symulacyjnych na modelach systemów ekonomicznych stanowią symulacje stochastyczne.

Aby można było mówić o badaniu symulacyjnym, należy poczynić pewne założenia odnośnie do struktury i procesów zachodzących w *systemie*. Założenia te, przedstawiane najczęściej w formie równań matematycznych, tworzą *model* (por. rys. 1.2). Model poddawany jest analizie poprzez *eksperymenty symulacyjne*, a sformułowane wnioski przenoszone są następnie na badany system. Kluczowa, dla odróżnienia symulacji od innych technik badania systemów poprzez modelowanie, jest konieczność *operowania* modelem komputerowym, czyli jak stwierdza Gajda [2001] *wprawianie modelu w ruch*.



Rys. 1.2. Związek pomiędzy systemem, modelem i symulacją

Nie istnieje jeden uniwersalny typ modelu symulacyjnego, a symulacja nie jest jednym podejściem, co mylnie sugerować może określenie *model symulacyjny*. *Symulacja* to obszerny zbiór metod, które służą do odtwarzania (naśladowania) *systemów* rzeczywistych poprzez analizę *modeli*, zwykle za pomocą komputera i odpowiedniego oprogramowania. W wielu znakomitych książkach na temat symulacji autorzy koncentrują się na różnych technikach i na wybranych zagadnieniach. Win-

ston [1997, 2005] oraz Hillier i współautorzy [2008] zajmują się symulacją statyczną prowadzoną w arkuszu kalkulacyjnym, Pidd [2004] omawia podstawy symulacji dyskretnej, Law i Kelton [1991, 2001] szczegółowo prezentują metodologię budowy i analizy dyskretnych, stochastycznych modeli symulacyjnych. Z kolei Fishman [1981] skupia się na zagadnieniach statystycznych, które występują na poszczególnych etapach procesu symulacji.

Modelowanie symulacyjne jest natomiast paradygmatem naukowym, rozumianym jako zbiór wzorcowych mechanizmów analizy złożonych systemów rzeczywistych, które umożliwiają budowę modelu uproszczonej wersji systemu, a następnie eksperymentowanie na nim w celu poznania struktury systemu i opisanie jego zachowania.

1.3. POJĘCIE SYSTEMU

System to *pewien zbiór powiązanych ze sobą obiektów scharakteryzowanych za pomocą atrybutów (cech), które również mogą być ze sobą powiązane* [Fishman 1981]. Według Gordona [1974] natomiast, system to *zbiór obiektów powiązanych określonymi wzajemnymi zależnościami lub oddziaływaniami*.

Tabela 1.1. Przykłady systemów oraz ich obiektów i atrybutów

System	Obiekty	Atrybuty
Szpital	Pacjent	Schorzenie Operowalny/nieoperowalny Przypadek ciężki/lekki Wiek
	Lekarz	Specjalizacja
	Oddział	Typ oddziału Liczba łóżek Średni czas pobytu pacjenta z określonym typem schorzenia Liczba lekarzy
	Laboratorium	Rodzaje wykonywanych badań Średni czas obsługi Dzienna liczba badań Liczba laborantek Liczba aparatów diagnostycznych
Stacja paliw	Samochód	Typ samochodu Pojemność baku Rodzaj tankowanego paliwa Częstotliwość podjeżdżania samochodów na stację
	Dystrybutor	Typ Średni czas obsługi
	Pracownik	Harmonogram pracy

Systemy składają się z elementów (obiektów) wydzielonych pod kątem pewnych interesujących badacza własności (atrybutów). Dany obiekt może posiadać wiele atrybutów. W systemie zachodzą również pewne oddziaływania wywołujące w nim zmiany. Każdy proces wywołujący zmiany w systemie nazwiemy działaniem.

Przykładowo, pacjent i lekarz pracujący w przychodni tworzą razem pewien system (por. tab. 1.1). Atrybutem lekarza jako obiektu jest np. jego specjalizacja (internista, laryngolog), natomiast atrybutami pacjenta, traktowanego również jako obiekt, mogą być np. wiek, rodzaj schorzenia. Schorzenie pacjenta może być związane z jego wiekiem, jak również pozostaje w relacji ze specjalnością wybranego lekarza. Kiedy mówimy o wzajemnych relacjach, mamy na myśli fakt, że wybrany przez pacjenta lekarz jest *funkcją* jego schorzenia.

Każdy system charakteryzuje się trzema cechami: ma on *linie graniczne*, istnieje w pewnym *otoczeniu* oraz posiada *podsystemy*. Otoczenie jest zbiorem obszarów, w których osadzony jest system, natomiast linie graniczne wyodrębniają obiekty systemu spośród obiektów, które stanowią jego otoczenie.

Działania zachodzące wewnątrz systemu będziemy nazywać *wewnętrznymi*, natomiast działania pochodzące z otoczenia i mające wpływ na system zdefiniujemy jako działania *zewnętrzne*. System *zamknięty* to taki, dla którego nie istnieją działania zewnętrzne, natomiast system *otwarty* pozostaje w interakcji z działaniami zewnętrznymi.

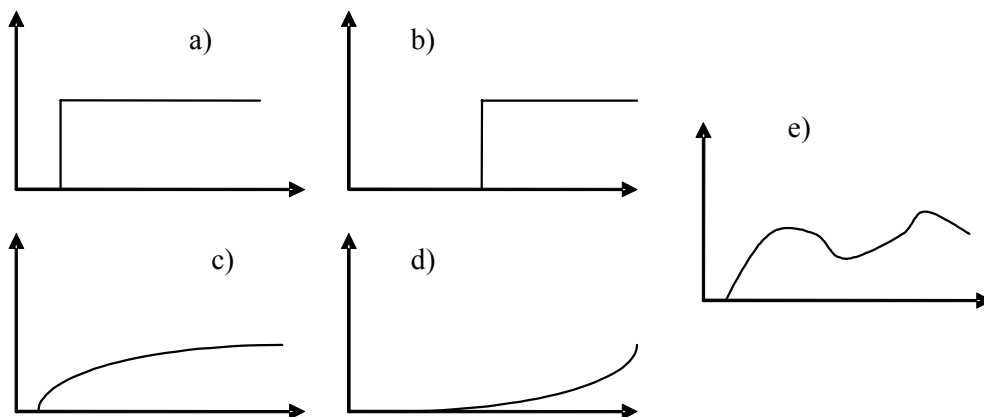
Celem prowadzenia analizy systemów jest najczęściej chęć poprawienia działania systemu. Celem idealnym jest *optymalizacja*. Niektóre jednak z elementów systemu są poza kontrolą i to one narzucają *ograniczenia* na sposób zachowania się systemu, wykluczając nieograniczoną optymalizację. Wtedy celem staje się optymalizacja działania systemu przy pewnych ograniczeniach.

Łącznie, wszystkie atrybuty obiektu definiują jego stan, a z kolei stany obiektów definiują *stan systemu*. *Stanem systemu* będziemy nazywać opis wszystkich obiektów, atrybutów i działań w danej chwili [Gordon 1974]. Badanie systemu polega na obserwacji zmian jego stanów w czasie, które w pewnym skończonym okresie tworzą jeden skończony lub nieskończony zbiór możliwych ciągów stanów. Im dłuższy jest badany okres, tym większy jest zbiór możliwych ciągów. Jeżeli zdefiniujemy małe, ustalone przedziały czasu, to będziemy mieli do czynienia z pojęciem prawdopodobieństwa natrafienia na system o określonym stanie oraz z pojęciem prawdopodobieństwa przejścia systemu do jednego z pozostałych stanów [Fishman 1981].

Zmianę stanu systemu w czasie określają dwa elementy: *wzmocnienie* i *opóźnienie*. Wzmocnienie zmiany jest równe bezwzględnej różnicy między wartością atrybutu w określonym okresie, a jego wartością przed zmianą. Jeżeli stany systemu w chwili t oraz $t + 1$ określimy odpowiednio za pomocą X_t i X_{t+1} , to $|X_{t+1} - X_t|$ oznacza wzmocnienie zmiany stanu w okresie $(t, t + 1)$.

Zmianę w stanie systemu wywołują najczęściej bodźce zewnętrzne. Po odebraniu bodźca zmiana może pojawić się albo po upływie pewnego czasu, albo w trakcie działania bodźca. Czas, jaki upływa między odebraniem bodźca a zmianą stanu, nosi na-

zwę *opóźnienia*. Łącznie charakterystykę zmiany wzmocnienia oraz opóźnienie nazywamy *reakcją systemu*.



Rys. 1.3. Typy reakcji systemu
Źródło: [Fishman 1981]

Na rysunkach 1.3 od a do e przedstawiono pięć przykładowych reakcji systemu. Rysunki a, b i c odpowiadają reakcjom *stabilnym*, ponieważ w każdym z tych przypadków stan systemu, reagując na pojedynczy bodziec, przechodzi do nowej, trwałej *równowagi* albo natychmiast, albo dopiero po upływie pewnego czasu. W procesie przechodzenia od jednego stanu równowagi do drugiego, system znajduje się w *stanie chwilowym*. Na rysunku d pokazano reakcję *niestabilną*. Na rysunku e przedstawiono natomiast system, w którym istnieje stan równowagi r , lecz reakcja systemu oscyluje wokół tego poziomu bez widocznej zbieżności do niego. System taki nazywamy *niestabilnym systemem wybuchowym* [Fishman 1981].

1.4. MODELE SYSTEMÓW

Badania symulacyjne wymagają zbudowania modelu matematycznego, który z kolei powinien być tak sformułowany, aby umożliwiać budowę wydajnego i sprawnego programu komputerowego. Użytkownik modelu powinien być w stanie zadawać właściwe pytania, posługując się dobrze zaprojektowaną procedurą eksperymentalną, aby uzyskać użyteczne wyniki.

Według Fishmana [1981] model *może być formalnym przedstawieniem teorii lub też formalnym opisem empirycznej obserwacji (bądź kombinacją powyższych)*. Gordon [1974] uważa, że model to *zbiór informacji o systemie, zebranych w celu jego zbadania*. Ponieważ jednak zbierane informacje określone są przez cel badania, nie istnieje

jedyny, właściwy model systemu. Różni analitycy systemów tworzą różne modele w zależności od rozwiązywanego problemu.

Przed wszystkim model (por. [Fishman 1981]):

- umożliwia badaczowi sprawdzenie swoich teoretycznych przekonań o systemie, dokonanie na nim empirycznych obserwacji oraz wyciągnięcie z tych poczynań logicznych wniosków;
- ułatwia zrozumienie systemu;
- skłania do prowadzenia w przyszłości badań szczegółowych;
- przyspiesza wykonanie analizy;
- określa metody testowania pożądanym modyfikacji systemu;
- ułatwia manipulowanie charakterystykami wejściowymi;
- umożliwia prowadzenie kontroli o wiele większej liczby źródeł zmienności, niż byłoby to możliwe przy bezpośrednim badaniu systemu;
- obniża koszt badań w porównaniu z badaniami prowadzonymi na rzeczywistym systemie.

W procesie budowy modelu można wyróżnić dwa szczególne etapy: formułowania struktury i zbierania danych. Struktura modelu wyznaczana jest przez granice systemu oraz jego obiekty, atrybuty i działania. Dane pozwalają natomiast zdefiniować wartości atrybutów i relacje związane z działaniami. Etapy określania struktury modelu i gromadzenia danych są ze sobą tak silnie powiązane, że należy je traktować jako części składowe jednego problemu, a nie jako dwa oddzielne problemy. Założenia dotyczące modelowanego systemu determinują rodzaj zbieranych danych, a z kolei analiza danych pozwala potwierdzić lub odrzucić przyjęte założenia. Zebrane dane często mogą ujawnić nowe relacje i powiązania, które wymuszają z kolei zmianę struktury modelu.

Ważną umiejętnością jest dostosowanie szczegółowości modelu do celu badań. Oczywiście, im więcej szczegółów zawiera model, tym bardziej upodabnia się do systemu rzeczywistego. Jednakże większa szczegółowość modelu przyczynia się do wzrostu trudności przy rozwiązywaniu problemu, jak również do wzrostu kosztów rozwiązania. W każdym modelu badacz musi, pod pewnym względem, ograniczyć szczegółowość rozważań. Zamiast uwzględniania wszystkich szczegółów, przyjmujemy pewne *założenia* o zachowaniu się systemu. Ponieważ jednak musimy być pewni, że założenia te nie są sprzeczne z obserwowanym zachowaniem się systemu – weryfikujemy je z rzeczywistością.

Nie istnieją uniwersalne zasady tworzenia modeli matematycznych, jednak warto pamiętać o pewnych regułach. Reguły te nie postulują kolejnych kroków budowy modelu, ale przedstawiają różne punkty widzenia oceny informacji, która powinna być zawarta w modelu [Gordon 1974].

- *Struktura blokowa*: system powinien być przedstawiony za pomocą zbioru bloków. Upraszcza to opis wzajemnych oddziaływań wewnątrz systemu. Każdy blok opisuje pewną część systemu i zawiera kilka zmiennych wejściowych i wyjściowych.

- *Zgodność z celem badań*: model powinien opisywać tylko te elementy systemu, które są istotne ze względu na cel badań. Wprawdzie informacje nieistotne nie mogą modelowi zaszkodzić, jednak go komplikują i zwiększają czasochłonność jego tworzenia, a następnie rozwiązywania.

- *Dokładność*: cecha ta jest istotnie związana z celem badań. Przykład: jeżeli chcielibyśmy oszacować czas składania zamówienia przez klientów w bibliotece, możemy przyjąć do dalszej analizy wyłącznie średni czas wyszukiwania pozycji w katalogach. Jeżeli jednak będziemy chcieli poznać „wydolność” komputerowego systemu katalogów, niezbędne będzie dokładniejsze szacowanie powyższego czasu.

- *Integracja*: należy wziąć pod uwagę zakres, w jakim zbiory różnych obiektów można łączyć w obiekty uogólnione. Na przykład, dyrektor sieci bibliotek nie potrzebuje modelu, w którym każde stanowisko komputerowe będzie wyodrębnione, natomiast dyrektor konkretnej biblioteki będzie już zainteresowany takim rozróżnieniem

Modele matematyczne można klasyfikować w różnych przekrojach. Oprócz przedstawionego na rys. 1.1 podziału na modele analityczne i symulacyjne (numeryczne), modele symboliczne mogą być również określane jako preskryptywne lub opisowe, dyskretne lub ciągłe, stochastyczne lub deterministyczne, statyczne lub dynamiczne, z otwartą lub zamkniętą pętlą.

Modele analityczne/numeryczne

W modelu analitycznym uzyskuje się rozwiązanie problemu poprzez badanie zależności zapisanych za pomocą wzorów i formuł matematycznych. Modele numeryczne dostarczają natomiast rozwiązań poprzez wykonywanie operacji na liczbach. Otrzymywane tą drogą wyniki są na ogół przybliżone.

Modele preskryptywne/opisowe

Modele preskryptywne są wykorzystywane w celu sformułowania i optymalizacji konkretnego problemu. Powinny zatem dostarczać jednego, najlepszego rozwiązania. Modele opisowe charakteryzują zachowanie się systemu poprzez wyznaczenie oszacowań badanych zmiennych wyjściowych. Modele symulacyjne to najczęściej modele opisowe (deskryptywne), natomiast modele preskryptywne zaliczane są do modeli analitycznych.

Modele dyskretne/ciągłe

Klasyfikacja ta odnosi się do zmiennych występujących w modelu. Zmienne ciągłe mogą przyjmować wartość każdej liczby rzeczywistej, podczas gdy zmienna dyskretna przyjmuje wartości tylko z pewnego, ściśle określonego przeliczalnego podzbioru liczb rzeczywistych. Szczególnie istotny jest tutaj charakter zmiennej czasu. Jeżeli zmiany w modelu występują w sposób ciągły w miarę upływu czasu, to określamy model jako ciągły. Jeżeli natomiast zmiany występują tylko w ściśle określonych chwilach – mówimy o modelu dyskretnym.

W procesach chemicznych zmiany temperatury i ciśnienia dokonują się w sposób ciągły w czasie. Zjawisko rozprzestrzeniania się wirusów, wywołujących choroby za-

każne (np. grype), mogłoby być również opisane za pomocą modelu ciągłego. Z kolei do obiektu handlowego klienci przybywają, są obsługiwani, kończą zakupy w ściśle określonych momentach. Do opisu takiego systemu należałoby zatem użyć modelu dyskretnego.

Należy zauważyć, że mało jest systemów, które są wyłącznie ciągle lub dyskretne. Przykładowo, wykonanie operacji na maszynie może przebiegać w sposób ciągły, ale początek i koniec operacji są zjawiskami dyskretnymi.

Modele stochastyczne/deterministyczne

W modelu deterministycznym wszystkie obiekty mają jednoznacznie ustalone względem siebie matematyczne powiązania, co prowadzi do całkowitej jednoznaczności rozwiązań. W modelu stochastycznym przynajmniej część zmiennych ma charakter losowy i występuje pod postacią zmiennych losowych. Wszystkie systemy, w których pewną rolę odgrywa człowiek, a także wiele systemów opartych jedynie na maszynach, to systemy stochastyczne. Stochastyczny będzie na przykład rzeczywisty harmonogram odlotów i przylotów w porcie lotniczym. Jako przykład relacji o charakterze deterministycznym moglibyśmy natomiast wskazać kierunek zmian w poziomie zanieczyszczeń wód na skutek obfitych opadów deszczu. Całkowicie zautomatyzowana linia produkcyjna może być również opisana modelem deterministycznym.

Modele statyczne/dynamiczne

W modelu statycznym albo pomijany jest udział czasu, albo też opisywany jest chwilowy stan systemu w pewnym momencie. W modelu dynamicznym wyraźnie podkreślany jest wpływ czasu, co pozwala obserwować zmiany pojawiające się w systemie w kolejnych chwilach.

Modele otwarte/zamknięte

W modelach otwartych uzyskane wyniki nie są wykorzystywane do prowadzenia modyfikacji danych wejściowych w celu poprawy wyników. W modelach zamkniętych wyniki porównuje się z założonym celem, czego konsekwencją jest modyfikacja danych wejściowych tak, aby uzyskać postawiony cel.

Klasycznym przykładem systemu zamkniętego jest system ogrzewania. Dokonywany jest pomiar temperatury i w zależności od wyniku porównania zmierzonej temperatury z temperaturą zamierzoną – system grzewczy jest włączany lub wyłączany. Innym przykładem systemu zamkniętego jest przedsiębiorstwo, w którym podejmowane są decyzje, a następnie obserwowany jest ich wpływ na założone parametry strategii rozwoju firmy. Jeżeli porównanie nie jest satysfakcjonujące, wprowadzane są zmiany, podejmowane są kolejne decyzje i ponownie obserwowany jest efekt.

1.5. CELE BADAŃ SYMULACYJNYCH

Jak podaje Gordon [1974], modele symulacyjne są konstruowane w celu *analizy, projektowania* lub tzw. *doboru modelu*.

Celem analizy jest poznanie działania systemu *istniejącego* lub *proponowanego* (hipotetycznego). Rozważmy przypadek firmy produkcyjnej, która ostatnio zanotowała wzrost zamówień, lecz okazuje się, że ma coraz poważniejsze trudności w dotrzymaniu terminów dostaw dla swoich klientów. Za pomocą modelu symulacyjnego można zbadać, w jakim stopniu obowiązujące reguły wyznaczania terminów dostaw, harmonogramy produkcyjne oraz zamówienia surowcowe wpływają na powstawanie zaobserwowanych opóźnień (opis systemu istniejącego). Z kolei kierownictwo pewnego szpitala zamierza zainstalować nowy, skomputeryzowany system uzupełniania zapasu leków i narzędzi lekarskich. Można zbudować model symulacyjny, w którym zastosowane byłyby dane historyczne w celu sprawdzenia, jaki byłby przeciętny wzrost nakładów inwestycyjnych, które należałoby ponieść oraz jak często w nowym systemie występowałyby określone braki w zaopatrzeniu (analiza hipotetycznego systemu).

Celem *projektowania* jest stworzenie systemu spełniającego pewne założenia. Rozważmy zakład produkcyjny, w którym maszyny są zlokalizowane według priorytetów określonych dla każdego typu operacji. Firma może zbudować model symulacyjny w celu wyznaczenia efektywnej metody ustalania priorytetów tak, aby operacje były wykonywane bez długich przestojów i aby wykorzystanie zasobów było odpowiednio duże.

Dobieranie modelu stosuje się w badaniach socjologicznych, ekonomicznych, politycznych i medycznych, czyli wszędzie tam, gdzie znane jest zachowanie się systemu, ale nie są znane wywołujące je procesy. Stawiane są hipotezy o przypuszczalnym zbiorze obiektów i działań mogących tłumaczyć zachowanie się systemu. Następnie porównuje się odpowiedź modelu opartego na tych hipotezach z zachowaniem się systemu rzeczywistego. W przypadku zgodności oczekuje się, że znajomość modelu pozwoli na lepsze zrozumienie systemu.

Altiok i Melamed [2007] definiują cele symulacyjne bardzo podobnie, podając trzy następujące motywacje: *ocena* działania systemu w warunkach rutynowych i ekstremalnych, *przewidywanie zachowania* systemu nieistniejącego oraz *porównanie* proponowanych rozwiązań.

Najczęściej cele badania symulacyjnego są złożone: za pomocą modelu analizuje się działanie systemu istniejącego (ocena), następnie wprowadza się modyfikacje (projektowanie), dążąc do sformułowania poprawnej prognozy, aby w ostateczności przeprowadzić wartościowanie proponowanych rozwiązań (porównanie).

1.6. ZALETY I WADY SYMULACJI

Jest wiele powodów, dla których warto stosować symulację.

System może być zbyt złożony, aby możliwe było zbudowanie modelu za pomocą metod analitycznych. Większość systemów rzeczywistych to systemy o takim stopniu złożoności, że „rozwiązywalny” model analityczny musiałby być albo bardzo uproszczony, albo niewiarygodny. Jeżeli nie jest możliwe analityczne rozwiązanie modelu matematycznego, to warto zastanowić się nad wyborem podejścia symulacyjnego, które zamiast prób poszukiwania jednoznacznego rozwiązania pozwala *poznać* system poprzez *obserwowanie* zachowania modelu.

Symulacja umożliwia prowadzenie eksperymentów bez ingerowania w system rzeczywisty i bez udziału ludzi, co jest szczególnie istotne w przypadku badań niosących utrudnienia dla systemu rzeczywistego, stwarzających zagrożenie czy też generujących duże koszty. Badania symulacyjne można ponadto prowadzić na systemie nieistniejącym, planowanym, mającym pojawić się dopiero w przyszłości.

W trakcie symulacji można dowolnie kondensować czas (badać kilka lat działalności systemu w ciągu minuty) i w ten sposób rozważać różnorodne plany operacyjne w bardzo krótkim czasie lub też rozciągać czas działania systemu (badać szczegółową strukturę zmian, której nie można zaobserwować w czasie rzeczywistym).

Symulacja dostarcza badaczowi doskonałej bazy eksperymentalnej. Obliczenia mogą być szybko powtarzane, co prowadzi do doskonalenia struktury modelu i umożliwia jego wielokrotne testowanie. Dzięki zdolności do odtwarzania stanu systemu można uruchomić symulację dla rozszerzonego zestawu danych wejściowych przy identycznych warunkach początkowych. Eksperyment symulacyjny może zostać całkowicie powtórzony po dokonaniu pewnych zmian w parametrach lub warunkach działania systemu. Z tych powodów model symulacyjny doskonale nadaje się do badań typu *what-if?*, czyli *co-się-stanie-jeśli?*

Zarządzanie wymaga podejmowania decyzji obarczonych ryzykiem. Symulacja jest jednym z nielicznych dostępnych narzędzi, które umożliwia badanie zjawisk charakteryzujących się losowością. Istotną zaletą symulacji jest łatwość wprowadzenia do modelu dowolnych zmiennych o charakterze losowym, czego nie umożliwiają inne metody analizy. Bardzo często spotykany w innych podejściach silny wymóg normalności rozkładu w symulacji nie występuje.

Brak jednolitej teorii symulacji pozwala natomiast na zbudowanie modelu charakteryzującego się dowolną liczbą dynamicznych współzależności. Łatwość wprowadzenia do modelu symulacyjnego dowolnego założenia, szczególnie w sytuacjach braku odpowiednich propozycji ze strony modeli analitycznych, czyni z symulacji atrakcyjne narzędzie badawcze. Za pomocą symulacji można badać wrażliwość systemu na zmianę warunków działania, np. czy zachowana zostanie dotychczasowa jakość obsługi klientów po zwiększeniu obrotów o 20%? Model symulacyjny pozwala ocenić planowaną do wdrożenia strategię działania, np. przy określaniu reguł odnawiania zapasów w magazynie.

Symulacja nie jest jednak metodą idealną. Podstawowe wady symulacji to brak ogólnych reguł tworzenia modeli symulacyjnych i brak jednolitej teorii symulacji. Co więcej, im bardziej złożony jest model, tym bardziej jesteśmy zależni od słabo rozwiniętej teorii statystyki, zajmującej się analizą i generowaniem danych. Modele symulacyjne nie mają zamkniętej formy rozwiązania. Każda zmiana warunków wejściowych powoduje konieczność wykonania ponownie pełnych badań symulacyjnych.

Istotnym mankamentem symulacji jest brak uniwersalnych metod rozstrzygnięcia o poprawności budowanych modeli. Model nie jest nigdy po prostu „dobry” lub „zły”. Każde badanie symulacyjne jest unikatowe, a uzyskane rozwiązanie trudno jest przenieść na inny, podobny obiekt rzeczywisty.

Ponieważ większość rzeczywistych systemów charakteryzuje się występowaniem pewnej, często znacznej, liczby czynników losowych, uzyskiwane wyniki także będą miały losowy charakter. Stochastyczny charakter większości modeli symulacyjnych wymusza konieczność traktowania uzyskanych wyników jako obciążonych błędem statystycznym i stosowania odpowiedniej procedury analizy danych wyjściowych. Wartości uzyskane w trakcie symulacji są jedynie *oszacowaniami* prawdziwych charakterystyk, uzyskanymi dla danego zestawu parametrów wejściowych. Na przykład w centrum logistycznym przybycia i odprawy towarów przebiegać będą z różną częstotliwością, którą można opisać pewnym rozkładem losowym. Wskaźniki funkcjonowania tego centrum, takie jak przepustowość czy czas trwania jednego cyklu, będą również wielkościami losowymi. Jednokrotne uruchomienie modelu symulacyjnego można utożsamić z obserwacją centrum tylko przez jeden dzień – dane uzyskane dnia następnego mogą być już krańcowo inne, chociaż nie zmieniły się warunki funkcjonowania systemu. W badaniu symulacyjnym radzimy sobie z tym problemem w ten sposób, że wystarczająco długo obserwujemy model. Większość wyników zaczyna wtedy wyraźnie zbliżać się do pewnej średniej. Pytanie tylko jak długo to jest „wystarczająco długo”? Pojawia się ponadto problem tych systemów, które wyraźnie wymagają końca badania symulacyjnego (np. bank jest czynny od 9:00 do 17:00), więc dłuższe obserwowanie modelu jest niepoprawne.

1.7. WARIANTY SYMULACJI

Symulacja często określana jest przymiotnikami przejętymi od rodzaju budowanego modelu. Mówimy na przykład o symulacji statycznej lub dynamicznej, ciągłej lub dyskretnej, deterministycznej lub stochastycznej. Oczywiście, symulacja może być jednocześnie np. statyczna i stochastyczna bądź dynamiczna, stochastyczna i dyskretna, bądź ciągła i deterministyczna itd.

W literaturze można znaleźć różne typologie prezentowanych podejść symulacyjnych.

Law i Kelton [1991] klasyfikują modele symulacyjne w trzech przekrojach i wyróżniają odpowiednio: symulację *statyczną i dynamiczną, deterministyczną i stochastyczną, ciągłą i dyskretną*.

Seila i in. [2003] wspominają o trzech typach podejść symulacyjnych: *grach symulacyjnych, symulacji statycznej* (tu wymieniają metodę *Monte Carlo*) *i dynamicznej* (w tej grupie wyróżniają symulację *dyskretną*, czyli *zdarzeniową* i metodę *dynamiki systemów*).

Z kolei Render i in. [2003] opisują symulację *Monte Carlo, dyskretną, metodę dynamiki systemów* oraz *gry symulacyjne*.

Evans i Olsen [2002] wyróżniają tylko dwa rodzaje modeli symulacyjnych: *modele Monte Carlo* i *modele symulacji systemów*, co pozwala im zdefiniować *symulację Monte Carlo w arkuszu kalkulacyjnym* (*Monte Carlo simulation in spreadsheets*) i *symulację systemów* (*system simulation*), wyróżniając w tej grupie *symulację przeglądu działań, symulację procesową, symulację zdarzeniową i symulację ciągłą*.

W podręcznikach polskich autorów można dowiedzieć się o symulacji systemów ciągłych [Krupa 2008], o symulacji stochastycznej [Snopkowski 2007] oraz o symulacji dyskretniej i ciągłej [Gajda 2001].

W dalszej części książki zostaną szczegółowo omówione dwa warianty symulacji: symulacja dyskretna-zdarzeniowa (dynamiczna, stochastyczna, dyskretna) oraz metoda Monte Carlo (którą można sklasyfikować jak symulację stochastyczną, dyskretną, najczęściej statyczną). Naturalnym środowiskiem dla symulacji Monte Carlo jest arkusz kalkulacyjny, natomiast symulacja dyskretna prowadzona jest zazwyczaj za pomocą odpowiednich specjalizowanych narzędzi (środowisk). W książce zaprezentowane będą modele zbudowane za pomocą arkusza kalkulacyjnego Excel oraz środowiska *Arena*, stworzonego na bazie języka *SIMAN*.

1.8. PRZEBIEG ANALIZY SYMULACYJNEJ

1.8.1. OGÓLNY SCHEMAT BADANIA SYMULACYJNEGO

Analiza symulacyjna jest deskryptywną techniką modelowania. Oznacza to, że nie istnieje precyzyjnie określony zestaw reguł, procedur, kroków, którego przestrzeganie i stosowanie doprowadziłoby do powstania modelu. Procedura budowy i wykorzystania modelu symulacyjnego musi być stworzona każdorazowo podczas prowadzenia analizy symulacyjnej. Można jednak wskazać pewne ogólne etapy prowadzenia symulacji (por. rys. 1.4), które nie muszą być realizowane sekwencyjnie. Jak w każdym procesie modelowania może zachodzić konieczność powrotu do

wcześniejszych etapów w miarę napływu nowych informacji lub w przypadku gdy uzyskane wyniki sugerują konieczność wprowadzenia modyfikacji do modelu.

Etapy budowy modelu symulacyjnego to (por. rys. 1.4):

1. Sformułowanie problemu.
2. Zebranie i analiza danych wejściowych.
3. Budowa modelu matematycznego.
4. Opracowanie programu komputerowego.
5. Walidacja i weryfikacja modelu.
6. Projektowanie i przeprowadzenie układu eksperymentów.
7. Analiza wyników.

Poniższy przykład będzie wykorzystany do zilustrowania przebiegu analizy symulacyjnej.

W supermarkecie znajduje się kilka kas obsługujących klientów. Kierownictwo zastanawia się nad zmianą organizacji pracy stanowisk kasowych w celu polepszenia standardu obsługi klientów.

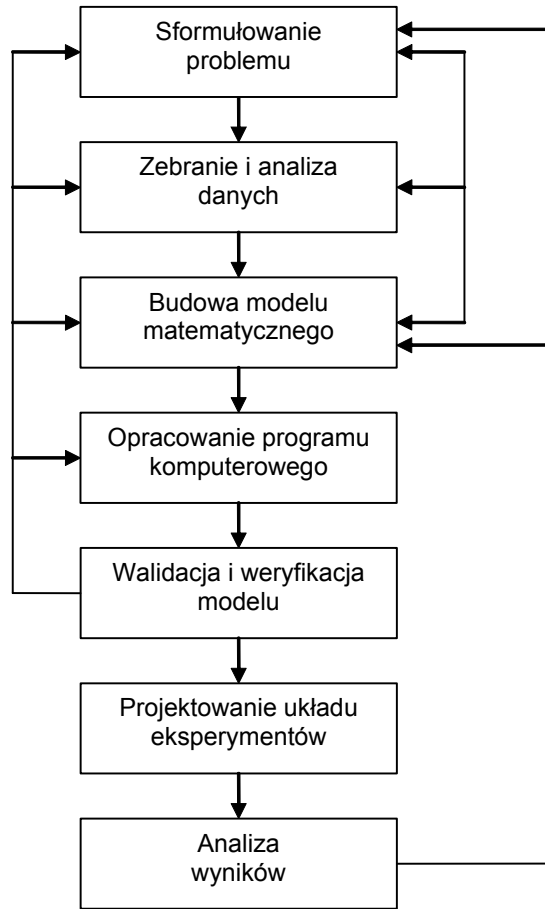
1.8.2. SFORMUŁOWANIE PROBLEMU

Jest to najistotniejszy krok analizy symulacyjnej. Nie jest możliwe uzyskanie poprawnego rozwiązania przy niepoprawnie sformułowanym problemie. Komputerowe eksperymenty symulacyjne powinny rozpocząć się albo od sformułowania problemu, albo od jasnego sprecyzowania celów eksperymentu. Cele te przyjmują zazwyczaj postać pytań, na które udziela się odpowiedzi, hipotez, które poddaje się testom sprawdzającym oraz skutków będących przedmiotem estymacji (por. [Naylor 1974]).

W omawianym przykładzie sformułowany problem mógłby brzmieć następująco: w jaki sposób zmodyfikować harmonogram pracy stanowisk kasowych, aby nie zwiększając kosztów zatrudnienia poprawić jakość obsługi klientów?

Nie wystarczy oczywiście ściśle określenie pytań, należy również dokładnie określić kryteria oceny odpowiedzi uzyskiwanych na te pytania. Jak na przykład będziemy mierzyć jakość obsługi klientów?

Aby model rzeczywiście okazał się użyteczny, musimy odszukać i sformułować *problem* występujący w systemie, który rodzi potrzebę uzyskania dodatkowej wiedzy i lepszego zrozumienia systemu rzeczywistego. Badacz powinien również wziąć pod uwagę odbiorców modelu. Model wyjaśniający istotę „kwaśnych deszczy” będzie wyglądał zupełnie inaczej, jeżeli kierowany będzie do studentów biologii, a inaczej do Ministerstwa Ochrony Środowiska.



Rys. 1.4. Etapy procesu symulacji
Źródło: [Law i Kelton 1991]

1.8.3. ZEBRANIE I ANALIZA DANYCH WEJŚCIOWYCH

Metody gromadzenia danych są bardzo różne, poczynając od sposobów manualnych po bardzo zaawansowane technicznie. Przy wyborze metody należy kierować się: możliwością szybkiego wprowadzenia danych do komputera, możliwością zebrania danych z pożądaną dokładnością, wpływem procesu zbierania danych na zachowanie się atrybutów systemu, które właśnie podlegają pomiarom. Najefektywniej jest korzystać z już istniejących zbiorów danych (w przykładzie: informacjami o liczbie zakupionych towarów, czasie finalizowania jednej transakcji), jednakże zawsze bazy

te należy obdarzać ograniczonym zaufaniem. Oprócz problemu *jak* zbierać, pozostaje jeszcze problem *ile* zbierać. Oczywiście im większy jest zbiór zgromadzonych danych, tym dokładniejsze będą charakterystyki wyznaczone na ich podstawie. Jednocześnie jednak zwiększa się czas i/lub koszt zbierania danych oraz czas i/lub koszt ich analizy. Teoria statystyki pomaga rozwiązać ten problem.

Zebrane dane można wykorzystać w modelu w dwojaki sposób: użyć zebranej próbki do odtworzenia rozkładu statystycznego lub na podstawie próbki określić teoretyczny rozkład statystyczny (analiza danych wejściowych – patrz rozdz. 2).

1.8.4. BUDOWA MODELU MATEMATYCZNEGO

Budowa modelu matematycznego jest bardziej sztuką niż nauką. Wśród twórców modeli symulacyjnych panuje jednak zgodne przekonanie, że dobrze jest zaczynać od modelu o wysokim stopniu ogólności, w którym ujęte byłyby wszystkie najważniejsze relacje. W fazie następnej należy uszczegółowić model.

Budowę modelu należy rozpocząć od przeprowadzenia następujących badań:

- zidentyfikowania zmiennych decyzyjnych i zmiennych kontrolowanych,
- określenia ograniczeń dla zmiennych decyzyjnych,
- zdefiniowania mierników działania systemu i funkcji celu,
- określenia wstępnej struktury modelu w celu ustalenia wzajemnych powiązań pomiędzy zmiennymi.

Zmienne i ograniczenia

Wyróżniamy *zmienne egzogeniczne*, czyli zmienne zewnętrzne, występujące niezależnie od modelu (odpowiedniki argumentów funkcji), oraz *zmienne endogeniczne*, czyli zmienne wewnętrzne, będące funkcją zmiennych egzogenicznych. Zmienne egzogeniczne mogą być dalej klasyfikowane jako zmienne *kontrolowane* lub *niekontrolowane*. Zmienne kontrolowane to takie zmienne, na które decydent może wpływać, natomiast zmienne niekontrolowane są niezależne od woli (i życzeń) decydenta. Zmienne kontrolowane określa się często jako *zmienne decyzyjne*, natomiast zmienne niekontrolowane jako *parametry* modelu.

W przykładzie zmienne egzogeniczne to: liczba kas, sposób sczytywania cen (skaner, klawiatura), istnienie (lub nie) linii ekspresowych dla klientów z liczbą towarów nie większą niż pięć, częstotliwość przybywania klientów do systemu, płace pracowników.

Z punktu widzenia kierownika sklepu zmienne niekontrolowane (parametry) to: liczba kas, wyposażenie, częstotliwość przybywania klientów, płace pracowników. Z punktu widzenia właściciela sklepu wszystkie zmienne, z wyjątkiem częstotliwości przybywania klientów, są zmiennymi decyzyjnymi.

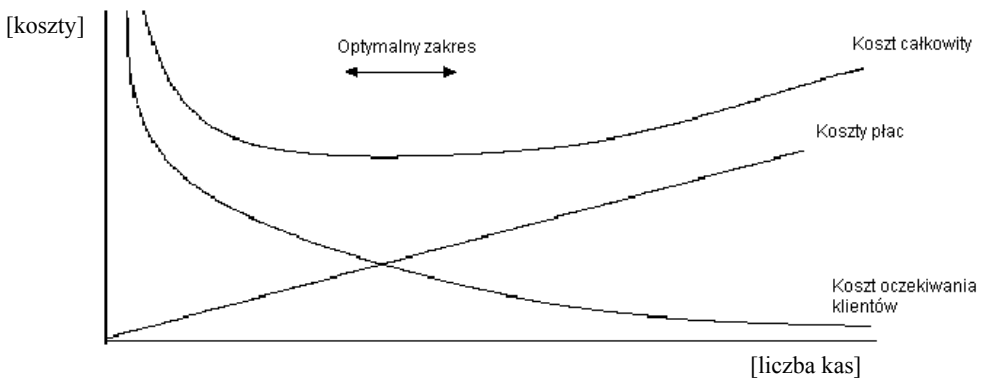
W przypadku zmiennych decyzyjnych występują oczywiście ograniczenia ich zmienności. Ograniczeniem jest na przykład powierzchnia, która może być przeznaczona na stanowiska kasowe, maksymalny koszt urządzeń kasowych itd.

Zmienne endogeniczne to z kolei: liczba oczekujących klientów, czas oczekiwania klientów, stopień wykorzystania kas, czas obsługi jednego klienta.

Mierniki działania i funkcja celu

Aby ocenić jakość uzyskanego rozwiązania, należy najpierw zdefiniować mierniki, którymi będziemy się w tej ocenie posługiwać. Mierniki działania systemu wybierane są spośród zmiennych endogenicznych. Te mierniki, które wybierzemy do minimalizacji lub maksymalizacji charakterystyk systemu, będziemy traktować jako funkcję celu.

W przykładzie możemy zdecydować się na minimalizację czasu oczekiwania klienta na obsługę. Jeżeli jednak nie będziemy uważać na stopień wykorzystania stanowisk, koszty obsługi będą wysokie (por. rys. 1.5). Jeżeli wybierzemy oba mierniki, nie będziemy ich prawdopodobnie mogli optymalizować jednocześnie. Oczywiście najbardziej pożądana byłaby sytuacja, w której czas oczekiwania klientów byłby minimalny, a stopień wykorzystania kas maksymalny. Problem ten można rozwiązać w dwojaki sposób: połączyć oba mierniki poprzez wprowadzenie nowego, wspólnego, np. kosztu, lub wybrać jedno kryterium jako dominujące, a dla drugiego wyznaczyć minimalny akceptowalny zakres.



Rys. 1.5. Wybór mierników działania

Źródło: [Hoover i Perry 1989]

W przykładzie, posługując się metodą drugą, można jako kryterium dominujące wybrać czas oczekiwania klientów, natomiast wykorzystanie stanowisk ograniczyć minimalną akceptowalną wartością. Wybierając metodę pierwszą, można stworzyć krzywą kosztu jako hipotetyczną funkcję liczby stanowisk kasowych i określić optymalną liczbę kas. Koszt wykorzystania kas można powiązać z kosztami płac, jak jednak określić koszt oczekiwania klienta? Trudności praktyczne wynikające z powiązania kilku mierników są często tak duże, że decydujemy się zwykle na jedną charakterystykę.

Ustalanie wstępnej struktury modelu

Ponieważ interesuje nas tylko obsługa przy kasach, więc zakładamy, że klient wchodzi do systemu w momencie, kiedy decyduje się na jedną z kolejek, a opuszcza

go po zapłaceniu i spakowaniu produktów. W systemie możemy wyróżnić następujące operacje: wypakowanie produktów na taśmę przy kasie (przez klienta), sczytanie kodów produktów (przez kasjera), zapłata za towar (klient i kasjer), pakowanie produktów (klient lub pracownik sklepu).

Jak bardzo należy uszczegółowić model? Czas obsługi klienta przy kasie jest zmienną endogeniczną, na którą mają wpływ takie zmienne egzogeniczne, jak: liczba zakupionych artykułów, doświadczenie kasjera, nastrój kasjera i in. Które z nich należy uwzględnić, a które można pominąć? Decyzja jest wynikiem porównania kosztów wprowadzenia dodatkowych danych do modelu i wzrostu dokładności modelu.

1.8.5. OPRACOWANIE PROGRAMU KOMPUTEROWEGO

Najistotniejsze kroki, które należy wykonać, tłumacząc model matematyczny na program komputerowy, to:

a) budowa schematu blokowego głównego szkieletu programu;

b) wybór języka programowania: GPSS (*General Purpose Simulation Language*), SLAM (*Simulation Language for Alternative Modeling*), SIMAN (*Simulation Analysis*), SIMSCRIPT, FORTRAN, PASCAL, BASIC, C++, lub pakiet do symulacji graficznej (np. Arena, Extend, Vensim i inne).

Przewagą języków symulacyjnych jest to, że mają już wbudowane specjalne procedury „symulacyjne” (sterowanie upływem czasu, generatory liczb losowych, narzędzia statystyczne służące do analizy wyników), które znacznie przyspieszają i ułatwiają prace nad budową modelu symulacyjnego. Odbywa się to niestety kosztem objętości programu i pewnych „uszczywnień” w opisie systemów;

c) wybór i oprogramowanie generatorów liczb losowych i zmiennych losowych. W każdym stochastycznym modelu symulacyjnym rzeczą ogromnej wagi jest właściwy dobór generatorów liczb i zmiennych losowych. W przykładzie czas nadejścia kolejnego klienta generowany byłby na podstawie pewnego rozkładu losowego;

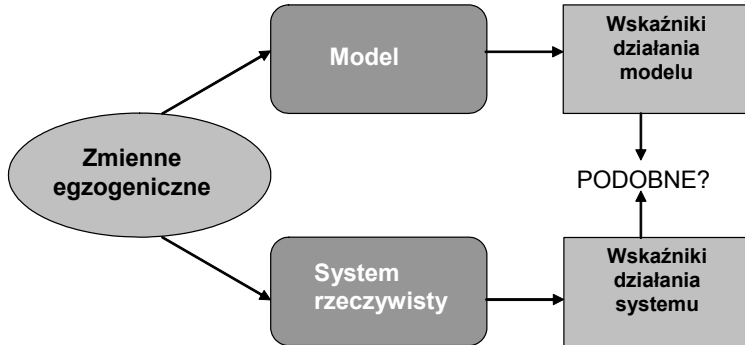
d) napisanie i sprawdzenie programu.

1.8.6. WERYFIKACJA I WALIDACJA MODELU

Proces weryfikacji i walidacji dotyczy bardzo istotnego problemu, który polega na ustaleniu, czy model symulacyjny poprawnie opisuje system rzeczywisty poddany badaniom. Innymi słowy – czy model jest *wiarygodny*. *Weryfikacja* dotyczy wewnętrznej spójności modelu, natomiast *walidacja* odnosi się do związków pomiędzy modelem a rzeczywistością.

Nie ma pełnej i klarownej zgodności, co to znaczy, że model jest wiarygodny. Najogólniej można powiedzieć, że za pomocą wiarygodnego modelu potrafimy z dużą dokładnością opisać wydarzenia w systemie rzeczywistym (por. rys. 1.6). W przykładzie, za pomocą modelu powinniśmy w sposób dokładny wyznaczyć czas oczekiwa-

nia klientów oraz stopień wykorzystania stanowisk, w przypadku gdy w systemie wprowadzone zostanie na przykład jeszcze jedno stanowisko kasowe.



Rys. 1.6. Ogólna idea weryfikacji i walidacji modelu

1.8.7. PROJEKTOWANIE UKŁADU EKSPERYMENTÓW

Celem prowadzenia badań symulacyjnych jest dostarczenie informacji o zachowaniu się systemu, które byłyby użyteczne w procesach podejmowania decyzji. Wyniki procesu symulacji nie dostarczają wprost odpowiedzi na pytania odnoszące się do charakterystyk systemu. Wynikiem symulacji są zbiory danych, z których należy wyznaczyć szukane charakterystyki. Wyniki symulacji muszą zatem spełniać warunki prób losowych, tj. być reprezentatywne dla typowego zachowania systemu i być odpowiednio liczne.

Wyróżniamy dwa typy analiz symulacyjnych: *terminating* (dotyczy systemów rozpoczynających i kończących kolejne cykle pracy, w wolnym tłumaczeniu można tu użyć określenia *symulacje pulsacyjne*), i *steady state* (*nonterminating*, symulacje *niezmiennicze*, czyli odnoszące się do systemów działających w sposób nieprzerwany). *Symulacja pulsacyjna* stosowana jest w sytuacji, kiedy koniec procesu symulacji jest zdefiniowany określonym zdarzeniem (np. godzina zamknięcia sklepu). *Symulacja niezmiennicza* to z kolei długookresowa analiza odnosząca się do okresu stabilności systemu działającego bez przerw (np. system ratownictwa medycznego). Sposób prowadzenia eksperymentów różni się w obu przypadkach. W symulacji pulsacyjnej otrzymujemy gotową do analizy próbę złożoną z niezależnych powtórzeń, na której możemy zastosować klasyczne techniki analizy statystycznej. Do rozstrzygnięcia pozostaje właściwy dobór liczby powtórzeń, warunków początkowych i warunków zakończenia eksperymentu. W symulacji niezmienniczej musimy zmierzyć się z problemem autokorelacji wyników i długością okresu początkowego, po którym rozpoczyna się właściwy okres stabilności działania systemu. Najtrudniejszym zagadnieniem jest pozyskanie próby losowej, na której możliwe będzie przeprowadzenie analizy statystycznej.

W trakcie projektowania układu eksperymentów posługujemy się pojęciami *czynnik* oraz *reakcji*. Oba te pojęcia odnoszą się do zmiennych. W zależności od roli pełnionej przez zmienną w eksperymencie może być ona traktowana jako czynnik lub reakcja. Przyjmijmy, że mamy dwie zmienne X oraz Y . Jeżeli w naszym eksperymencie szukamy odpowiedzi na pytanie: *w jaki sposób zmiany w wartości X oddziałują na Y* , to X określamy jako czynnik, a Y – jako reakcję. Reakcją jest zwykle zmienna endogeniczna (obserwowana na wyjściu), natomiast czynnikiem jest zmienna egzogeniczna lub decyzyjna (obserwowana na wejściu).

Cztery główne problemy występujące w projektowaniu eksperymentów (por. [Naylor 1975]) to problem zbieżności stochastycznej, problem rozmiaru badania, problem motywu i problem reakcji wielokrotnej.

Problem zbieżności stochastycznej

W eksperymentach symulacyjnych poszukuje się najczęściej informacji o wartościach przeciętnych badanej zmiennej (np. o średniej wartości zysku, średnim czasie oczekiwania, średnim wykorzystaniu stanowisk obsługi itd.). Wykonując wiele prób symulacyjnych, zwiększa się prawdopodobieństwo tego, że przeciętne wartości z próby będą dokładniej odpowiadać przeciętnym wartościom populacji (tzw. *zbieżność stochastyczna*). Problem polega na tym, że zbieżność stochastyczna przebiega bardzo powoli. Jeżeli σ jest odchyleniem standardowym zmiennej dla pojedynczej obserwacji, to odchyleniem standardowym dla przeciętnej wartości n obserwacji jest σ/\sqrt{n} . Zmniejszenie o połowę błędu losowego wymaga czterokrotnego zwiększenia wielkości próby n .

Problem rozmiaru badania

Problem ten można nazwać „problemem zbyt wielu czynników”. Pełny eksperyment symulacyjny wymagałby przeprowadzenia tylu prób, ile wynosi iloczyn wszystkich poziomów wartości u wszystkich czynników uwzględnianych w eksperymencie. Jeśli w eksperymencie dotyczącym sprawności pracy obiektu handlowego wyróżnimy 6 różnych harmonogramów pracy, 5 alternatywnych planów wynagrodzeń pracowników, 5 możliwych schematów organizacji pracy oraz 10 różnych akcji promocyjnych, to otrzymamy $6 \times 5 \times 5 \times 10 = 1500$ kombinacji czynników. Gdybyśmy eksperymentowali tylko na dziesięciu czynnikach i przyjęli tylko dwa poziomy dla każdego z nich, pełny eksperyment wymagałby $2^{10} = 1024$ kombinacji. Jeżeli wymagamy pełnego układu czynników, to rozwiązanie problemu jest trudne lub wręcz niemożliwe do uzyskania. Jeżeli jednak zdecydujemy się na układ mniej kompletny, to można opracować takie metody projektowania zestawu eksperymentów, które umożliwią realizację naszych celów i będą wymagać mniejszej liczby kombinacji, niż narzuca metoda projektowania pełnego układu.

Problem motywu

Ze względów statystycznych istnieją dwa główne cele symulacji: *porównanie* wyników powiązanych z alternatywnymi warunkami działania tak, aby możliwa była

optymalizacja procesów oraz *wykrywanie i estymacja* funkcjonalnych powiązań, jakie istnieją między wejściowymi czynnikami ilościowymi a wynikami. Precyzyjne sformułowanie celów badania symulacyjnego ułatwia zaplanowanie właściwego układu eksperymentów.

Problem reakcji wielokrotnej

W badaniu symulacyjnym obserwujemy często wiele zmiennych będących reakcjami w danym eksperymencie. W symulacji pracy supermarketu przedmiotem zainteresowania może być liczba obsłużonych klientów, liczba niezadowolonych klientów, wykorzystanie stanowisk obsługi, kolejki przed stanowiskami kasowymi. Rozwiązaniem może być potraktowanie eksperymentu o wielu reakcjach jako wiele eksperymentów, w których występuje reakcja pojedyncza. Można też zrealizować kombinacje kilku reakcji, tworząc reakcję pojedynczą. Niemniej jednak ominięcie problemu reakcji wielokrotnych nie zawsze jest możliwe.

Ogólnie, w teorii planowania układu eksperymentów wyróżnia się dwa podejścia:

a) *łączenie informacji a priori o systemie z danymi, które gromadzone są w trakcie trwania symulacji*. Podejście to polega na tym, że wyróżnia się czynniki, które (jak domyślamy się lub wiemy) mają wpływ na miary końcowe i przeprowadzamy eksperymenty nadając tym czynnikom różne (zbiory) wartości. Następnie stosujemy metodę redukcji wariancji, aby wybrać czynniki, dla których ten wpływ możemy potwierdzić. Wadą tego podejścia jest to, że nie pozwala na znalezienie *najlepszego* rozwiązania, ani nawet nie gwarantuje, że znalezione rozwiązanie jest optymalne lokalnie,

b) *metody szukania rozwiązania optymalnego*. Podejście to polega na tym, że przeprowadza się dużą liczbę badań symulacyjnych, modyfikując zmienną decyzyjną i obserwuje się zmiany mierników działania systemu, aż do momentu, kiedy zmiany ponownie następują w przeciwnym kierunku. Oznacza to, że znalezione zostało optimum lokalne.

1.8.8. ANALIZA WYNIKÓW

Kluczowym zadaniem w analizie wyników symulacji stochastycznych jest zebranie obserwacji z wielu niezależnych powtórzeń symulacyjnych. Większość klasycznych technik analizy statystycznej, takich jak wyznaczanie przedziałów ufności przy szacowaniu średniej, obliczanie standardowego błędu średniej, analiza wrażliwości, porównanie średnich w dwóch populacjach wyników, analiza wariancji ANOVA i inne, może być wtedy automatycznie wykorzystana.

Metoda Monte Carlo i tzw. pulsacyjne (*terminating*) symulacje dynamiczne automatycznie zapewniają uzyskanie zbioru niezależnych powtórzeń, a użytkownikowi pozostaje do rozstrzygnięcia, jaka liczba powtórzeń powinna być wykonana, aby osiągnąć satysfakcjonującą dokładność wyników. Większa liczba powtórzeń zapewnia większą precyzję szacowania średniej, a jako miernik dokładności służą *przedziały ufności*. Przedział ufności jest przedziałem „losowym” (następna seria powtórzeń za-

kończy się nieco innymi wartościami brzegowymi przedziału ufności), który z określonym (przyjętym na wstępie) prawdopodobieństwem pokrywa rzeczywistą wartość średniej populacji μ . Jest oczywiste, że im przedział ufności jest mniejszy, tym precyzyjniej wyznaczyliśmy wartości badanych zmiennych. Obserwowanie długości przedziału ufności pozwala zatem dobrać właściwą liczbę powtórzeń symulacyjnych n i poprawnie określić czas trwania powtórzenia.

Ważnym miernikiem precyzji oszacowań symulacyjnych jest *standardowy błąd średniej* równy σ/\sqrt{n} gdzie σ to odchylenie standardowe w populacji. Długość 68-procentowego przedziału ufności dla średniej odpowiada jednej wartości standardowego błędu średniej odmierzonego po obu stronach średniej z próby, długość 90-procentowego przedziału ufności odpowiada 1,645 wartości błędu, a długość 99-procentowego przedziału ufności to 2,576 wartości błędu. Im mniejsza wielkość próby, tym standardowy błąd średniej jest oczywiście większy.

W symulacjach dynamicznych typu *steady-state* (niezmienniczych) problem jest bardziej złożony, ponieważ należy rozpoznać i pominąć w analizie końcowej tzw. okres rozgrzewania modelu. Jest to okres, w trakcie którego nie został osiągnięty właściwy poziom obciążenia modelu, co z kolei może zaburzać trafną interpretację wyników. W uzyskaniu zbioru obserwacji do próby statystycznej przydatna może okazać się tzw. metoda *średnich odcinkowych* (ang. *batch means method*). Metoda ta często prowadzi jednak do autokorelacji obserwacji, dlatego lepiej jest rozważyć wykonanie wielu tzw. *uciętych* (ang. *truncated*) przebiegów symulacyjnych, w których każdorazowo pominięty jest okres początkowy.

1.9. PROGRAMY SYMULACYJNE

Symulacja początkowo była prowadzona ręcznie. W 1733 roku George Louis Leclerc (później uzyskał on tytuł szlachecki jako Le Comte de Buffon) opisał eksperyment, który pozwala oszacować wartość π . Jeżeli będziemy rzucać igłą o długości l na stół z wymalowanymi liniami równoległymi w odstępach d ($d \geq l$), okaże się, że prawdopodobieństwo przecięcia przez igłę linii na stole wynosi

$$p = \frac{2l}{\pi d} \quad (1.1)$$

Chociaż eksperyment ten wydaje się być niezwykle prosty (by nie rzec trywialny), można wskazać na kilka jego cech, niewątpliwie „symulacyjnych”.

- Celem eksperymentu jest oszacowanie jakiegoś wskaźnika, którego wartość trudno byłoby dokładnie obliczyć (oczywiście w 1733 roku!).
- Końcowe oszacowanie nie będzie nigdy dokładne. *Zawsze* wystąpi błąd oszacowania.

- Im więcej eksperymentów wykonamy (rzutów igłą), tym mniejszy błąd oszacowania popełnimy.

Wraz z rozwojem komputerów pojawiły się szerokie perspektywy dla badań symulacyjnych, którym towarzyszył rozwój narzędzi do prowadzenia symulacji. Był to rozwój ewolucyjny od programów w językach symbolicznych o specjalnych własnościach, przez komercyjne języki problemowe, takie jak FORTRAN, ALGOL, COBOL (a w późniejszym okresie PASCAL i C++) po języki symulacyjne. Język symulacyjny programowania to język problemowy odznaczający się specjalnymi własnościami. Zaletą języków symulacyjnych, takich jak np. GPSS, SIMSCRIPT, SIMULA, SLAM, SIMAN, jest to, że realizują specyficzne potrzeby symulacji: dostarczają odpowiednich reprezentacji danych, zezwalają na łatwe odtwarzanie dynamiki modelowanego systemu, są nastawione na badania systemów stochastycznych, w ich strukturę wbudowane są mechanizmy niezbędne do prowadzenia procesu symulacji. Nadal jednak są to języki programowania i wymagają od użytkownika poznania syntaktyki i semantyki języka.

Kolejnym etapem było powstanie środowisk symulacyjnych (IThink, PowerSim, Vensim, Arena, Extend i inne) zwanych czasami „symulatorami” (nie mylić z „symulatorami fizycznymi”). Są to interaktywne środowiska graficzne niewymagające od użytkownika znajomości języków programowania. Modele buduje się, korzystając z interaktywnych menu, okien dialogowych i gotowych modułów. Ich wadą jest pewne ograniczenie odnoszące się do swobody w modelowaniu dowolnych relacji, związków, zależności w porównaniu z językami programowania. Mogą jednak być stosowane przez użytkowników, którzy nie uczęszczali wcześniej na kursy programowania. Środowiska symulacyjne można ogólnie podzielić na programy do symulacji procesów ciągłych i programy do symulacji procesów dyskretnych, chociaż podział ten jest dość miękki. Proponowane aktualnie użytkownikom narzędzia do prowadzenia symulacji oferują możliwość budowy jednocześnie modeli jednego i drugiego typu za pomocą tego samego środowiska (np. Arena czy Extend).

Rozwój języków symulacyjnych i sukcesywnie udoskonalane specjalizowane środowiska do prowadzenia symulacji nie wyeliminowały podstawowego narzędzia, po które często sięga się przy rozwiązywaniu problemów analityczno-decyzyjnych, jakim jest arkusz kalkulacyjny. Tak zwany *spreadsheet simulation* odgrywa znaczącą rolę w analizie wielu problemów decyzyjnych. Uważa się (por. Powell i Baker [2004]), że arkusze kalkulacyjne to *druga* w kolejności najlepsza platforma pozwalająca na wykonanie analiz i *najlepsza* umożliwiająca budowę modelu. Stwierdzenie to oznacza, że w celu rozwiązania danego problemu decyzyjnego można posłużyć się silniejszym, bardziej elastycznym, o większych możliwościach i lepiej dostosowanym do badanego zagadnienia narzędziem, jednak najczęściej brakuje czasu, pieniędzy i wiedzy, aby sięgnąć po środek najlepszy z możliwych. Praktycznym wyborem staje się wtedy arkusz kalkulacyjny. Dostępne specjalne nakładki symulacyjne do arkuszy kalkulacyjnych (takie jak Crystal Ball czy @Risk), dodatkowo znacząco poszerzają możliwości prowadzenia badań symulacyjnych za pomocą arkusza.

2. ZBIERANIE I ANALIZA DANYCH WEJŚCIOWYCH

2.1. SZTUKA ZBIERANIA DANYCH

Zebranie i analiza danych wejściowych jest istotną fazą procesu symulacji. Niewłaściwe dane wejściowe mogą całkowicie zdyskredytować model symulacyjny poprzez doprowadzenie do błędnych wyników badania symulacyjnego (por. [Uziółko 2006].) W opracowaniach statystycznych najczęściej rozpoczyna się omawianie technik analizy statystycznej danych od założenia, że jesteśmy w posiadaniu próbki losowej składającej się z n niezależnych obserwacji. W praktyce jednakże niezwykle trudne jest uzyskanie „czystej” próbki losowej, która spełniałaby wszystkie założenia statystyczne. Jakość takiej próbki zależy w dużej mierze od umiejętności osoby zbierającej dane. Proces zbierania danych statystycznych jest zwykle długi, nużący i niewdzięczny, ale odgrywa istotną rolę w symulacji. Niewłaściwe (lub niewłaściwie zebrane) dane doprowadzają bowiem do błędów w wynikach symulacji.

Zalóżmy, że chcemy określić, ile czasu klient spędza przy stanowisku kasowym.

Najprostszym sposobem zebrania danych byłoby ustawienie się ze stoperem i odmierzenie czasu obsługi każdego klienta. Taka metoda nie doprowadziłaby nas jednak do pomyślnego rezultatu, ponieważ obserwowany kasjer zachowywałby się inaczej niż w przypadku „normalnej” pracy, obserwowani klienci mogliby być nie-reprezentatywni dla ogółu klientów sklepu (np. trafilibyśmy na grupę młodzieży), zbieralibyśmy dane w szczególnym dniu czy o szczególnej porze. Aby uniknąć powyższych problemów, należy skorzystać z istniejących baz danych oraz jednocześnie zbierać dane o całym zbiorze atrybutów, takich jak np.: czas obsługi przy kasie, data i godzina obserwacji, liczba kupowanych produktów, płeć i wiek klienta, numer kasjera, nieoczekiwane zdarzenia, np. awaria kasy. Ponadto dane powinny być zbierane nie w trakcie jednego okresu, ale w różnych momentach (w różnych dniach, miesiącach i w różnych porach doby).

2.2. METODA REPREZENTACYJNA

Metoda reprezentacyjna zajmuje się metodami wnioskowania o całej zbiorowości (populacji) pewnego typu elementów na podstawie zbadania pewnej jej części zwanej *próbą*. Polega ona na losowym doborze próby (reprezentacji) ze skończonej zbiorowości generalnej, opisie tej próby za pomocą charakterystyk statystycznych, a następnie na uogólnieniu otrzymanych wyników na zbiorowość generalną, z której próba ta pochodzi, [Steczkowski 1995]. Charakterystyczna dla metody reprezentacyjnej jest skończona liczba elementów zbiorowości.

Elementy zbiorowości wybierane są do próby w sposób systematyczny (np. co dziesiąty pacjent poradni internistycznej) – mówimy wtedy o wyborze *systematycznym* albo (najczęściej) – *w procesie losowania*. *Losowanie* oznacza taki sposób wyboru elementów, w którym jedynie przypadek decyduje o tym, który element zbiorowości wchodzi do próby, a który nie. Próba otrzymana w wyniku losowania nazywa się *próbą losową*. Próba może być *próbą reprezentatywną*, jeżeli jej struktura pod względem badanej cechy nie różni się istotnie od struktury zbiorowości.

2.2.1. PODSTAWOWE SCHEMATY LOSOWANIA

Sposoby wyboru elementów do próby można sklasyfikować ze względu na kilka kryteriów:

1. *Liczbę losowanych elementów* – wyróżniamy **losowanie indywidualne**, jeżeli oddzielnie losujemy do próby poszczególne elementy zbiorowości; przykład: losowanie poszczególnych klientów oraz **losowanie zespołowe**, jeżeli jednostkami losowania są pewne naturalne zespoły elementów zbiorowości; przykład: losowanie ustalonej liczby dni w roku i wybranie do próby wszystkich klientów, którzy robili zakupy w tych dniach.

2. *Zakresy losowania* – wyróżniamy **losowanie warstwowe**, jeżeli elementy losujemy oddzielnie z każdej części, tzw. warstwy zbiorowości, na które została ona podzielona przed losowaniem; przykład: klienci obsługiwani przy poszczególnych kasach oraz **losowanie nieograniczone**, gdy elementy do próby losujemy od razu z całości populacji.

3. *Możliwości wielokrotnego wyboru* – wyróżniamy **losowanie niezależne**, tzw. losowanie ze zwracaniem, jeśli po wylosowaniu każdego elementu może on być brany pod uwagę podczas dalszego losowania; przykład: z zestawu kartek ponumerowanych od 1 do 10 000, oznaczających poszczególnych klientów losujemy jedną kartkę i po zanotowaniu wylosowanego numeru włączamy ją z powrotem do zestawu oraz **losowanie zależne**, tzw. losowanie bez zwracania, jeżeli po wylosowaniu element nie bierze więcej udziału w losowaniu. W tym losowaniu prawdopodobieństwo trafienia na każdy następny element jest uzależnione od dotychczasowych wyników wyboru.

W pierwszym ciągnięciu mamy N możliwych wyborów, w drugim już tylko $N - 1$, potem $N - 2$ itd.

Wyboru schematu losowania dokonuje się, biorąc pod uwagę tzw. *efektywność*, czyli wielkość wariancji estymatorów, *koszt badania* oraz możliwości *praktycznego przeprowadzenia losowania*. W przypadku skończonej populacji wygodniejszy oraz efektywniejszy jest np. schemat losowania zależnego oraz losowanie warstwowe. Losowanie zespołowe jest natomiast najczęściej tańsze i łatwiejsze do przeprowadzenia niż losowanie indywidualne.

Po wybraniu schematu losowania należy określić sposób przeprowadzenia właściwego losowania. Możemy „losować” w znaczeniu potocznym, np. poprzez wyciąganie ponumerowanych kartek z kapelusza lub za pomocą tablic losowych, czy generatorów liczb losowych.

2.2.2. ETAPY STOSOWANIA METODY REPREZENTACYJNEJ

Określenie badanej zbiorowości

1. Należy sprecyzować cel prowadzonego badania statystycznego, a następnie – wyróżnić w badanej zbiorowości te obiekty (*jednostki badania*), które będą istotne ze względu na przyjęty cel analizy. Jeżeli celem naszej analizy będzie ocena sprawności obsługi klientów przy stanowiskach kasowych, to jednostkami badania będą klienci podchodzący do kas.

2. Należy określić cechy zgłoszeń, interesujące ze względu na przyjęty cel analizy (czas obsługi przy kasie).

3. Należy sprecyzować horyzont czasowy badania (np. rok 2004), a następnie określić liczebność zbiorowości (założmy, na podstawie danych historycznych z 2004 roku, że było to 20 000 klientów).

4. W zależności od celu badania możemy określić kilka warstw zbiorowości: np. możemy oddzielnie analizować dni robocze, soboty i niedziele lub np. klientów „zwykłych” i „ekspresowych”. W każdym przypadku musimy określić liczebność elementów w warstwie.

Określenie parametrów badania i schematu losowania

Wyniki losowania będą przedstawione w postaci pewnych wskaźników, zwanych *parametrami badania*, opisujących w sposób statystyczny badaną zbiorowość. Parametrami badania mogą być:

- liczba elementów zbiorowości mających określone własności, np. liczba klientów „zwykłych” w dni świąteczne,
- frakcja (udział procentowy) elementów mających określoną własność, np. procentowy udział klientów kupujących więcej niż 10 towarów w liczbie wszystkich klientów,

- średnia wartość badanej cechy, np. średni czas pobytu przy kasie,
- globalna wartość badanej cechy, np. łączna wartość zakupionych towarów w danym dniu,
- rozkłady zmienności – rozkłady prawdopodobieństwa, np. rozkład czasu obsługi klientów.

2. Po ustaleniu parametrów badania należy ustalić te *cechy elementów* zbiorowości, których analiza umożliwi określenie wybranych parametrów badania. Jeżeli parametrem badania będzie średni czas pobytu przy kasie, to analizowanymi cechami zbiorowości będzie czas „rozpoczęcia obsługi” i czas „zakończenia obsługi”.

3. Kolejną czynnością jest wybór schematu losowania. W badaniach reprezentacyjnych najczęściej wybiera się losowanie zależne ze względu na jego korzystne własności statystyczne. Wybór losowania indywidualnego lub zespołowego jest uzależniony od wielu czynników, m.in. od charakteru analizowanej zbiorowości. Można pokazać, że jeśli ugrupowanie elementów zbiorowości w zespoły jest takie, że wewnątrz zespołów występuje duże zróżnicowanie elementów pod względem badanej cechy, to losowanie zespołowe jest lepsze. Gdy ugrupowanie elementów zbiorowości w zespoły jest losowe (przypadkowe), wówczas losowanie zespołowe jest przeciętnie tak samo efektywne jak losowanie indywidualne, a przy tym z reguły łatwiejsze. W rozpatrywanym przykładzie ustalamy zastosowanie zespołowego losowania zależnego: zamiast losować spośród 20 tys. klientów obsłużonych w ciągu roku, będziemy losować określoną liczbę dni i włączymy do próby wszystkich klientów przyjętych w tych dniach.

Jeżeli podjęliśmy decyzję o uwzględnieniu warstw, to losowanie przeprowadzać będziemy oddzielnie w każdej warstwie, w przeciwnym razie elementy do próby będziemy pobierali z całej populacji.

4. Kolejne czynności to:

- określenie precyzji oszacowania, tj. maksymalnego dopuszczalnego błędu oszacowania interesującego nas parametru; precyzja oszacowania może być wyrażona w procentach (np. dopuszczamy nie więcej niż 5% błędu) lub w jednostkach naturalnych (np. dopuszczalny błąd oszacowania średniego czasu obsługi przy kasie nie może być większy niż 2 minuty),
- określenie poziomu ufności, tj. akceptowanego przez nas prawdopodobieństwa niepełnienia błędu. Zazwyczaj jest to 0,9; 0,95 lub 0,99.

Badanie wstępne

Przed przystąpieniem do właściwego badania statystycznego w niektórych przypadkach wykonuje się tzw. *badanie wstępne*. Polega ono na wybraniu ograniczonej liczby elementów zbiorowości, zbadaniu ich ze względu na interesującą nas cechę i określeniu wartości parametrów niezbędnych do wyznaczenia liczebności badania właściwego (średniej i wariancji).

Jeśli się przyjmie, że liczebność populacji generalnej wynosi N , próba wstępna zawiera NW elementów, a symbol y_i oznacza wartość cechy i -tego elementu (np. czas

obsługi i -tego klienta przy kasie), to analiza wyników badania wstępnego będzie obejmowała:

– obliczenie średniej wartości \bar{y} cechy wszystkich badanych elementów w próbie wstępnej:

$$\bar{y} = \frac{1}{NW} \sum_{i=1}^{NW} y_i \quad (2.1)$$

– obliczanie wartości s^2 badanej cechy w próbie (tzw. estymatora wariancji), tj. współczynnika odchylenia wartości y_i cechy dla poszczególnych elementów od obliczonej średniej \bar{y} :

$$s^2 = \frac{1}{NW - 1} \sum_{i=1}^{NW} (y_i - \bar{y})^2 \quad (2.2)$$

Zaplanowanie i przeprowadzenie badania zasadniczego

Zaplanowanie badania zasadniczego obejmuje wykonanie wymienionych etapów procedury obliczania liczebności próby właściwej, a następnie określenie tzw. operatu losowania i przeprowadzenie losowania.

Wzory określające minimalną wielkość próby w losowaniu zasadniczym różnią się w zależności od przyjętego schematu losowania i parametrów badania. Poniżej przedstawiono trzy wzory, odpowiadające sytuacjom najczęściej występującym w praktyce i dotyczącym losowania zależnego.

Liczebność próby dla losowania indywidualnego (LMI) przy szacowaniu średniej wartości cechy uniwersalnej:

$$LMI = \frac{N}{1 + \frac{Nd^2}{u_\alpha^2 s^2}} \quad (2.3)$$

gdzie: N – liczba elementów badanej zbiorowości (liczebność populacji generalnej), d – precyzja oszacowania (maksymalny błąd w jednostkach naturalnych), u_α – odczytana wartość dla przyjmowanego współczynnika ufności $1 - \alpha$ z tablicy rozkładu $N(0,1)$. Gdy nie mamy bardziej szczegółowych informacji o sposobie kształtowania się cechy badanej w całej zbiorowości lub gdy rozkład populacji nie jest zbliżony do normalnego, można przyjąć, że wartość współczynnika $u_\alpha = 3$, korzystając z reguły 3σ ($u_\alpha^2 = 9$), s^2 – wariancja w próbie cechy badanej (znana lub oszacowana wg wzoru).

Liczebność próby dla losowania zespołowego (LMZ) przy szacowaniu średniej wartości cechy uniwersalnej:

$$LMZ = \frac{NZ}{1 + \frac{NZd^2M^2}{SZ^2u_\alpha^2}} \quad (2.4)$$

gdzie: NZ – liczebność zespołów, tj. jednostek losowania, u_α , d – jak we wzorze na LMI , M – przeciętna liczba elementów zbiorowości (jednostek badania) w jednym zespole (jednostce losowania) obliczona na podstawie badania wstępnego w następujący sposób:

$$M = \frac{1}{NW} \sum_{i=1}^{NW} m_i \quad (2.5)$$

gdzie: m_i – liczba elementów zbiorowości w każdym spośród NW zespołów wylosowanych do próby wstępnej, SZ^2 – wariancja cechy badanej, znana lub oszacowana na podstawie próby wstępnej w następujący sposób:

$$SZ^2 = \frac{1}{NW} \sum (\hat{y}_i - m_i \bar{y})^2 \quad (2.6)$$

przy czym:

$$\hat{y}_i = m_i \bar{y}_i \quad (2.7)$$

to globalna wartość cechy y dla elementów należących do i -tego zespołu (m_i – liczebność i -tego zespołu, \bar{y}_i – średnia wartość cechy y dla elementów i -tego zespołu)

$$\bar{y} = \frac{1}{m} \sum_{i=1}^{NW} y_i \quad (2.8)$$

to średnia wartość cechy dla całej próby wstępnej (m – łączna liczba elementów zbiorowości we wszystkich zespołach w próbie wstępnej).

Liczebność próby dla losowania indywidualnego przy szacowaniu frakcji (udziału procentowego) elementów określonego typu (LFI):

$$LFI = \frac{N}{1 + \frac{d^2(N-1)}{u_\alpha^2 F}} \quad (2.9)$$

gdzie: N , d , u_α – znaczenie jak powyżej, F – współczynnik równy $1/4$, jeśli nie mamy dokładniejszych informacji o sposobie kształtowania się badanej cechy (frakcji) w całej zbiorowości, natomiast $F = p(1-p)$, jeśli można się spodziewać, że rząd wielkości szacowanej frakcji w zbiorowości wynosi p . Wartość p można przyjąć także

Jeżeli można założyć, że rozkład badanej cechy w całej zbiorowości jest zbliżony do rozkładu normalnego, to u_α jest wartością odczytywaną dla przyjętego poziomu ufności z tablic rozkładu $N(0, 1)$. Dla najczęściej przyjmowanych wartości poziomu ufności, wartości u_α i u_α^2 wynoszą:

- poziom ufności 0,9 $u_\alpha = 1,645$ $u_\alpha^2 = 2,71$,
- poziom ufności 0,95 $u_\alpha = 1,96$ $u_\alpha^2 = 3,84$,
- poziom ufności 0,99 $u_\alpha = 2,58$ $u_\alpha^2 = 6,66$.

Przykład: chcemy oszacować średni czas obsługi klientów przy kasie. Wielkość próby określimy za pomocą losowania zależnego (bez zwracania), zespołowego (wzór (2.4)). Przyjmijmy, że $d = 1$ minuta (dopuszczalny błąd szacowania średniego czasu oczekiwania), $u_\alpha^2 = 9$ (ze względu na to, że nie mamy dokładniejszych informacji o sposobie kształtowania się badanej cechy w całej zbiorowości). Obliczamy pozostałe parametry. Dla uproszczenia rozważań założmy, że w wyniku badania wstępnego liczącego $NZ = 50$ zespołów (dni) otrzymaliśmy następujące wyniki:

$M = 48$ klientów (przeciętna liczba klientów w jednym dniu w wylosowanych 50 dniach w próbie wstępnej),

$$\bar{y} = 32 \text{ minuty,}$$

$$SZ^2 = 32\,400.$$

Zgodnie ze wzorem $LMZ = 93,98$. Oznacza to, że próba zasadnicza powinna liczyć co najmniej 94 zespoły (dni).

Po ustaleniu liczebności próby należy obliczyć całkowity koszt przeprowadzenia badania. Jeżeli koszt ten okaże się zbyt duży, to liczebność próby musimy dopasować do możliwości finansowych. Należy wtedy obliczyć, jaką precyzję szacunku osiągniemy, badając taką próbę. Jeśli precyzja (błąd oszacowania) okaże się zbyt mała, to albo zwiększamy fundusze, albo ograniczamy zakres badania.

W następnej kolejności sporządza się **operat losowania**, tj. wykaz wszystkich jednostek losowania, uporządkowanych i ponumerowanych. Ostatnim elementem przygotowań jest przeprowadzenie losowania (por. rys. 2.1).

Aby wylosować próbę z danej zbiorowości generalnej, można posłużyć się tablicami liczb losowych lub generatorami. Tablice zawierają cyfry podzielone na bloki, które z kolei dzielą się na wiersze oraz kolumny (rys. 2.1, część B). Po sporządzeniu operatu losowania (rys. 2.1, część A), wyznacza się w sposób przypadkowy miejsce w tablicy (rys. 2.1, część B) i rozpoczyna się odczytywanie numerów. Przesuwając się wzdłuż wierszy lub kolumn, odczytuje się n kolejnych numerów, ale tylko takich, które spełniają warunek $n \leq N$, gdzie N to liczba wszystkich jednostek losowania, a n to liczebność próby (rys. 2.1, część C).

2.3. WYKORZYSTANIE DANYCH WEJŚCIOWYCH W MODELU SYMULACYJNYM

Zebrane dane wejściowe (po wstępnym przetworzeniu) mogą być wykorzystane w modelu symulacyjnym na jeden z trzech sposobów.

Dane są bezpośrednio używane w trakcie symulacji. Na przykład, jeżeli zebrane dane wejściowe określają czas obsługi klienta na stanowisku obsługi, to poszczególne wartości są pobierane (np. z zewnętrznej bazy danych) za każdym razem, kiedy model symulacyjny żąda informacji o czasie obsługi klienta.

Na podstawie zebranych danych formułuje się *rozkład empiryczny*, z którego pobiera się wartości zawsze wtedy, kiedy jest to niezbędne w procesie symulacji.

Za pomocą metod statystycznych dopasowuje się *teoretyczne rozkłady statystyczne* do zebranych danych, a następnie testami statystycznymi sprawdza się poprawność postawionych hipotez. Jeżeli rozkład teoretyczny dostatecznie dobrze opisuje zebrane dane, to w procesie symulacji wykorzystuje się wartości naliczane za pomocą tych rozkładów.

Podejście 1: wadą tego podejścia jest to, że symulacja może jedynie odtwarzać to, co już było; zwykle dysponujemy zbyt małą liczbą danych, aby w pełni przeprowadzić proces symulacji; nie jesteśmy w stanie prognozować zachowania się systemu.

Podejście 2: nie ma wad podejścia 1, ale rozkład empiryczny może zawierać w sobie pewne nieregularności; nie jesteśmy w stanie generować danych spoza zakresu rozkładu empirycznego (jest to znaczące utrudnienie, ponieważ modele symulacyjne budowane są często w celu badania zachowania się systemu w warunkach ekstremalnych); rozkład empiryczny wymaga dużych zasobów pamięci, ponieważ jeżeli niezbędnych będzie nam n wartości w trakcie symulacji – musimy wprowadzić do pamięci $2n$ pozycji (wartości zmiennej losowej i wartości prawdopodobieństw/dystrybuanty).

Podejście 3: jest zdecydowanie najlepsze, ponieważ rozkład teoretyczny „wygładza” nierówności rozkładu empirycznego; możliwe jest generowanie wartości ekstremalnych; przy zastosowaniu tego podejścia nie obciążamy zasobów pamięciowych komputera.

Opis sposobów wykorzystania danych wejściowych w modelu symulacyjnym wyraźnie wskazuje na wybór podejścia trzeciego. Zdarzają się jednak sytuacje, kiedy nie jesteśmy w stanie dopasować rozkładu teoretycznego do zebranych danych (por. podrozdz. 2.8). Zaleca się wtedy budowę rozkładu empirycznego.

2.4. CIĄGŁE I DYSKRETNE ROZKŁADY ZMIENNEJ LOSOWEJ

Załóżmy, że planujemy przedsięwzięcie polegające na uruchomieniu produkcji pewnego produktu (np. kartek walentynkowych). Powodzenie zależy w dużej mierze od popytu, którego nie można dokładnie przewidzieć. Popyt na kartki może być podobny do zaobserwowanego w zeszłym roku, może go jednak przewyższać lub kształtować się poniżej zeszłorocznych wartości. Za pomocą analizy danych historycznych jesteśmy w stanie określić *prawdopodobieństwo* kształtowania się popytu na poziomie niższym, równym lub wyższym od zeszłorocznego, ale nie możemy wskazać tej jednej jedynej wartości, która odzwierciedlać będzie rzeczywisty przyszły poziom popytu na kartki walentynkowe.

W budowanym modelu matematycznym popyt na kartki walentynkowe możemy opisać za pomocą *zmiennej losowej*. *Zmienną losową* definiujemy jako (por. [Aczel 2005]) *zmienną, która przyjmuje różne wartości liczbowe, wyznaczone przez los*. Przyporządkowanie prawdopodobieństwa różnym możliwym wartościom zmiennej losowej nazywa się natomiast *rozkładem prawdopodobieństwa zmiennej losowej*. Zmienna losowa może być skokowa (dyskretna) lub ciągła. Zmienna losowa dyskretna przyjmuje wartości ze zbioru przeliczalnego, natomiast zmienna losowa ciągła może przyjmować wartości z dowolnego przedziału liczbowego.

Ciągłe rozkłady prawdopodobieństwa opisuje się, podając funkcję gęstości oraz parametryzując ją. Ogólnie, parametry funkcji gęstości klasyfikuje się jako:

- *parametry położenia* definiujące punkt na osi X ; zmiana tego punktu powoduje przesunięcie rozkładu w prawo lub w lewo,
- *parametry skali* określające „wielkość” rozkładu; zmiana tego parametru powoduje ekspansję wykresu bez zmiany jego podstawowego kształtu,
- *parametry kształtu* definiujące kształt charakterystyki.

Dyskretne rozkłady losowe opisuje się za pomocą funkcji prawdopodobieństwa (odpowiednik funkcji gęstości rozkładów ciągłych), zwanej również funkcją rozkładu masy prawdopodobieństwa oraz za pomocą funkcji dystrybuanty.

2.5. PARAMETRIZACJA PODSTAWOWYCH ROZKŁADÓW CIĄGŁYCH

2.5.1. ROZKŁAD JEDNOSTAJNY (RÓWNOMIERNY, PROSTOKĄTNY)

Jest to zwykle pierwszy rozkład, który przyjmuje się w badaniach symulacyjnych wszędzie tam, gdzie podejrzewa się losowość zdarzeń w pewnym przedziale wartości

od a do b (rys. 2.2). Jest to także podstawowy rozkład stosowany do generowania wartości losowych z innych rozkładów.

Funkcja gęstości:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{dla } a \leq x \leq b \\ 0 & \text{w p.p.} \end{cases} \quad (2.11)$$

Dystrybuanta:

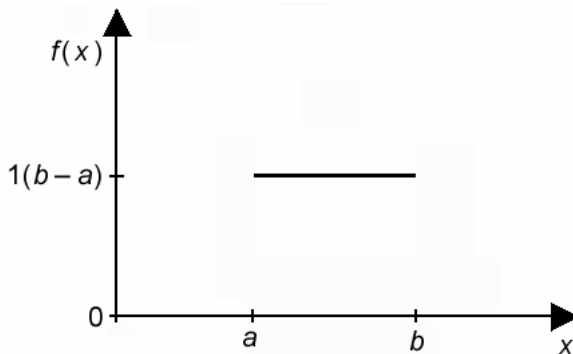
$$F(x) = \begin{cases} 0 & \text{dla } x < a \\ \frac{x-a}{b-a} & \text{dla } a \leq x \leq b \\ 1 & \text{dla } b < x \end{cases} \quad (2.12)$$

Parametry a i b to liczby rzeczywiste, $a < b$, a jest parametrem położenia, natomiast $a - b$ to parametr skali.

Zakres: $[a, b]$

Wartość oczekiwana: $\frac{a+b}{2}$

Wariancja: $\frac{(b-a)^2}{12}$



Rys. 2.2. Funkcja gęstości rozkładu jednostajnego

2.5.2. ROZKŁAD WYKŁADNICZY

Za pomocą rozkładu wykładniczego w symulacji dyskretnej opisuje się czas, jaki upływa między nadejściem kolejnych klientów do systemu, przy założeniu, że zgłoszenia są całkowicie przypadkowe (mogą nastąpić w dowolnej chwili) oraz zachowa-

ne jest średnie natężenie zgłoszeń. Oznacza to przyjęcie założenia, że czas nadejścia zgłoszenia jest niezależny od poprzednich zgłoszeń, a prawdopodobieństwo zgłoszenia w przedziale dt jest proporcjonalne do długości tego przedziału.

Rozkład wykładniczy jest często używany w teorii niezawodności do opisu czasu pracy elementów niestarczających się. Jeżeli T jest czasem pracy takiego elementu, to $T = 1/\lambda$ jest średnim czasem pracy, a parametr $\lambda = 1/T$ nazywamy intensywnością uszkodzeń. Niestarzenie się elementu oznacza, że prawdopodobieństwo uszkodzenia się elementu nie zależy od dotychczas przepracowanego czasu.

Funkcja gęstości:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{dla } x \geq 0 \\ 0 & \text{wp.p.} \end{cases} \quad (2.13)$$

Dystrybuanta:

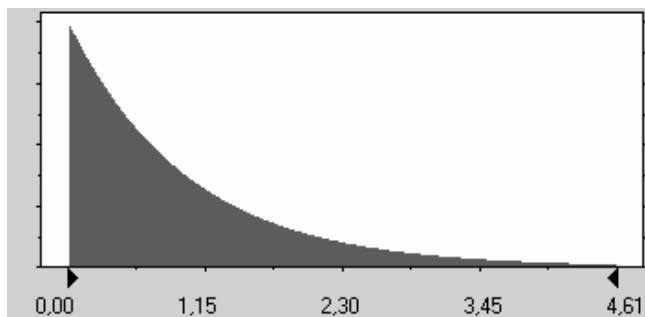
$$F(x) = \begin{cases} 1 - e^{-\lambda x} & \text{dla } x \geq 0 \\ 0 & \text{w p.p.} \end{cases} \quad (2.14)$$

Parametry: parametr położenia $1/\lambda > 0$

Zakres: $[0, \infty]$

Wartość oczekiwana: $1/\lambda$

Wariancja: $(1/\lambda)^2$



Rys. 2.3. Funkcja gęstości rozkładu wykładniczego

2.5.3. ROZKŁAD NORMALNY

Rozkład normalny dobrze opisuje błędy i zakłócenia różnorodnych typów. Stosuje się go do odzwierciedlenia procesów charakteryzujących się symetrią. Ze względu na zakres określoności $(-\infty, \infty)$ należy rozważnie stosować rozkład normalny do przedstawiania takich zjawisk, które opisywane są wyłącznie wartościami dodatnimi (np. czas realizacji procesu).

Funkcja gęstości:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \text{ dla wszystkich } x \in R \quad (2.15)$$

Dystrybuanta: brak jednolitej postaci

Parametry:

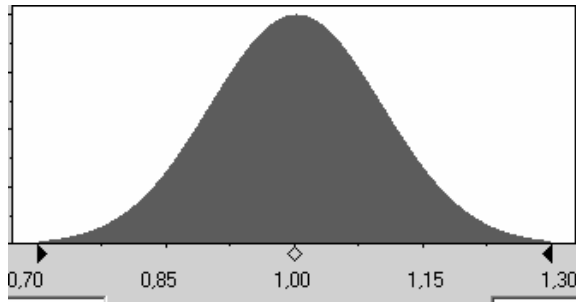
parametr położenia $\mu \in (-\infty, \infty)$

parametr skali $\sigma > 0$

Zakres: $(-\infty, \infty)$

Wartość oczekiwana: μ

Wariancja: σ^2



Rys. 2.4. Funkcja gęstości rozkładu normalnego

2.5.4. ROZKŁAD TRÓJKĄTNY

Rozkład trójkątny stosowany jest jako przybliżony rozkład w przypadku braku danych, w sytuacjach, kiedy potrafimy oszacować wartość minimalną, maksymalną i najczęściej występującą. Decydenci pytani o losowe zachowanie danego parametru rzadko powołują się na konkretny rozkład losowy, natomiast często są w stanie przytoczyć właśnie te trzy wartości.

Funkcja gęstości:

$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)} & \text{dla } a \leq x \leq c \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{dla } c < x \leq b \\ 0 & \text{w p.p.} \end{cases} \quad (2.16)$$

Dystrybuanta:

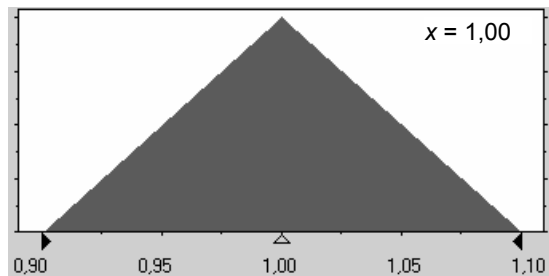
$$F(x) = \begin{cases} 0 & \text{dla } x < a \\ \frac{(x-a)^2}{(b-a)(c-a)} & \text{dla } a \leq x \leq c \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & \text{dla } c < x \leq b \\ 1 & \text{dla } b < x \end{cases} \quad (2.17)$$

Parametry: a , b i c należą do R oraz $a < c < b$; a to parametr położenia, $b - a$ to parametr skali, c to parametr kształtu.

Zakres: $[a, b]$

Wartość oczekiwana: $\frac{a+b+c}{3}$

Wariancja: $\frac{a^2 + b^2 + c^2 - ab - ac - bc}{18}$



Rys. 2.5. Funkcja gęstości rozkładu trójkątnego

2.5.5. INNE ROZKŁADY CIĄGŁE

Rozkład gamma stosuje się często do opisu czasu zakończenia zadań, np. czasu obsługi na stanowisku obsługi lub czasu naprawy.

Uogólnieniem rozkładu wykładniczego jest rozkład Weibulla, który ma liczne zastosowania w teorii niezawodności, np. do opisu czasu bezawaryjnej pracy urządzeń. Jeżeli system składa się z wielu części, które psują się niezależnie od siebie, i jeżeli awaria dowolnej z tych części powoduje awarię całego urządzenia, to czas bezawaryjnej pracy urządzenia można przybliżyć rozkładem Weibulla.

Rozkład logarytmiczno-normalny służy do opisu czasów zakończenia zadań o rozkładach prawoskośnych. Rozkład ten jest ściśle związany z rozkładem normalnym. Jeżeli zmienna losowa X ma rozkład lognormalny (μ, σ) , to $\ln(X)$ ma rozkład normalny (μ, σ) .

Rozkład logarytmiczno-normalny ma duże znaczenie w modelach finansowych. Słynna formuła Blacka–Scholesa wyceny opcji zakłada, że logarytmy stóp zwrotu z akcji będącej przedmiotem opcji mają rozkład normalny. Symulacja Monte Carlo, która jest stosowana obok innych metod ilościowych do wyceny opcji (por. [Jackson i Staunton 2004]), musi się do tego rozkładu odwoływać.

Szczególnym rozkładem jest rozkład beta. Ze względu na różnorodność kształtów, które może przyjmować, jest on stosowany jako przybliżony rozkład w przypadku braku danych. Rozkład beta jest też często wykorzystywany do opisu losowych proporcji, np. udziału części wadliwych w całej partii towaru.

2.6. PARAMETRYZACJA PODSTAWOWYCH ROZKŁADÓW DYSKRETYCH

2.6.1. ROZKŁAD DWUPUNKTOWY

Z rozkładem dwupunktowym mamy do czynienia, gdy zmienna losowa może przyjąć jedną z dwóch możliwych wartości (np. jednorazowy rzut monetą). Może on też służyć jako podstawowy rozkład do generowania innych rozkładów dyskretnych.

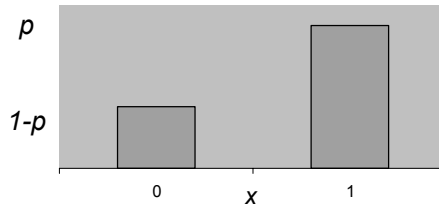
Funkcja prawdopodobieństwa:

$$p(x) = \begin{cases} 1-p & \text{dla } x=0 \\ p & \text{dla } x=1 \\ 0 & \text{w pozostałych przypadkach} \end{cases} \quad (2.18)$$

Dystrybuanta:

$$F(x) = \begin{cases} 0 & \text{dla } x < 0 \\ 1-p & \text{dla } 0 \leq x < 1 \\ 1 & \text{dla } x \geq 1 \end{cases} \quad (2.19)$$

Parametr: $p \in (0; 1)$
 Zakres: $\{0; 1\}$
 Średnia: p
 Wariancja: $p(1-p)$



Rys. 2.6. Funkcja prawdopodobieństwa rozkładu dwupunktowego

Zmienna losowa rozkładu dwupunktowego może być traktowana jako wynik eksperymentu, który albo kończy się powodzeniem, albo niepowodzeniem. Jeżeli prawdopodobieństwo sukcesu oznaczymy jako p , to zmienna losowa $X = 0$, jeżeli eksperyment kończy się niepowodzeniem oraz zmienna losowa $X = 1$, gdy eksperyment kończy się sukcesem.

2.6.2. ROZKŁAD DYSKRETNY JEDNOSTAJNY

Rozkład jednostajny dyskretny opisuje losowe wystąpienie kilku możliwych wartości, przy czym prawdopodobieństwo wystąpienia tych wartości jest takie samo. Jest on używany w sytuacjach, kiedy i wartości jest wyróżnianych spośród j .

Funkcja prawdopodobieństwa:

$$p(x) = \begin{cases} \frac{1}{j-i+1} & \text{dla } x \in \{i, i+1, \dots, j\} \\ 0 & \text{w p.p.} \end{cases} \quad (2.20)$$

Dystrybuanta:

$$F(x) = \begin{cases} 0 & \text{dla } x < i \\ \frac{\lfloor x \rfloor - i + 1}{j - i + 1} & \text{dla } i \leq x \leq j \\ 1 & \text{dla } x > j \end{cases} \quad (2.21)$$

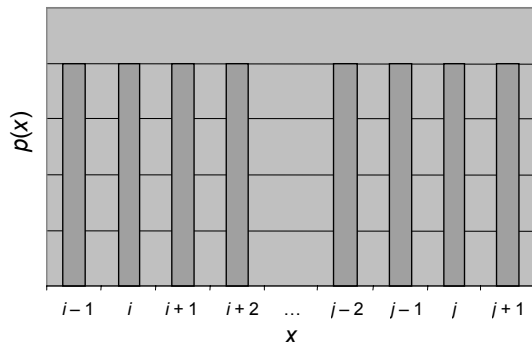
gdzie: $\lfloor x \rfloor$ oznacza największą liczbę całkowitą $\leq x$.

Parametry: liczby całkowite i, j , gdzie $i \leq j$.

Zakres: $\{i, i+1, \dots, j\}$

Średnia: $\frac{i+j}{2}$

Wariancja $\frac{(j-i+1)^2 - 1}{12}$



Rys. 2.7. Funkcja prawdopodobieństwa rozkładu jednostajnego dyskretnego

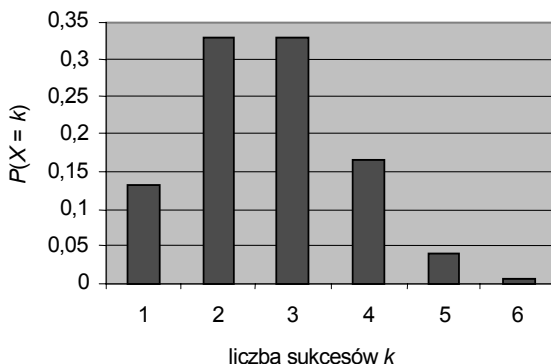
2.6.3. ROZKŁAD DWUMIANOWY

Rozkład ten stosujemy, aby określić liczbę sukcesów w serii niezależnych eksperymentów. Wynik każdego eksperymentu (próby) może być tylko sukcesem lub porażką, próby są niezależne, a prawdopodobieństwo sukcesu jest stałe w trakcie eksperymentu. Przykładowo, za pomocą tego rozkładu możemy określić liczbę błędnych produktów w serii. Aby poprawnie opisać rozkład, potrzebujemy dwóch informacji: o liczbie prób oraz prawdopodobieństwie odniesienia sukcesu w każdej próbie.

Funkcja prawdopodobieństwa:

$$P(X = k) = C_n^k p^k q^{n-k} \quad (2.22)$$

gdzie: k – liczba sukcesów, n – liczba wykonanych prób, p – prawdopodobieństwo sukcesu w jednej próbie, q – prawdopodobieństwo porażki w jednej próbie.



Rys. 2.8. Funkcja prawdopodobieństwa rozkładu dwumianowego

2.6.4. ROZKŁAD GEOMETRYCZNY

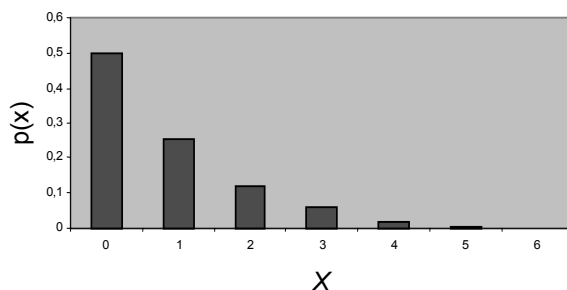
Rozkład geometryczny może być zastosowany do wyznaczenia: liczby produktów przebadanych do momentu pojawienia się pierwszego produktu uszkodzonego, liczby elementów w grupie o losowej liczebności, liczby produktów uszkodzonych, które pojawiają się przed pierwszym poprawnym w serii niezależnych prób Bernoulliego.

Funkcja prawdopodobieństwa:

$$p(x) = \begin{cases} p(1-p)^x & \text{dla } x \in \{0, 1, \dots\} \\ 0 & \text{w p.p.} \end{cases} \quad (2.23)$$

Dystrybuanta:

$$F(x) = \begin{cases} 1 - p(1-p)^{|x|+1} & \text{dla } x \geq 0 \\ 0 & \text{w p.p.} \end{cases} \quad (2.24)$$



Rys. 2.9. Funkcja prawdopodobieństwa rozkładu geometrycznego, $p = 0,5$

Parametr: $p \in (0, 1)$

Zakres: $\{0, 1, \dots\}$

Średnia: $\frac{1-p}{p}$

Wariancja: $\frac{1-p}{p^2}$

2.6.5. ROZKŁAD POISSONA

Rozkład Poissona dobrze opisuje liczbę zgłoszeń napływających w pewnym czasie, przy czym tempo napływania zgłoszeń jest stałe. Prawdziwe jest stwierdzenie, że jeżeli czas między napływającymi zgłoszeniami ma rozkład wykładniczy, to liczba zgłoszeń pojawiających się w jednostce czasu ma rozkład Poissona.

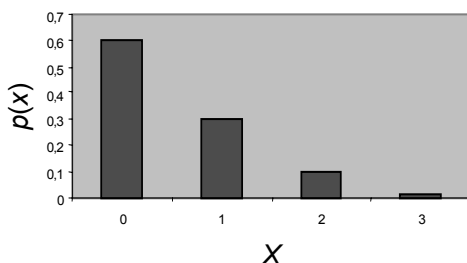
Funkcja prawdopodobieństwa:

$$p(x) = \begin{cases} \frac{e^{-\lambda} \lambda^x}{x!} & \text{dla } x \in \{0, 1, \dots\} \\ 0 & \text{w p.p.} \end{cases} \quad (2.25)$$

Dystrybuanta:

$$F(x) = \begin{cases} 0 & \text{dla } x < 0 \\ e^{-\lambda} \sum_{i=0}^{\lfloor x \rfloor} \frac{\lambda^i}{i!} & \text{dla } 0 \leq x \end{cases} \quad (2.26)$$

Parametr: $\lambda > 0$
 Zakres: $\{0, 1, \dots\}$
 Średnia: λ
 Wariancja: λ



Rys. 2.10. Funkcja prawdopodobieństwa rozkładu Poissona, $\lambda = 0,5$

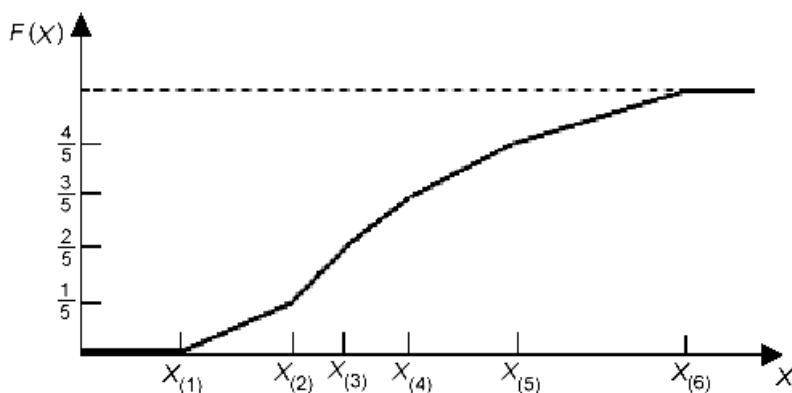
2.7. ROZKŁADY EMPIRYCZNE

Jeżeli zebrane dane służą wprost do zbudowania pewnego rozkładu prawdopodobieństwa, mówimy o *rozkładzie empirycznym*. Z rozkładu empirycznego wartości losowe mogą być generowane w trakcie symulacji **zamiast** z rozkładu teoretycznego. Zdarza się tak najczęściej wtedy, gdy nie możemy dopasować rozkładu teoretycznego do zebranej próbki danych.

Budowa *ciągłego rozkładu* empirycznego (rys. 2.11) przebiega odmiennie w zależności od tego, czy wartości badanej cechy są oryginalne, czy też jedynie znamy liczbę obserwacji wystąpienia danej cechy w wyszczególnionych przedziałach. W pierwszym przypadku możemy zbudować ciągłą odcinkową dystrybuantę F :

- sortujemy wszystkie obserwacje $X_{(i)}$ rosnąco, tak że $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(n)}$,
- budujemy dystrybuantę według wzoru:

$$F(x) = \begin{cases} 0 & x < X_{(1)} \\ \frac{i-1}{n-1} + \frac{x - X_{(i)}}{(n-1)(X_{(i+1)} - X_{(i)})} & X_{(i)} \leq x \leq X_{(i+1)} \text{ dla } i=1, 2, \dots, n-1 \\ 1 & x \geq X_{(n)} \end{cases} \quad (2.27)$$

Rys. 2.11. Przykład krzywej odcinkowej dla $n = 6$ obserwacji

Podstawową wadą rozkładu empirycznego jest to, że nie możemy generować wartości mniejszych od $X_{(1)}$ ani większych od $X_{(n)}$. Ponadto, rzeczywiste rozkłady ciągłe są najczęściej przesunięte i mają szerokie końce („długie ogony”). W trakcie budowy rozkładu empirycznego, gdy próbka danych nie jest zbyt liczna, możemy nie zaobserwować wartości przypadających na „długi ogon”.

Tabela 2.1. Przykład dystrybuanty empirycznej dla rozkładu dyskretnego

Liczba zdarzeń x	Prawdopodobieństwo $P(x)$	Dystrybuanta empiryczna y
0	0	0
1	0,10	0,10
2	0,51	0,61
3	0,19	0,80
4	0,15	0,95
5	0,05	1,0

Jeżeli nie dysponujemy oryginalnymi wartościami badanej cechy, możemy posłużyć się *dyskretną dystrybuantą empiryczną*, która jest definiowana poprzez określenie częstości skumulowanych. Załóżmy, że należy wygenerować zmienną losową określającą liczbę towarów zakupionych przez klienta w domu handlowym. Funkcja prawdopodobieństwa zadana jest rozkładem jak w tab. 2.1.

Po wygenerowaniu liczby losowej u jej wartość porównywana jest z wartością y . Jeżeli mieści się ona w przedziale $y_i < u \leq y_{i+1}$, $i = 0, 1, \dots, 4$, to jako wynik przyjmuje się wartość wielkości x_{i+1} .

2.8. DOPASOWYWANIE ROZKŁADÓW TEORETYCZNYCH

2.8.1. NIEZALEŻNOŚĆ OBSERWACJI

Większość technik statystycznych stosowanych w trakcie dopasowywania rozkładów opiera się na założeniu niezależności obserwacji. Jeżeli warunek niezależności nie jest zachowany, testy mogą zawieść. Na przykład, chcemy zbudować model symulacyjny systemu sterowania zapasami i zbieramy dane określające dzienną liczbę zamówionych produktów. Budując model symulacyjny, musimy najpierw sprawdzić, czy dzienna liczba zamówień zależy od dnia tygodnia. Jeżeli tak rzeczywiście jest, powinniśmy określić rozkład prawdopodobieństwa zamawianych produktów dla każdego dnia tygodnia oddzielnie.

Jednym z popularniejszych testów badania niezależności statystycznej jest test chi-kwadrat (χ^2). Wymogiem tego testu jest duża liczebność próby, której elementy zostały rozdzielone na dwie kategorie ze względu na obie cechy. Sporządza się odpowiednią tablicę kombinowaną dla dwu cech, zwaną *tablicą niezależności*, która po wypełnieniu daje macierz liczebności empirycznych. Nakłada się na nią macierz liczebności teoretycznych, obliczonych przy założeniu niezależności obu cech. Porównanie elementów macierzy za pomocą statystyki χ^2 daje odpowiedź, czy można odrzucić hipotezę o niezależności cech na skutek wystąpienia zbyt dużych różnic liczebności empirycznych i teoretycznych.

2.8.2. STAWIANIE HIPOTEZY ODNOŚNIE DO RODZINY ROZKŁADÓW. ESTYMACJA PARAMETRÓW ROZKŁADU

Stawiamy hipotezę odnośnie do typu rozkładu (normalny, wykładniczy, Poissona itd.) bez szacowania wielkości parametrów. Możemy w tym celu wykorzystać *posiadaną wiedzę* na temat typu rozkładu pasującego do danej sytuacji. Możemy również stosować *wnioskowanie heurystyczne*:

- szacując – na podstawie danych z próbki – podstawowe parametry statystyczne, takie jak: średnia, mediana, wariancja i inne. Wartości i wzajemne relacje pomiędzy

tymi parametrami mogą zasugerować typ rozkładu (np. dla rozkładów symetrycznych wartości średniej i mediany są prawie identyczne),

- wyznaczając histogramy i dopasowując do nich (wg kształtu) znane wykresy funkcji gęstości (dla rozkładów ciągłych) i funkcje prawdopodobieństwa (dla rozkładów dyskretnych).

Aby określić parametry wskazanego wcześniej rozkładu, korzystamy z działu statystyki zajmującego się estymacją, czyli szacowaniem parametrów rozkładu badanej cechy populacji:

- szacując średnią wartość badanej cechy metodą estymacji punktowej, średniej lub przedziałowej,
- szacując wariancję σ^2 lub odchylenie standardowe σ .

2.8.3. BADANIE POPRAWNOŚCI DOPASOWANIA ROZKŁADU

Kolejny dział statystyki zajmuje się weryfikacją testowanych (stawianych) hipotez statystycznych. Hipotezy statystyczne, będące sformułowanymi przypuszczeniami dotyczącymi rozkładu populacji, mogą mieć różną postać. *Hipotezy parametryczne*, precyzujące wartości parametrów w rozkładzie populacji, należą do najczęściej sprawdzanych hipotez statystycznych. Weryfikacja hipotezy odbywa się przez zastosowanie narzędzia zwanego *testem statystycznym*. Jest to reguła postępowania, która każdej możliwej próbie losowej przyporządkowuje decyzję przyjęcia lub odrzucenia sprawdzanej hipotezy, w zależności od postaci postawionej *hipotezy zerowej* (tzn. bezpośrednio sprawdzanej) oraz postaci *hipotezy alternatywnej* (tzn. konkurencyjnej dla hipotezy zerowej). Istota rzeczy przy budowie testu polega jednak na tym, aby uchronić się zarówno przed popełnieniem *błędu pierwszego rodzaju*, polegającym na odrzuceniu hipotezy prawdziwej, jak i przed popełnieniem *błędu drugiego rodzaju*, polegającym na przyjęciu hipotezy fałszywej.

Przyjęcie lub odrzucenie hipotezy w teście statystycznym nie jest równoznaczne z logicznym udowodnieniem jej prawdziwości lub fałszywości. Należy bowiem pamiętać, że odrzucając sprawdzaną hipotezę, kierujemy się jedynie tym, że dane liczbowe wynikające z badania rzeczywistości w małym stopniu potwierdzają prawdziwość tej hipotezy. Możliwe jednak, że jest na odwrót, tzn. hipoteza jest prawdziwa, a tylko nasze dane liczbowe z próby są błędne lub po prostu mało prawdopodobne przy tej hipotezie.

W ogólnej teorii weryfikacji hipotez statystycznych stosuje się różne rodzaje testów: np. *testy najmocniejsze, nieobciążone, istotności*. Testy istotności to taki rodzaj testów, w których na podstawie wyników próby losowej podejmuje się decyzję odrzucenia hipotezy sprawdzanej lub stwierdza się, że nie ma podstaw do jej odrzucenia. Nie podejmuje się natomiast decyzji o przyjęciu sprawdzanej hipotezy, gdyż bierze się w nim pod uwagę jedynie błąd pierwszego rodzaju, którego prawdopodobieństwo nosi nazwę *poziomu istotności*, a nie uwzględnia się konsekwencji popełnienia błędu drugiego rodzaju.

Testy istotności, w których weryfikowana hipoteza dotycząca rozkładu badanej cechy w populacji generalnej nie precyzuje wartości parametrów tego rozkładu, noszą nazwę *nieparametrycznych testów istotności*. Wśród nich najpowszechniejsze to *testy zgodności*:

- Test zgodności χ^2 (chi kwadrat) pozwala na sprawdzenie, czy populacja ma określony typ rozkładu (tj. określoną postać funkcyjną dystrybuanty). Jedyńm ograniczeniem jest to, że próba musi być duża. Wyniki próby dzielimy na klasy i dla każdej klasy z rozkładu hipotetycznego obliczamy liczebności teoretyczne, które porównujemy z liczebnościami empirycznymi za pomocą odpowiedniej statystyki χ^2 . Gdy rozbieżności między liczebnościami empirycznymi i teoretycznymi są zbyt duże, hipoteza, że populacja ma ten właśnie rozkład teoretyczny, musi być odrzucona.

- Test zgodności Kołmogorowa oparty na statystyce λ . W porównaniu z poprzednim testem ma on liczne ograniczenia. Wymaga bowiem wyników dokładnych, które nie są pogrupowane w klasy oraz dużej próby. Test ten ma zastosowanie tylko dla rozkładów ciągłych, w których sprecyzowano wartości parametrów.

- Test Kołmogorowa–Smirnowa dla weryfikowania hipotezy, że dwie próby pochodzą z tej samej populacji. Test ten dotyczy określenia skumulowanego rozkładu częstości dla danych symulowanych oraz danych rzeczywistych. Traktuje on oddzielnie poszczególne obserwacje i – inaczej niż test χ^2 – nie gubi informacji przez łączenie ze sobą różnych kategorii.

Jednym z rutynowo wyznaczanych wskaźników jakości dopasowania jest tzw. wartość p (*p-value*), która zawiera się w przedziale od 0 do 1. Interpretujemy ją jako prawdopodobieństwo tego, że inny zbiór danych będzie bardziej rozbieżny z wybranym rozkładem niż próba, którą przyjęliśmy do wskazania tego rozkładu, przy założeniu, że H_0 (bezpośrednio sprawdzana hipoteza) jest prawdziwa:

- p małe ($< 0,05$) świadczy przeciwko H_0 , czyli dostarcza silnych dowodów za odrzuceniem H_0 ,

- p duże ($> 0,1$, ale im większe, tym lepiej) świadczy o braku dowodów za odrzuceniem H_0 , czyli pozwala uznać przeprowadzone dopasowanie za poprawne.

Oczywiście, duża wartość p nie stanowi „dowodu” na poprawność dopasowania rozkładu. Świadczy jedynie o tym, że nie ma dowodów nakazujących odrzucenie hipotezy o poprawności dopasowania.

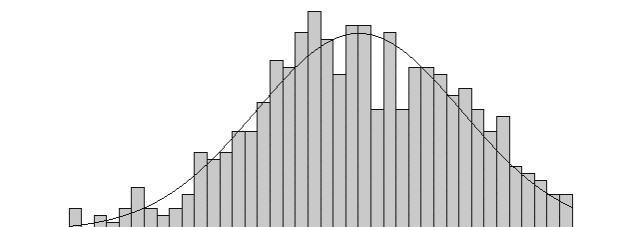
Oceniając poprawność dopasowania rozkładu, nie możemy niestety kierować się rygorystycznymi zaleceniami przesądzającymi o przyjęciu lub odrzuceniu badanej hipotezy. Różne testy statystyczne mogą różnie oceniać poprawność dopasowania wybranego rozkładu do próby danych (por. rys. 2.12 i 2.13). Co więcej, inny sposób przygotowania danych (np. zmieniony podział na klasy) może doprowadzić do decyzji o wyborze innego rozkładu.

Pomyślne wyniki testów statystycznych stawiają nas często w kłopotliwej sytuacji, wskazując kilka „dobrych” rozkładów. Decydując się na wybór jednego z nich, dobrze jest wziąć pod uwagę kilka przesłanek:

Rozkład : Normalny
 Wyrażenie: NORM(26,4; 7,95)
 Błąd kwadratowy: 0,001211

Test chi-kwadrat
 Liczba przedziałów = 29
 Liczba stopni swobody = 26
 Wartość statystyki = 22,1
 Odpowiadająca wartość $p = 0,683$

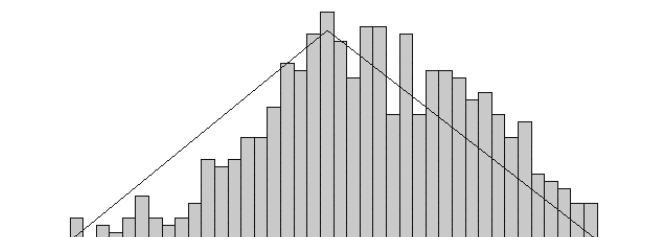
Test Kolmogorowa–Smirnowa
 Wartość statystyki = 0,77
 Odpowiadająca wartość $p < 0,01$



Rys. 2.12. Przykładowy wynik dopasowania rozkładu teoretycznego (normalnego) do zebranych danych

Rozkład: Trójkątny
 Wyrażenie: TRIA(4, 23, 43)
 Błąd kwadratowy: 0,003597

Test chi-kwadrat
 Odpowiadająca wartość $p < 0,005$



Rys. 2.13. Przykładowy wynik dopasowania rozkładu teoretycznego (trójkątnego) do zebranych danych

(a) *Rozkłady ograniczone i nieograniczone.* Rozkładów trójkątnego (ograniczonego) i normalnego (nieograniczonego) używa się zwykle do opisu czasu trwania procesów (por. rys. 2.12 i rys. 2.13). W długich przebiegach symulacyjnych oba rozkłady wygenerują podobne ciągi wartości zmiennej losowej, jednakże w rozkładzie normalnym mogą się czasami pojawiać ekstremalnie duże wartości, które nie występują w systemie rzeczywistym. Rozkład normalny o średniej zbliżonej do zera może również generować sztucznie zawyżoną liczbę zerowych wartości zmiennej losowej – będzie się tak działo w przypadku odwzorowywania wartości nieujemnych (np. czasu obsługi) i automatycznej zamiany ujemnych wartości na wartości zerowe (jak to jest w większości narzędzi do prowadzenia symulacji). Z drugiej jednak strony, użycie rozkładu trójkątnego nie pozwoli nam uzyskać w modelu symulacyjnym wartości ekstremalnych, rzadko występujących w rzeczywistości.

(b) *Dobór parametrów rozkładu.* Jeżeli planujemy zmiany w modelu, polegające na modyfikacji parametrów rozkładu (np. w celu przeprowadzenia analizy wrażliwości), wygodniej będzie zastosować rozkład o intuicyjnie zrozumiałych parametrach. Przykładowo, dopasowując rozkład do tempa przybywania zgłoszeń, możemy wybierać między rozkładem wykładniczym a rozkładem Weibulla. Pojęcie średniej rozkładu wykładniczego jest dla nas oczywiste, natomiast parametry rozkładu Weibulla są opisane skomplikowaną funkcją matematyczną.

Tabela 2.2. Charakterystyka niektórych rozkładów teoretycznych i przesłanki ich wyboru

Rozkład	Parametry	Charakterystyki	Gdzie stosować
Wykładniczy	Średnia	Duża zmienność Ograniczony z lewej Nieograniczony z prawej	Intensywność napływania zgłoszeń Czas między kolejnymi awariami
Trójkątny	Minimalna, modalna, maksymalna	Symetryczny lub niesymetryczny Ograniczony z obu stron	Czas wykonania
Normalny	Średnia, odchylenie	Symetryczny Nieograniczony z obu stron Może dawać ujemne wartości	Czas wykonania
Równomierny	Minimalna, maksymalna	Wszystkie wartości jednakowo prawdopodobne Ograniczony z obu stron	Mało informacji o cechach procesu

(c) *Przeznaczenie rozkładów.* Przy wyborze rozkładu spośród kilku, warto również przyjrzeć się potencjalnym możliwościom ich „stosowalności”, potwierdzonym w literaturze lub w innych modelach (por. tab. 2.2). W ostateczności, kiedy znamy jedynie wartość minimalną i maksymalną występującą w danym procesie rzeczywistym, możemy sięgnąć po rozkład jednostajny.

3. GENEROWANIE LICZB I ZMIENNYCH LOSOWYCH

3.1. WPROWADZENIE

Ustalenie postaci rozkładu losowego, który będzie poprawnie opisywał badane zjawisko, jest znaczącym osiągnięciem modelującego, ale nie zapewni mu uzyskania w modelu zmienności o postulowanym charakterze. Dostępne obecnie narzędzia do prowadzenia symulacji pozwalają stosunkowo prosto otrzymać pożądany rozkład losowy (najczęściej wystarczy wywołać odpowiednią funkcję), jednakże znajomość mechanizmów generowania zmiennych losowych może okazać się niezwykle użyteczna podczas realizacji eksperymentów symulacyjnych.

Uzyskanie w modelu losowych zmian zgodnych z wybranym rozkładem obejmuje w zasadzie dwa etapy: generowanie liczb losowych o rozkładzie równomiernym z przedziału $(0, 1)$, a następnie przekształcanie uzyskanych liczb losowych w dowolny rozkład zmiennej losowej czy w dowolny proces stochastyczny (np. niestacjonarny rozkład Poissona). Liczby losowe uzyskane poprzez, na przykład, rzut kostką są liczbami rzeczywiście losowymi. Liczby losowe generowane programowo są jedynie *liczbami pseudolosowymi*.

Metodologia generowania liczb losowych ma niezwykle długą i interesującą historię. W najwcześniejszych metodach proponowano: rzuty monetą, wyciąganie żetonów z kapelusza czy wybór kart z talii. Począwszy od lat dwudziestych XX wieku, również i w tę dziedzinę zaczęła wkraczać mechanizacja. Pojawiły się pierwsze maszyny losujące, które w sposób mechaniczny wybierały liczby losowe. Urządzenia takie istnieją zresztą do dzisiaj i mają się całkiem dobrze (np. maszyny losujące używane w *Totalizatorze Sportowym*). Wraz z rozwojem komputeryzacji wzrastało jednak zapotrzebowanie na szybko i łatwo dostępne liczby losowe i to w dużych ilościach. Początkowo stosowano tablice liczb losowych (np. te przygotowane przez Rand Corporation), ale coraz bardziej złożone modele symulacyjne potrzebowały coraz obfitszych strumieni liczb losowych i tablice (konsumujące znaczne zasoby pamięciowe) przestały satysfakcjonować użytkowników. Na przełomie lat czterdziestych i pięćdziesiątych XX wieku opracowano algorytmy umożliwiające pozyskanie liczb „losowych”. Algo-

rytmy te, nazwane *generatorami liczb pseudolosowych*, dostarczają ciągów liczb, które spełniają statystyczne własności liczb rzeczywiście losowych, jednakże są całkowicie deterministyczne, tj. każda z liczb jest zdeterminowana przez poprzednią, wszystkie zaś te liczby są zależne od liczby początkowej. Znając konstrukcję generatora oraz jego parametry, możemy bezbłędnie przewidzieć cały ciąg liczb. Takie rozwiązanie zadowala statystyków, natomiast symulogom oferuje dodatkowo niezwykle użyteczną w badaniach symulacyjnych własność. Pozwala bowiem na wielokrotne powtarzanie eksperymentu symulacyjnego w identycznych warunkach, co nie byłoby możliwe przy rezygnacji z determinizmu liczb produkowanych przez generatory.

Badania nad poszukiwaniem doskonałego generatora liczb losowych ciągle trwają. Jak jednak pokazuje praktyka, nawet bardzo dobry generator napotyka kiedyś w końcu na taki model, w którym działa jak antybiotyk: leczy, ale powoduje efekty uboczne. Użytkownicy języków symulacyjnych korzystają najczęściej z generatorów oferowanych wraz z pakietami. Ich jakość może być jednak różna i warto to wziąć pod uwagę przed podjęciem decyzji o wyborze oprogramowania.

Przekształcenie liczb losowych w pożądany rozkład zmiennej losowej odbywa się najczęściej bez udziału użytkowników pakietów symulacyjnych. Nie jest im w zasadzie potrzebna wiedza o tym, którą z metod matematycznych (transformacji odwrotnej, kompozycji, skreću czy inną) zastosowano. Modelujący musi być natomiast świadomy tego, że pobieranie wartości z rozkładów losowych skutkuje losowym charakterem wyników symulacji i automatycznie wprowadza wariancję do pozyskiwanego na wyjściu zbioru wartości. Im wyższa wariancja, tym mniej precyzyjne będą wyniki, tym szersze przedziały ufności, tym większa niepewność rezultatów symulacji.

3.2. LICZBY LOSOWE

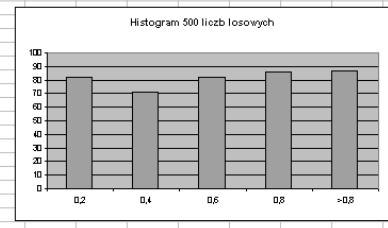
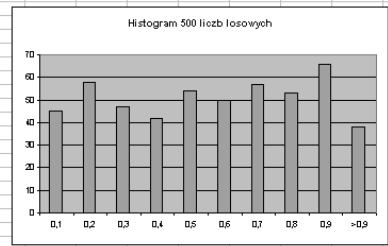
Liczby losowe są ilościowym odbiciem losowości. Zasadniczą cechą liczb losowych jest to, że znajomość liczb występujących w przeszłości nie wpływa na skuteczność przewidywania liczb przyszłych, a szansa uzyskania kolejnej liczby podczas kolejnej próby nie ulega zmianie. Ciąg liczb dwucyfrowych, np.: 83, 01, 22, 89, 27, 35, 41, 08, ... można uznać za ciąg liczb losowych, jeżeli liczby w nim zawarte spełniają następujące własności:

Własność nr 1: Częstości pojawiania się w tym ciągu liczb z przedziałów jednakowej długości (np. z przedziałów 00–49, 50–99 albo też 00–24, 25–49, 50–74, 75–99) są jednakowe. Innymi słowy – szanse na to, że „losowo” wybrana liczba (od 00 do 99) znajdzie się w którymkolwiek z przedziałów są takie same dla jednakowych przedziałów. Omówiona własność nosi nazwę *równomierności rozkładu*.

Własność nr 2: Elementy dowolnego podciągu złożonego z tych liczb są łącznie *niezależne*, a każdy z nich ma rozkład jednostajny w przedziale (0, N). Poszczególne liczby nie pojawiają się w sposób okresowy. Można zatem oczekiwać, że każda liczba od 00 do 99 występuje mniej więcej raz na 100 losowań. Jeżeli w pewnym momencie uzyskamy liczbę losową o określonej wartości (np. równą 43), nie poszerzymy w żaden sposób naszej wiedzy o wyniku kolejnego losowania. Kolejna wylosowana liczba nadal będzie miała 50% szans na to, że będzie mniejsza od 50.

Na rysunku 3.1 przedstawiono poglądową ilustrację pierwszej cechy losowości: 500 liczb losowych (wygenerowanych za pomocą funkcji *LOS()* w Excelu) opisano dwoma histogramami. Histogram górny pokazuje częstość występowania liczb w przedziałach o skoku 0,1, natomiast histogram dolny – w przedziałach o skoku 0,2. Zastosowanie odpowiednich testów statystycznych (np. χ^2) potwierdzi równomierność rozkładu 500 liczb w omawianych przedziałach.

Tablica 500 wartości uzyskanych za pomocą funkcji LOS()													Rozkład liczb losowych					
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	0,29734	0,42082	0,66578	0,8638	0,56140	0,60083	0,36342	0,32396	0,86158	0,79279			Wartość	Częstość		Wartość	Częstość	
2													0,1	46		0,2	82	
3	0,1124	0,07782	0,24761	0,02547	0,39142	0,38591	0,24753	0,18673	0,13146	0,23447			0,2	58		0,4	71	
4	0,46881	0,01706	0,93409	0,89946	0,7658	0,09912	0,66101	0,99159	0,58011	0,43648			0,3	47		0,6	82	
5	0,77941	0,82387	0,89327	0,82465	0,11428	0,51606	0,41775	0,92993	0,04071	0,99153			0,4	42		0,8	86	
6	0,44139	0,02062	0,05423	0,83134	0,75288	0,01712	0,37969	0,04655	0,35381	0,59759			0,5	54	>0,8		87	
7	0,62376	0,19551	0,17352	0,57196	0,85294	0,88933	0,49534	0,50169	0,42073	0,22655			0,6	50				
8	0,95919	0,65431	0,62382	0,79128	0,25373	0,15584	0,19987	0,12168	0,1841	0,26808			0,7	57				
9	0,22192	0,68787	0,05035	0,70739	0,18165	0,70135	0,18707	0,02495	0,21467	0,47575			0,8	53				
10	0,38308	0,41179	0,58518	0,25992	0,90239	0,58963	0,92545	0,04318	0,2468	0,23833			0,9	66				
11	0,41305	0,76364	0,85214	0,05884	0,09702	0,08386	0,82609	0,22966	0,65032	0,02133			>0,9	38				
12	0,94244	0,88673	0,67479	0,29839	0,70333	0,61627	0,58462	0,73208	0,98214	0,79117								
13	0,76125	0,49888	0,30098	0,1148	0,62482	0,54605	0,95271	0,75353	0,60329	0,10628								
14	0,11862	0,45923	0,9972	0,49949	0,89146	0,13447	0,87361	0,86768	0,94423	0,50109								
15	0,31595	0,19551	0,4331	0,84099	0,57398	0,14878	0,85949	0,40648	0,28011	0,70334								
16	0,62398	0,21975	0,71358	0,26318	0,85638	0,74674	0,99238	0,40528	0,4476	0,11607								
17	0,33798	0,48569	0,11872	0,18182	0,88616	0,16265	0,62927	0,24655	0,74273	0,80202								
18	0,70408	0,81024	0,6761	0,18302	0,69542	0,91519	0,33149	0,74117	0,22592	0,94535								
19	0,6195	0,35583	0,53871	0,8702	0,40534	0,08843	0,70045	0,30289	0,91991	0,64879								
20	0,83721	0,43448	0,94649	0,65529	0,86233	0,13561	0,05329	0,53678	0,93985	0,60235								
21	0,19298	0,48146	0,75944	0,82527	0,52046	0,52336	0,17003	0,07213	0,58741	0,12228								
22	0,58412	0,41957	0,97257	0,82351	0,9039	0,55337	0,32394	0,71987	0,04707	0,47559								
23	0,79568	0,37626	0,0132	0,84833	0,82247	0,44126	0,86562	0,48034	0,25206	0,42745								
24	0,4432	0,89613	0,21482	0,92486	0,28239	0,57347	0,23419	0,15623	0,56216	0,20422								
25	0,12575	0,21305	0,82924	0,16264	0,45458	0,65576	0,4179	0,38835	0,20041	0,48586								
26	0,58703	0,4196	0,89925	0,49842	0,18642	0,07355	0,56871	0,52011	0,18959	0,15426								
27	0,12217	0,01949	0,18263	0,84223	0,77925	0,27391	0,27203	0,82071	0,3855	0,19672								
28	0,61817	0,08023	0,18589	0,90082	0,18621	0,9657	0,0031	0,60779	0,33595	0,61208								
29	0,18495	0,58369	0,47021	0,11143	0,80926	0,94630	0,70943	0,89143	0,7089	0,62404								
30	0,85748	0,24473	0,93996	0,26366	0,47471	0,119	0,57379	0,36361	0,2847	0,82611								
31	0,8228	0,96275	0,56572	0,11438	0,53819	0,77159	0,0088	0,68616	0,54511	0,7085								
32	0,39002	0,62208	0,35505	0,18161	0,81688	0,83444	0,24392	0,77431	0,73975	0,26713								
33	0,59831	0,31551	0,69391	0,13442	0,30627	0,57601	0,21067	0,22143	0,87213	0,43342								
34	0,84968	0,14163	0,85521	0,22599	0,66993	0,38515	0,79604	0,08964	0,58147	0,42527								
35	0,13368	0,1011	0,77559	0,29006	0,8662	0,14374	0,87442	0,89539	0,87484	0,07195								
36	0,11258	0,83377	0,91667	0,80569	0,95381	0,49608	0,89969	0,88725	0,85881	0,61171								
37	0,89713	0,67995	0,84979	0,52577	0,77742	0,22268	0,21798	0,98583	0,05451	0,78174								
38	0,54018	0,33488	0,83572	0,57655	0,55288	0,88448	0,71305	0,11332	0,14685	0,47574								
39	0,0612	0,25752	0,46784	0,86754	0,34391	0,74731	0,57688	0,72918	0,47985	0,83494								
40	0,3544	0,7612	0,18788	0,3684	0,29084	0,58978	0,48705	0,53423	0,86387	0,32465								
41	0,94575	0,38656	0,64348	0,82754	0,52798	0,47888	0,05218	0,66704	0,2092	0,73639								
42	0,86466	0,84661	0,31788	0,98164	0,50475	0,35102	0,58465	0,06785	0,58335	0,31251								
43	0,75935	0,79381	0,39708	0,74287	0,88717	0,13464	0,52283	0,95699	0,76107	0,68261								
44	0,02212	0,47395	0,15443	0,87502	0,80289	0,39952	0,93952	0,24688	0,0697	0,17199								
45	0,84968	0,34494	0,59648	0,03117	0,29745	0,26307	0,89545	0,70022	0,75292	0,86487								
46	0,90768	0,53085	0,93154	0,8266	0,28727	0,0898	0,08959	0,0811	0,67623	0,44186								
47	0,68048	0,18188	0,67052	0,42238	0,40867	0,75155	0,20137	0,88784	0,73593	0,8278								
48	0,07438	0,88995	0,06485	0,09116	0,78103	0,7192	0,4714	0,85067	0,60046	0,8047								
49	0,43047	0,88275	0,92851	0,12284	0,62348	0,39784	0,41271	0,44395	0,25199	0,03291								
50	0,65851	0,48263	0,42607	0,89561	0,69664	0,83071	0,57718	0,67763	0,22573	0,05945								
51	0,1638	0,91856	0,11092	0,54634	0,03093	0,93818	0,46262	0,81449	0,13916	0,8946								
52	0,70075	0,30024	0,12361	0,2203	0,74986	0,83329	0,7583	0,47795	0,2598	0,69729								
53																		
54																		



Rys. 3.1. Ilustracja pierwszej własności liczb losowych

Źródło: opracowanie własne

Przykładowa ilustracja drugiej cechy mogłaby wyglądać następująco: policzmy przypadki, w których po „Małej” liczbie następuje „Duża” liczba. Zdefiniujemy „Małą” liczbę jako mniejszą od (lub równą) 0,5, natomiast „Dużą” liczbę jako większą od 0,5. Sprawdzenie wykonamy przesuwać się kolumnami (rys. 3.2). Uzyskane wyniki

	A	B	C	D	E	F	G	H	I	J	K
1	tablica 500 wartości uzyskanych za pomocą funkcji LOS()										
2											
3	0,42716	0,92233	0,98348	0,5655	0,385841989	0,31791	0,80492	0,10361	0,39761	0,69964	
4	0,62831	0,20663	0,32593	0,32982	0,324826238	0,223	0,41737	0,24338	0,2566	0,73201	
5	0,67697	0,77034	0,28597	0,13044	0,493118737	0,16946	0,46675	0,77184	0,02022	0,93777	
6	0,66406	0,1428	0,56124	0,13505	0,455139367	0,35569	0,11711	0,85764	0,54553	0,91837	
7	0,44198	0,45303	0,15459	0,92108	0,60174216	0,72885	0,95831	0,3646	0,58326	0,29074	
8	0,83546	0,71333	0,68878	0,88641	0,890100304	0,672	0,88835	0,86913	0,65713	0,72838	
9	0,99639	0,71304	0,8463	0,92685	0,711914828	0,12295	0,42938	0,44926	0,66014	0,49063	
10	0,86827	0,41459	0,19613	0,12504	0,53669827	0,64528	0,19218	0,46054	0,30641	0,11079	
11	0,27491	0,24194	0,47582	0,58866	0,763198996	0,40602	0,56035	0,47025	0,66047	0,83468	
12	0,4267	0,15082	0,37317	0,82682	0,096794164	0,00824	0,66407	0,90902	0,09619	0,57956	
13	0,10705	0,89236	0,34779	0,63152	0,499177293	0,28503	0,43642	0,42686	0,11284	0,79461	
14	0,71842	0,03508	0,2131	0,93453	0,733502006	0,25308	0,67804	0,19534	0,76845	0,6785	
15	0,54144	0,78244	0,53361	0,28738	0,268819474	0,22188	0,48264	0,53368	0,52432	0,48237	
16	0,54213	0,19879	0,01639	0,10425	0,365735966	0,76694	0,93316	0,57507	0,0203	0,5427	
17	0,68998	0,83174	0,88164	0,98561	0,344035667	0,00582	0,05975	0,6711	0,18628	0,35006	
18	0,72774	0,91564	0,26135	0,08462	0,972189522	0,666	0,99669	0,44953	0,20214	0,5308	
19	0,29471	0,4187	0,80859	0,24576	0,981980901	0,96139	0,12287	0,99918	0,6215	0,10573	
20	0,09764	0,18997	0,64678	0,04464	0,512761767	0,26049	0,71327	0,82817	0,54237	0,20734	
21	0,49893	0,87581	0,46598	0,53286	0,266298904	0,96661	0,49799	0,12894	0,99775	0,61916	
22	0,48451	0,32393	0,46394	0,54243	0,286257113	0,49018	0,85096	0,52863	0,38899	0,28543	
23	0,88841	0,80589	0,15799	0,37114	0,974866084	0,14405	0,2129	0,86244	0,73981	0,15041	
24	0,04052	0,29181	0,03117	0,58679	0,572485028	0,53501	0,05675	0,12136	0,9053	0,91393	
25	0,39616	0,35436	0,54933	0,91267	0,74704219	0,71729	0,30257	0,38724	0,67755	0,21347	
26	0,84075	0,12989	0,1456	0,4687	0,280342875	0,96905	0,49117	0,697	0,986	0,52976	
27	0,85982	0,64888	0,32649	0,17698	0,566435765	0,60873	0,84802	0,61894	0,89781	0,93378	
28	0,76659	0,72157	0,9639	0,16656	0,804903292	0,8572	0,93449	0,9682	0,51937	0,02294	
29	0,92036	0,0017	0,80153	0,35888	0,556882961	0,09381	0,17889	0,19051	0,32074	0,46901	
30	0,42473	0,32647	0,97221	0,85987	0,711195249	0,96451	0,16686	0,98197	0,54534	0,18443	
31	0,87852	0,92584	0,60663	0,38238	0,604945633	0,71993	0,0979	0,48029	0,8533	0,85731	
32	0,05362	0,74095	0,47913	0,12639	0,791153148	0,31428	0,23471	0,24733	0,74647	0,11328	
33	0,11057	0,85153	0,91627	0,77311	0,897929109	0,9349	0,31725	0,31505	0,33725	0,88963	
34	0,09274	0,46481	0,31547	0,93281	0,151585529	0,55981	0,76228	0,67514	0,25379	0,31993	
35	0,53081	0,54295	0,61026	0,22069	0,65736782	0,70527	0,56802	0,40784	0,39468	0,6526	
36	0,48653	0,5069	0,16732	0,88575	0,627499687	0,76829	0,51988	0,06408	0,35433	0,30385	
37	0,52653	0,57818	0,9474	0,8061	0,483921676	0,98333	0,59533	0,97418	0,49992	0,15796	
38	0,00458	0,9454	0,39576	0,74791	0,187306443	0,73363	0,48738	0,26864	0,74189	0,24198	
39	0,46493	0,89252	0,78113	0,61924	0,518115051	0,34154	0,23237	0,23491	0,64973	0,39135	
40	0,59283	0,41466	0,81206	0,12184	0,257703952	0,51044	0,20174	0,8546	0,57058	0,06205	
41	0,40595	0,57847	0,58195	0,40061	0,064269349	0,88146	0,24841	0,42817	0,98037	0,0461	
42	0,01291	0,80815	0,1027	0,6371	0,802700924	0,32773	0,11069	0,84909	0,22224	0,84961	
43	0,51824	0,83243	0,92869	0,38435	0,402827922	0,42116	0,06488	0,44821	0,09553	0,56633	
44	0,21386	0,90307	0,05455	0,08154	0,396076796	0,77866	0,06767	0,33366	0,05072	0,0903	
45	0,39577	0,89005	0,74022	0,76719	0,837344044	0,31806	0,10407	0,63628	0,4332	0,96836	
46	0,26034	0,62693	0,63134	0,01							
47	0,88471	0,4433	0,0713	0,04							
48	0,91283	0,65082	0,1035	0,98							
49	0,14997	0,61612	0,01216	0,83							
50	0,65431	0,71056	0,81697	0,57							
51	0,86887	0,35441	0,07058	0,52							
52	0,773	0,5221	0,97763	0,06							
53											
54											
55											
56											
57											

Rys. 3.2. Ilustracja drugiej własności. Porównywanie wzdłuż kolumn, czy po małej liczbie (mniejszej od 0,5) występuje duża liczba (większa od 0,5)

wskazują, że zbiory liczb „po Małej liczbie występuje Duża liczba” (zaciemnione komórki na rys. 3.2) i „po Małej liczbie występuje Mała liczba” są równoliczne (patrz – wyświetlany komunikat na rysunku). Dla czytelności, na rysunku zaciemniono tylko zbiór pierwszy, czyli zbiór „Po Małej liczbie występuje Duża liczba”. Nie ma znacze-

nia, które komórki ze sobą porównujemy: czy te leżące obok siebie w wierszu, czy w kolumnie. Powtórzenie eksperymentu, ale tym razem przez porównanie liczb położonych obok siebie wzdłuż wierszy dałoby podobny wynik. Oczywiście, podany przykład nie stanowi *dowodu* na spełnienie własności nr 2.

3.3. TABLICE LICZB LOSOWYCH

Jedną z metod pozyskiwania liczb losowych polega na przygotowaniu tablic liczb losowych (tablice można potem np. zapamiętać w komputerze). Przyjmując, że liczby te poddano badaniom testów, których zadaniem jest wykrycie odchyłeń od zjawisk niezależności i równomierności, należy oczekiwać, iż metoda ta pomoże w uzyskaniu właściwych pod względem statystycznym liczb losowych. Firma Rand Corporation przygotowała i opublikowała zbiór miliona cyfr losowych, który można wykorzystać do budowy tablic liczb losowych. Publikowane tablice liczb losowych spełniających wszystkie wymienione warunki zawierają liczby cztero- i pięciocyfrowe. Odczytywanie liczb z tablic liczb losowych można rozpocząć od dowolnego miejsca w tablicach, a kolejne liczby mogą być odczytywane po kolei wierszami lub kolumnami. W przypadku jednoczesnego symulowania kilku zjawisk losowych w ramach jednego eksperymentu symulacyjnego zaleca się stosowanie odrębnych ciągów liczb losowych. Słabą stroną tablicy liczb losowych jest to, że może ona się wyczerpać, zanim zakończy się proces generowania liczb losowych.

3.4. GENEROWANIE LICZB PSEUDOLOSOWYCH

W praktyce, w celu uzyskania liczb o losowym charakterze, tworzy się nielosowe ciągi nazywane *liczbami pseudolosowymi*. Generatory liczb pseudolosowych dostarczają ciągów liczb, które spełniają statystyczne własności liczb rzeczywiście losowych (por. podrozdz. 3.2), jednakże są całkowicie deterministyczne (każda z liczb jest zdefiniowana przez poprzednią, wszystkie zaś te liczby są zależne od liczby początkowej). Jeżeli możemy wykazać, że liczby występujące w ciągu wydają się mieć rozkład równomierny i są w sensie statystycznym niezależne, to można założyć, że proces jest losowy. Znając konstrukcję generatora oraz jego parametry, możemy bezbłędnie przewidzieć cały ciąg liczb. Takie podejście ma jednak swoje zalety, bowiem własności statystyczne uzyskanych ciągów liczb są zadowalające, a ponadto możliwe jest powtarzanie eksperymentu symulacyjnego w dokładnie takich samych warunkach

wielokrotnie, co nie byłoby możliwe, gdyby się zrezygnowało z determinizmu liczb produkowanych przez generatory.

Komputerowe generatory liczb losowych muszą mieć następujące cechy:

- długi okres, tzn. musi minąć długi ciąg liczb, zanim ich kolejność zacznie się powtarzać,
- źródło powinno być efektywne, tzn. generator musi być szybki i wymagać mało pamięci,
- źródło powinno być odpowiednio gęste, tzn. przerwy pomiędzy generowanymi wartościami nie mogą być duże,
- powinna istnieć możliwość dokładnego powtórzenia ciągu liczb,
- generator musi produkować liczby pseudolosowe, które przejdą wszystkie testy losowości.

Komputerowe generatory liczb losowych wykorzystują ustalone wzory matematyczne i obliczają kolejne liczby na podstawie wartości liczb poprzednio wygenerowanych. Pierwsza liczba, zwana wartością początkową, jest zadawana. Dla zadanej wartości początkowej generator zawsze wytworzy ten sam ciąg liczb.

Historycznie, najczęściej używanymi generatorami liczb pseudolosowych są *generatory liniowe* oparte na kongruencji liczb (generatory LCG – *Linear Congruential Generators*). Dwie liczby całkowite a i b są *w kongruencji z modulem m* , jeżeli ich różnica jest liczbą całkowitą, stanowiącą wielokrotność m . Relację kongruencji liczb zapisujemy jako $a \equiv b \pmod{m}$, co odczytuje się: „ a jest kongruentne z b modulo m ”. Oznacza to, że $a - b$ jest podzielne przez m i że a oraz b dają identyczne reszty, gdy dzieli się je przez $|m|$. Ciągi liczb generowane metodą LCG są definiowane przez wzór rekurencyjny:

$$Z_i = (aZ_{i-1} + c) \pmod{m} \quad (3.1)$$

gdzie m (moduł), a (mnożnik), c (przyrost), Z_0 (ziarno lub liczba startowa) są nieujemnymi liczbami całkowitymi.

Gdy $c = 0$, generatory te nazywamy *multiplikatywnymi*, w przeciwnym przypadku *mieszanymi*. Generatory *multiplikatywne* wykazują dobre własności statystyczne. Oznacza to, że przy odpowiednim wyborze mnożnika a i wartości początkowej Z_0 , można generować takie ciągi liczb, które nie są ze sobą skorelowane i mają rozkład równomierny. Ponadto, przy zapewnieniu pewnych warunków dla a i Z_0 , możemy zapewnić maksymalnie długi okres dla ciągów generowanych tą metodą.

Korzyść ze stosowania generatora *mieszanego* polega na tym, że przy właściwym doborze a i c można otrzymać ciąg o takim okresie, który obejmuje pełny zbiór m różnych liczb. Drugą korzyścią jest to, że przy właściwym wyborze c możliwe jest zminimalizowanie korelacji serii. Generator LCG o parametrach $m = 2^{31} - 1 = 2\,147\,483\,647$, $a = 7^5 = 16\,807$, $c = 0$ pozwala uzyskać ciąg o $2,1 \times 10^9$ (2,1 miliarda) pozycji. Obecnie stosowane generatory charakteryzują się okresami o długości $2,1 \times 10^{57}$ pozycji.

3.5. PRZYKŁAD DZIAŁANIA GENERATORA LCG

Przyjmijmy do ilustracji działania generatora LCG, że: $m = 63$, $a = 22$, $c = 4$, $Z_0 = 19$. Wzór rekurencyjny będzie miał postać: $Z_i = (22 Z_{i-1} + 4) \pmod{63}$, ziarno $Z_0 = 19$. W tabeli 3.1 zaprezentowano wyniki działania generatora LCG dla pierwszych 70 wartości.

Wartościami, które dostarcza generator jest kolumna Z_i złożona z liczb całkowitych. Ponieważ są to reszty z dzielenia przez m , będą to zatem liczby z zakresu od 0 do $m - 1$. Końcowym krokiem będzie podzielenie wartości z kolumny Z_i przez m , czyli wykonanie działania $U_i = Z_i/m$, co pozwoli nam uzyskać liczby z przedziału od 0 do 1.

Tabela 3.1. Wartości wygenerowane przez generator LCG

i	$22 Z_{i-1} + 4$	Z_i	U_i	i	$22 Z_{i-1} + 4$	Z_i	U_i	i	$22 Z_{i-1} + 4$	Z_i	U_i
0		19		24	1060	52	0,825396825	48	400	22	0,349206349
1	422	44	0,698412698	25	1148	14	0,222222222	49	488	47	0,746031746
2	972	27	0,428571429	26	312	60	0,952380952	50	1038	30	0,476190476
3	598	31	0,492063492	27	1324	1	0,015873016	51	664	34	0,53968254
4	686	56	0,888888889	28	26	26	0,412698413	52	752	59	0,936507937
5	1236	39	0,619047619	29	576	9	0,142857143	53	1302	42	0,666666667
6	862	43	0,682539683	30	202	13	0,206349206	54	928	46	0,73015873
7	950	5	0,079365079	31	290	38	0,603174603	55	1016	8	0,126984127
8	114	51	0,80952381	32	840	21	0,333333333	56	180	54	0,857142857
9	1126	55	0,873015873	33	466	25	0,396825397	57	1192	58	0,920634921
10	1214	17	0,26984127	34	554	50	0,793650794	58	1280	20	0,317460317
11	378	0	0	35	1104	33	0,523809524	59	444	3	0,047619048
12	4	4	0,063492063	36	730	37	0,587301587	60	70	7	0,111111111
13	92	29	0,46031746	37	818	62	0,984126984	61	158	32	0,507936508
14	642	12	0,19047619	38	1368	45	0,714285714	62	708	15	0,238095238
15	268	16	0,253968254	39	994	49	0,777777778	63	334	19	0,301587302
16	356	41	0,650793651	40	1082	11	0,174603175	64	422	44	0,698412698
17	906	24	0,380952381	41	246	57	0,904761905	65	972	27	0,428571429
18	532	28	0,444444444	42	1258	61	0,968253968	66	598	31	0,492063492
19	620	53	0,841269841	43	1346	23	0,365079365	67	686	56	0,888888889
20	1170	36	0,571428571	44	510	6	0,095238095	68	1236	39	0,619047619
21	796	40	0,634920635	45	136	10	0,158730159	69	862	43	0,682539683
22	884	2	0,031746032	46	224	35	0,555555556	70	950	5	0,079365079
23	48	48	0,761904762	47	774	18	0,285714286				

Pozornie wydaje się, że wartości umieszczone w kolumnie U_i są „porządnymi” liczbami losowymi: pochodzą z przedziału od 0 do 1, wydają się pokrywać równomiernie cały przedział (0, 1) i są dobrze wymieszane (niezależne). Średnia z próby wynosi 0,4984, a odchylenie standardowe 0,2867, co dość dobrze pokrywa się z wartością oczekiwaną i odchyleniem rozkładu jednostajnego.

Bardziej wnikliwa analiza uzyskanego ciągu pozwala jednak zauważyć, że od wartości Z_{63} liczby zaczynają się powtarzać (19, 44, 27, 31, ...), co jest niestety zjawiskiem nieuniknionym.

3.6. GENEROWANIE ZMIENNYCH LOSOWYCH

3.6.1. GENEROWANIE CIĄGŁYCH ZMIENNYCH LOSOWYCH – METODA TRANSFORMACJI ODWROTNEJ

Kolejnym problemem, jaki stoi przed twórcą modelu symulacyjnego, jest konieczność generowania realizacji zmiennych losowych (a w dalszej kolejności generowanie ciągów odpowiadających procesom stochastycznym) o parametrach, które zostały wyestymowane poprzez badanie rzeczywistych zjawisk. W praktyce stosuje się cztery sposoby generowania zmiennych losowych: *metodę transformacji odwrotnej*, *metodę eliminacji*, *metodę kompozycji* i *metodę skrętu*. Omówiono pierwszą z metod (zwaną inaczej *metodą odwracania dystrybuanty*), która jest najczęściej wykorzystywana w pakietach symulacyjnych.

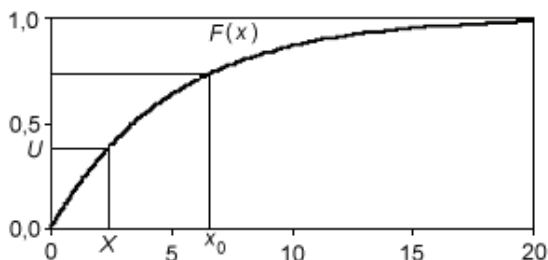
Założmy, że $f(x)$ jest pewną funkcją gęstości prawdopodobieństwa. Dystrybuan-tą jest funkcja:

$$F(x) = \int_{-\infty}^x f(x) dx \quad (3.2)$$

niemalejąca i o wartościach z przedziału (0, 1). Założmy również, że dany jest ciąg liczb losowych u_i z przedziału (0,1) o rozkładzie jednostajnym. Jeżeli liczby losowe potraktujemy jako kolejne wartości funkcji $F(x)$ i znajdziemy odpowiadające im wartości argumentu x_i , to otrzymamy ciąg losowy o funkcji gęstości $f(x)$. Inaczej mówiąc, metoda transformacji odwrotnej polega na odczytywaniu wartości odwrotnej funkcji dystrybuanty dla losowych wartości argumentu, por. rys. 3.3. Najpierw określana jest liczba losowa z przedziału (0, 1), czyli wartość $F(x)$, a następnie za pomocą funkcji odwrotnej wyznaczana jest wartość x :

$$x = F^{-1}(u) \quad (3.3)$$

Użyteczność tej metody zależy od tego, jak łatwo można określić *funkcję odwrotną* do danej.



Rys. 3.3. Ilustracja metody transformacji odwrotnej

Przykład 1

Załóżmy, że funkcja gęstości dana jest wzorem:

$$f(x) = \begin{cases} 2x & 0 \leq x \leq 1 \\ 0 & \text{w p.p.} \end{cases} . \quad (3.4)$$

Dystrybuanta tej funkcji to:

$$F(x) = \begin{cases} 0 & x < 0 \\ x^2 & 0 \leq x \leq 1, \\ 1 & x > 1 \end{cases} . \quad (3.5)$$

natomiast funkcja odwrotna:

$$x = F(u)^{-1} = \sqrt{u} . \quad (3.6)$$

Generujemy liczbę losową u (np. $u = 0,3545$) i korzystamy ze wzoru na funkcję odwrotną: $x = \sqrt{0,3545} = 0,5696$.

Przykład 2. Rozkład jednostajny

Dystrybuanta rozkładu jednostajnego to, dla przypomnienia:

$$F(x) = \begin{cases} 0 & \text{dla } x < a \\ \frac{x-a}{b-a} & \text{dla } a \leq x \leq b . \\ 1 & \text{dla } b < x \end{cases} . \quad (3.7)$$

Po podstawieniu $u = F(x)$, czyli:

$$u = \frac{x - a}{b - a} \quad (3.8)$$

i po dokonaniu prostego przekształcenia otrzymujemy:

$$x = a + (b - a)u \quad (3.9)$$

W celu wyznaczenia x najpierw generujemy liczbę losową u , a następnie wykorzystujemy przekształcony wzór.

Przykład 3. Rozkład wykładniczy

Dystrybuanta rozkładu wykładniczego ma postać:

$$F(x) = 1 - e^{-\lambda x} \quad (3.10)$$

Po podstawieniu $u = F(x)$ otrzymujemy funkcję odwrotną:

$$\lambda x = -\ln(1 - u). \quad (3.11)$$

Po podstawieniu pod u ciągu liczb losowych o rozkładzie jednostajnym z przedziału $(0, 1)$ otrzymujemy ciąg liczb losowych o rozkładzie wykładniczym. Jeżeli liczby u mają rozkład jednostajny, to taki sam rozkład mają liczby $1 - u$. Można zatem użyć prostszego wzoru:

$$\lambda x = -\ln(u). \quad (3.12)$$

Przykład 4. Rozkład trójkątny

Dystrybuanta rozkładu trójkątnego to:

$$F(x) = \begin{cases} 0 & \text{dla } x < a \\ \frac{(x - a)^2}{(b - a)(c - a)} & \text{dla } a \leq x \leq c \\ 1 - \frac{(b - x)^2}{(b - a)(b - c)} & \text{dla } c < x \leq b \\ 1 & \text{dla } b < x \end{cases} \quad (3.13)$$

Szczególny przypadek występuje, gdy $b = c$, przy założeniu $a < c$. Dystrybuantę oblicza się wtedy wg wzoru:

$$F(x) = \begin{cases} 0 & \text{dla } x < a \\ \left(\frac{x-a}{c-a}\right)^2 & \text{dla } a \leq x \leq c \\ 1 & \text{dla } b < x \end{cases} \quad (3.14)$$

Funkcja odwrotna to:

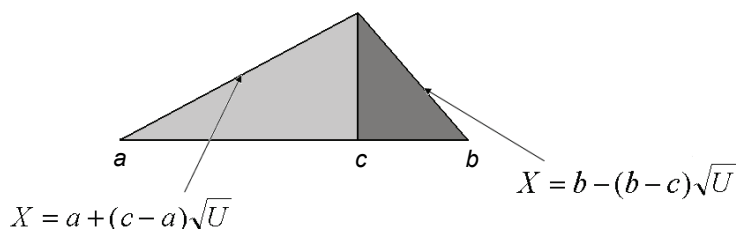
$$x = F(u)^{-1} = a + (c-a)\sqrt{u} \quad (3.15)$$

W podobny sposób wyznaczmy funkcję odwrotną, zakładając, że $a = c$:

$$x = F(u)^{-1} = b - (b-c)\sqrt{1-u} \quad (3.16)$$

Procedura uzyskania wartości zmiennej losowej dla rozkładu trójkątnego niesymetrycznego, zakładając, że $a < c < b$ (por. rys. 3.4) jest dwustopniowa. Czynnością wstępną jest obliczenie wartości p według wzoru:

$$p = \frac{c-a}{b-a} \quad (3.17)$$



Rys. 3.4. Rozkład trójkątny o parametrach (a, b, c)

Źródło: opracowanie własne

Następnie generujemy dwie liczby losowe u_1 i u_2 . Pierwszą liczbę u_1 wykorzystujemy, porównując jej wartość z wartością p . Jeżeli $u_1 \leq p$, to do wygenerowania wartości zmiennej losowej o rozkładzie trójkątnym posługujemy się formułą (3.15), podstawiając jako u drugą liczbę losową u_2 , w przeciwnym przypadku wybieramy formułę (3.16), również podstawiając jako u drugą liczbę losową u_2 . Jeżeli liczby u mają rozkład jednostajny, to taki sam rozkład mają liczby $1-u$. W praktyce, formuła (3.16) przybiera zatem postać:

$$x = F(u)^{-1} = b - (b-c)\sqrt{u} \quad (3.18)$$

Cały algorytm wygląda następująco:

$$p = \frac{c-a}{b-a} \quad (3.19)$$

Jeżeli $u_1 \leq p$, to

$$x = a + (c-a)\sqrt{u_2} \quad (3.20)$$

W przeciwnym razie

$$x = b - (b-c)\sqrt{u_2} \quad (3.21)$$

3.6.2. GENEROWANIE CIĄGLYCH ZMIENNYCH LOSOWYCH – PRZYKŁADOWE FORMUŁY

Przykładowe formuły umożliwiające pozyskiwanie w modelu symulacyjnym wartości zmiennej losowej typu ciągłego podano w tabeli 3.2. Symbolem u oznaczone są liczby losowe o rozkładzie równomiernym. W przypadku rozkładu trójkątnego musimy wygenerować dwie liczby losowe: u_1 i u_2 .

Użyteczność metody transformacji odwrotnej zależy od tego, jak łatwo można określić funkcję odwrotną do danej. W niektórych przypadkach (np. dla rozkładu normalnego) nie jest możliwe zastosowanie tego podejścia. Stosuje się wtedy inne, bardziej złożone metody.

Tabela 3.2. Przykładowe formuły pozyskiwania w modelu symulacyjnym wartości ciągłej zmiennej losowej

Rozkład	Parametry	Formuła
Jednostajny	$a < b$	$x = a + (b-a)u$
Trójkątny w (0,1)	0, ½, 1	$x = \frac{1}{2}(u_1 + u_2)$
Trójkątny symetryczny	$a < b$	$x = a + \frac{(b-a)}{2}(u_1 + u_2)$
Trójkątny niesymetryczny	$a < c < b$	$p = \frac{(c-a)}{(b-a)}$ $x = a + (c-a)\sqrt{u_2}, \text{ gdy } u_1 \leq p$ $x = b - (b-c)\sqrt{u_2}, \text{ gdy } u_1 > p$
Wykładniczy	μ	$x = -\mu \cdot \ln(u)$

W arkuszu kalkulacyjnym Excel można korzystać z gotowych funkcji do generowania wartości zmiennych losowych. Są to między innymi następujące rozkłady losowe:

- rozkład normalny: *ROZKŁAD.NORMALNY.ODW(Los(), Średnia, Odchylenie)*,
 - rozkład beta: *ROZKŁAD.BETA.ODW(Los(), Alfa, Beta)*,
 - rozkład gamma: *ROZKŁAD.GAMMA.ODW(Los(), Alfa, Beta)*,
 - rozkład logarytmiczno-normalny: *ROZKŁAD.LOG.ODW(Los(), Średnia, Odchylenie)*,
- gdzie *Los()* to bezparametrowa funkcja generująca liczby losowe z przedziału (0, 1).

3.6.3. GENEROWANIE DYSKRETNYCH ZMIENNYCH LOSOWYCH

W przypadku empirycznych rozkładów dyskretnych (a z takimi mamy najczęściej do czynienia w modelach symulacyjnych) również posługujemy się metodą odwracania dystrybuanty. Różnica polega na tym, że ponieważ nie dysponujemy formułą teoretyczną, wykorzystujemy empiryczny rozkład prawdopodobieństwa oraz empiryczną dystrybuantę. Tabela z przykładowymi wartościami prawdopodobieństwa i dystrybuanty dla pewnej zmiennej losowej przyjmującej cztery wartości została przedstawiona w tab. 3.3. Zauważmy, że dodana została kolumna *Zakres liczb losowych*, która pokrywa wszystkie wartości z przedziału (0, 1), ale jednocześnie przydziela je poszczególnym wartościom zmiennej losowej tak, aby odpowiadało to dokładnie *szansie* (prawdopodobieństwu) wystąpienia danej wartości. Jeżeli wartość 1 stanowi 9% wszystkich wartości (prawdopodobieństwo równe 0,09), to przydzielamy tej wartości zmiennej losowej 9% zakresu liczb losowych (czyli liczby od 0,01 do 0,09). Skoro wartość 2 stanowi 21% wszystkich wartości (prawdopodobieństwo równe 0,21), to przydzielamy 21% zakresu liczb losowych (od 0,09 do 0,30), i tak dalej.

Tabela 3.3. Przykładowe wartości prawdopodobieństwa i dystrybuanty dla zmiennej losowej dyskretnej.
Rozkład empiryczny

Numeryczna wartość zmiennej losowej	Liczba wystąpień	Prawdopodobieństwo $P()$	Dystrybuanta	Zakres liczb losowych
1	32	0,09	0,09	(0,00 do 0,9]
2	78	0,21	0,30	(0,09 do 0,30]
3	158	0,43	0,73	(0,30 do 0,73]
4	97	0,27	1,00	(0,73 do 1,00)

W modelu symulacyjnym będziemy generowali kolejne liczby losowe z przedziału (0, 1), następnie będziemy sprawdzali, do którego z czterech przedziałów z tab. 3.3 wygenerowana liczba należy i w ten sposób uzyskamy informację o wartości zmiennej losowej. Na przykład, jeżeli generator liczb losowych (czyli funkcja *LOS()*) zwróci nam wartość 0,245, to będzie to oznaczało, że zmienna losowa przyjmie wartość 2. Jeżeli *LOS()* = 0,8765, to zmienna losowa przyjmie wartość 4.

3.7. GENEROWANIE STRUMIENI ZGŁOSZEŃ

3.7.1. STACJONARNY ROZKŁAD POISSONA

W wielu modelach symulacyjnych niezbędne jest zdefiniowanie mechanizmów generujących strumienie napływających zgłoszeń. Strumienie takie opisuje się zazwyczaj za pomocą tzw. *interwału*, czyli odstęp czasu między nadejściem kolejnych zgłoszeń (por. [Law i Kelton 1991]). Jeśli strumień zgłoszeń nie wykazuje zmienności, to interwał jest oczywiście stały. Gdy zgłoszenia są losowe, interwał jest również zmienną losową i należy określić jego funkcję prawdopodobieństwa.

Zgłoszenia bywają najczęściej całkowicie przypadkowe. Oznacza to przyjęcie założenia, że czas nadejścia zgłoszenia jest niezależny od poprzednich zgłoszeń, a prawdopodobieństwo zgłoszenia w przedziale dt jest proporcjonalne do długości tego przedziału. Przy tych założeniach można wykazać, że rozkład interwałów jest wykładniczy i funkcja gęstości prawdopodobieństwa interwału dana jest wzorem:

$$f(t) = \lambda e^{-\lambda t}, \quad t \geq 0 \quad (3.22)$$

Przy wykładniczym rozkładzie interwałów prawdopodobieństwo nadejścia n zgłoszeń w przedziale o długości t dane jest z kolei wzorem (rozkład Poissona):

$$P(n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \quad n = 0, 1, 2, \dots \quad (3.23)$$

gdzie λ to średnia liczba zgłoszeń w danym przedziale czasu.

Rozkład Poissona ma duże znaczenie praktyczne, gdyż opisuje on strumienie zgłoszeń (klientów, wyrobów, wniosków kredytowych, pacjentów...) w wielu rodzajach systemów. Sięga się po niego wtedy, gdy zgłoszenia przybywają do systemu pojedynczo, momenty ich pojawiania się są od siebie niezależne, zachowane jest przy tym pewne średnie tempo napływania. Rozkład ten jest również stosowany do opisu procesów obsługi. Jeżeli czas obsługi jest traktowany jako całkowicie przypadkowy, to można go przedstawić za pomocą rozkładu wykładniczego. Związany z nim rozkład Poissona określa liczbę zgłoszeń obsłużonych w danym przedziale czasu.

Algorytm generowania kolejnych momentów przybywania zgłoszeń jest bardzo prosty.

Załóżmy, że znamy wartość procesu w chwili t_{i-1} , wtedy wartość w chwili t_i uzyskujemy w następujący sposób :

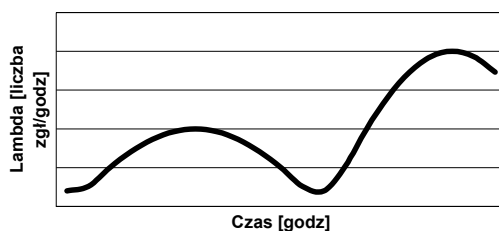
1. Wygenerujemy $u \sim U(0, 1)$ niezależną od poprzednich wartości zmiennej.
2. Wyznamy t_i za pomocą wzoru:

$$t_i = t_{i-1} + \left(-\frac{\ln u}{\lambda}\right), \quad (3.24)$$

gdzie λ jest intensywnością napływania zgłoszeń.

3.7.2. NIESTACJONARNY PROCES POISSONA

W wielu systemach rzeczywistych tempo pojawiania się zgłoszeń zmienia się jednak w czasie (pora obiadowa w barach, godziny korków ulicznych itd.; por. rys. 3.5) i zaniedbanie niestacjonarności może doprowadzić do znacznych błędów. Symulacja oferuje bardzo istotną możliwość uwzględniania niestacjonarności poprzez wprowadzenie funkcji parametru przybywania zgłoszeń $\lambda = f(t)$, która podaje różne wartości parametru λ dla różnych momentów t . W trakcie trwania symulacji kolejne zgłoszenia generowane są według parametru natężenia zgłoszeń właściwego dla danego okresu, co pozwala na dynamiczne dopasowanie liczby pojawiających się w modelu obiektów do rzeczywistego strumienia zgłoszeń.



Rys. 3.5. Zmiana tempa przybywania zgłoszeń w ciągu doby

Powyższe rozwiązanie, polegające na wprowadzeniu do modelu rozkładu wykładniczego, w którym parametr λ zmieniałby się w zależności od pory doby, jest niestety nie zawsze prawidłowe. W przykładzie (por. tabela 3.4) zmienialibyśmy parametr λ co 30 minut. Takie postępowanie przyniosłoby przybliżone rozwiązanie pod warunkiem, że tempo przybywania zgłoszeń różni się nieznacznie w poszczególnych okresach. Jeżeli jednak tempo przybywania zgłoszeń zmienia się w sposób istotny, taka metoda doprowadzi do błędnych wyników.

Tabela 3.4. Średnie tempo przybywania klientów [liczba klientów na godzinę]

Czas	Liczba klientów	Czas	Liczba klientów
14:00–14:30	95	18:00–18:30	105
14:30–15:00	100	18:30–19:00	95
15:00–15:30	120	19:00–19:30	125
15:30–16:00	150	19:30–20:00	150
16:00–16:30	160	20:00–20:30	155
16:30–17:00	150	20:30–21:00	95
17:00–17:30	160	21:00–21:30	70
17:30–18:00	110	21:30–22:00	60

Założmy, że mamy tylko dwa okresy 30-minutowe. Liczba klientów w pierwszym okresie to 3 osoby/godzinę (lub inaczej, czas między kolejnymi zgłoszeniami to około 20 minut), natomiast w drugim okresie mamy 60 osób/godzinę (czas między

dzy zgłoszeniami to 1 minuta). Załóżmy, że ostatnie zgłoszenie w pierwszym okresie miało miejsce w 29 minucie. Za pomocą pierwszego parametru wygenerowalibyśmy czas nadejścia kolejnego zgłoszenia grubo po 31 minucie. Tymczasem w drugim okresie powinno nadejść około 30 klientów. Ogólnie mówiąc, stosując tę metodę, zmniejszylibyśmy liczbę zgłoszeń przy przejściu z okresu o niskiej do wysokiej częstotliwości zgłoszeń, a zwiększylibyśmy ich liczbę przy przejściu z okresu o większej do mniejszej intensywności.

Jedną z metod, która pozwala rozwiązać problem niestacjonarności, jest metoda *rozzrzedzania* (ang. *thinning method*) (por. [Kelton i in. 2007]). Zastosowanie tej metody do przedstawionego przykładu polegałoby na:

(1) Określeniu maksymalnego tempa przybywania zgłoszeń (160 zgłoszeń/godz., czyli 60/160 minut pomiędzy pojawieniem się kolejnych zgłoszeń).

(2) Wykorzystaniu *stacjonarnego procesu Poissona* do generowania kolejnych przybyć z parametrem obliczonym w punkcie 1.

(3) Podjęciu próby „odrzućenia” wygenerowanego zgłoszenia (zgłoszenie zostało wygenerowane, ale może nie być uwzględnione w modelu). Decyzję o odrzuceniu wygenerowanego zgłoszenia podejmujemy, obliczając każdorazowo (dla każdego wyróżnionego okresu) prawdopodobieństwo pozostawienia lub odrzucenia zgłoszenia. Prawdopodobieństwo, że zgłoszenie wygenerowane z maksymalnym parametrem pojawi się w systemie jest równe ilorazowi bieżącego tempa przybywania przez tempo maksymalne.

W przykładzie jak w tab. 3.4 określilibyśmy maksymalną wartość parametru λ jako 60/160 minut, a zatem $\lambda = 0,375$. W okresie od 14:00 do 14:30 prawdopodobieństwo, że zgłoszenie pojawi się w systemie, będzie wynosiło 95/160, czyli 0,5938. Dla okresu od 14:00 do 14:30 generujemy następne zgłoszenia według maksymalnego parametru $\lambda = 0,375$, a następnie przyjmujemy lub odrzucamy wygenerowane zgłoszenie zgodnie z obliczonym wcześniej prawdopodobieństwem równym 0,5938.

Okres 14:00 - 14:30			lambda = 0,375	1/lambda = 2,7 minuty	Prawdopodobieństwo zaakceptowania = 0,5938			
Nr klienta	Aktualny czas		Liczba losowa 1	Czas do następnego klienta	Liczba losowa 2	Odrzucamy?	Moment przyjścia	
	Godzina	Minuty					Godzina	Minuty
1	14	0	0,9040285	0	0,419376102	NIE	14	0
2	14	0	0,6754707	1	0,592017926	NIE	14	1
3	14	1	0,0090797	13	0,633070022	TAK		
4	14	1	0,3826486	3	0,776745926	TAK		
5	14	1	0,5672891	2	0,203070559	NIE	14	3
6	14	3	0,0155832	11	0,557560763	NIE	14	14
7	14	14	0,264679	4	0,254256527	NIE	14	17
8	14	17	0,479819	2	0,623894562	TAK		
9	14	17	0,1271278	6	0,513992038	NIE	14	23
10	14	23	0,073399	7	0,733108856	TAK		

Rys. 3.6. Ilustracja zastosowania metody rozrzedzania

Źródło: opracowanie własne

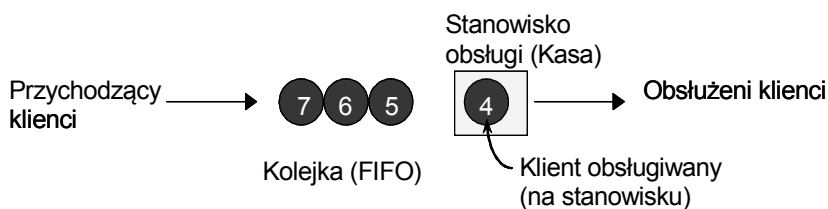
Ilustrację metody rozrzedzania dla przykładu z tabeli 3.4 prezentuje rys. 3.6. W kolejnych wierszach generowany jest czas, jaki upłynie do nadejścia kolejnego klienta z wykorzystaniem liczby losowej nr 1 i formuły dla rozkładu wykładniczego $t = (-1/\lambda)\ln(u_1)$ – porównaj kolumna *Czas do następnego klienta*. Każdorazowo, poprzez wygenerowaną liczbę losową numer 2, sprawdzamy, czy możemy zaakceptować nowe zgłoszenie. Jeżeli $u_2 < 0,5938$, pozostawiamy wygenerowane zgłoszenie i wyznaczamy czas jego pojawienia się w modelu. W przeciwnym razie zgłoszenie odrzucamy i generujemy kolejne zgłoszenie.

Oczywiście, w kolejnym badanym przedziale czasu, prawdopodobieństwo decydujące o przyjęciu lub odrzuceniu zgłoszenia będzie inne: przykładowo, w godzinach od 14:30 do 15:00 będzie wynosiło $100/160$, czyli $0,625$.

4. SYMULACJA ZDARZEŃ DYSKRETNYCH

4.1. PRZYKŁAD WPROWADZAJĄCY

Omawianie zagadnień związanych z symulacją dyskretną rozpoczniemy od prostego przykładu (rys. 4.1).



Rys. 4.1. Model prostego systemu obsługowego

Klienci przybywają do systemu (sklep, bankomat, biuro obsługi, stacja benzynowa, przychodnia...), w którym znajduje się jedno stanowisko obsługi. Po wejściu do systemu klient może albo natychmiast znaleźć się na stanowisku obsługi i zrealizować cel swojej wizyty (kupić mleko i bułki, pobrać pieniądze z bankomatu, wykupić wczasy, zatankować paliwo, ...), albo niestety będzie musiał stanąć w kolejce i (mniej lub bardziej cierpliwie) oczekiwać, aż wszyscy klienci stojący przed nim zostaną obsłużeni. Kolejki mogą funkcjonować według różnych zasad. Najczęściej mamy do czynienia z kolejką FIFO (*First In First Out: Pierwszy Przyszedł Pierwszy Obsłużony*), ale nie jest to oczywiście jedyna możliwa reguła obsługi kolejki. W przychodni może na przykład obowiązywać zasada „według listy”, czyli zgodnie z pewnym wcześniej ustalonym porządkiem; towar z półek hurtowni może być wydawany przy zastosowaniu zasady LIFO (*Last In First Out: Ostatni Przyszedł Pierwszy Obsłużony*); karetka pogotowia przyjedzie w pierwszej kolejności do najcięższej poszkodowanej osoby (obsługa według priorytetów) itd.

W chwili, kiedy klient znajdzie się na stanowisku obsługi, zostanie ono „zablokowane”, czyli stanie się niedostępne dla innych klientów. Niedostępność stanowiska obsługi będzie trwała do momentu, w którym zakończy się obsługa „naszego” klienta.

Wtedy klient opuści stanowisko (w przypadku opisanego powyżej prostego systemu klient opuści również system) i możliwa będzie obsługa kolejnego klienta.

Model symulacyjny będzie miał na celu zbadanie sprawności działania systemu. Będą nas interesowały przede wszystkim tworzące się kolejki (średnie długości kolejek, czas oczekiwania, wartości ekstremalne), obciążenie stanowisk obsługi, przepustowość systemu. O badanych systemach powiemy, że są to *systemy kolejkowe*, natomiast podejście symulacyjne, które wykorzystamy nazwiemy *symulacją dyskretną*. Dlaczego *systemy kolejkowe*? Jest to chyba oczywiste w świetle podanego przykładu. *Symulacja dyskretna* umożliwi nam natomiast skupienie się na pojedynczych zdarzeniach, takich jak pojawienie się klienta w systemie, ustawienie się w kolejce, rozpoczęcie obsługi na stanowisku obsługi itd.

4.2. PODSTAWOWE ELEMENTY DYSKRETNEGO MODELU SYMULACYJNEGO

Zgłoszenia

W dyskretnych modelach symulacyjnych musimy przede wszystkim wskazać (zdefiniować) *zgłoszenia* (ang. *entities*), czyli obiekty dynamiczne, które będą przemieszczały się przez modelowany system. Zgłoszenia to inaczej „gracze”, którzy poruszają się po modelu, zmieniają się, oddziałują (i wzajemnie) na inne obiekty (zgłoszenia), jak również na elementy samego modelu. Zgłoszenia muszą być *wykreowane* na początku symulacji, następnie *przemieszczane są* przez model i na koniec *niszczone*, po uprzednim zebraniu rozmaitych statystyk związanych z ich pobytom w systemie. W modelu może występować więcej niż jeden typ zgłoszeń (pacjenci lżej i ciężiej chorzy, klienci którzy przyszli tylko pooglądać towar i tacy, którzy robią całotygodniowe zakupy, samochody tankujące benzynę i takie, które napędzane są gazem, czytelnicy którzy wypożyczają/oddają książki i tacy, którzy uiszczają kary za zwłokę). Zgłoszenia mogą reprezentować „rzeczywiste” elementy (np. „klienci” w omawianym przykładzie) lub stanowić przenośnię, która w świecie rzeczywistym nie występuje (np. podstępny chochlik psujący komputer).

Zgłoszenia pojawiają się w systemie zgodnie z pewnym *schematem*: deterministycznym lub losowym. Schemat przybywania zgłoszeń podawany jest najczęściej w postaci *czasu, który upływa pomiędzy nadejściem kolejnych zgłoszeń* (ang. *inter-arrival time*). Jeżeli zgłoszenia napływają w równych odstępach czasu, mamy do czynienia ze schematem *deterministycznym* (wygodnym, ale niestety rzadko dającym się zaobserwować w rzeczywistości). Nieregularny, losowy napływ zgłoszeń jest natomiast zwykle przedstawiany za pomocą *rozkładu losowego*.

Zgłoszenia mają pewne *cechy*, własności, dzięki którym możemy je różnicować, dzielić na grupy, identyfikować w modelu. Własności te zwane są *atrybutami*. Atrybuty charakteryzują wszystkie obiekty (opisują, różnicują), przy czym obiekty jednego typu mają ten sam zestaw atrybutów, ale o różnych wartościach: czas przybycia klienta do banku, priorytet obsługi, rodzaj wykonywanej operacji... **Atrybut jest nierozdzielnie związany z obiektem**, co oznacza, że w trakcie symulacji każde zgłoszenie przemieszczając się przez model „unoszą” ze sobą wszystkie swoje atrybuty, oczywiście wraz z wartościami. Podobnie dzieje się przecież w systemie rzeczywistym z prawdziwymi obiektami: student zmierzający w stronę wypożyczalni „niesie” (i to dosłownie) zestaw książek do oddania, jak również atrybut charakteryzujący jego staż czytelnicy, historię wypożyczeń, plany dotyczące kolejnych wypożyczeń, cierpliwą/niecierpliwą naturę, która każe mu przez godzinę stać w bardzo długiej kolejce lub opuścić wypożyczalnię po 15 minutach czekania.

Zasoby

Innym, bardzo ważnym elementem systemu kolejkowego i dyskretnego modelu symulacyjnego są *zasoby* (stanowiska obsługi, ang. *resources*). O dostęp do zasobów „walczą” zgłoszenia: zasobami mogą być ludzie (urzędnicy w banku), sprzęt (maszyny na hali produkcyjnej), przestrzeń (podjazdy na stacji benzynowej). Zgłoszenie *zajmuje* zasób, a następnie po pewnym czasie *zwalnia* go. **To zgłoszenie „rządzi” zasobem, a nie odwrotnie** (zgłoszenie potrafi zająć kilka zasobów jednocześnie!). Zasób natomiast może mieć różną *pojemność* (liczba miejsc przy stolikach w restauracji, liczba kas w supermarkecie itd.), a co więcej, pojemność zasobu może się zmieniać w trakcie symulacji (stanowiska kasowe otwierane i zamykane w trakcie pracy banku, liczba lekarzy na dyżurze w szpitalu). Czas trwania obsługi może być, podobnie jak schemat napływania klientów, opisany deterministycznie lub losowo za pomocą rozmaitych rozkładów losowych.

Kolejki

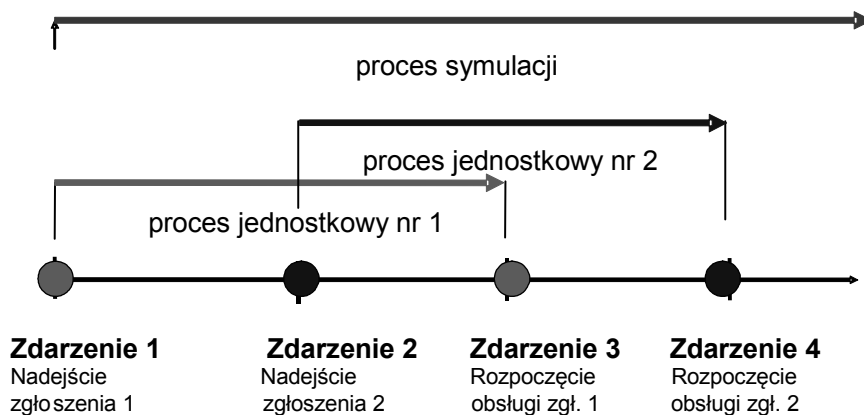
Ważnym elementem omawianej grupy systemów są oczywiście *kolejki*. Kolejki (rzeczywiste lub wirtualne) to miejsca, w których zgłoszenia czekają w przypadku, gdy nie mogą przemieszczać się dalej. Kolejki mogą być fizycznie zlokalizowane (pacjenci czekający na wizytę u lekarza) lub mogą istnieć wyłącznie w postaci pewnego logicznego uszeregowania (pacjenci oczekujący na przeszczep serca). Kolejki są zwykle związane ze stanowiskami obsługi – to właśnie ograniczony dostęp do zasobów powoduje, że zgłoszenia nie zawsze mogą być natychmiast obsługiwane i muszą oczekiwać na zwolnienie stanowiska. Kolejki mogą mieć ograniczoną pojemność (np. podjazd do stacji benzynowej, liczba miejsc w poczekalni) lub ich długość może być teoretycznie nieograniczona (liczba osób oczekujących na połączenie telefoniczne z konsultantem). Jak już wspomniano wcześniej, zasady wyboru zgłoszeń z kolejki opisane są różnymi regułami: FIFO, LIFO, wg priorytetów, wg najmniejszej odległości, najkrótszego czasu dojazdu itd.

Statystyki

Pod koniec procesu symulacji zbierane są rozmaite *statystyki*, które pozwalają ocenić działanie systemu. Statystyki uaktualniają się w ciągu całego procesu symulacji, a ich ostateczna wartość naliczana jest po zakończeniu eksperymentu. Najczęściej obliczane statystyki w symulacji dyskretniej to: liczba obsłużonych klientów, liczba klientów, którzy musieli czekać w kolejce, najdłuższa kolejka, najdłuższy czas pobytu w kolejce, najdłuższy czas pobytu w systemie, wykorzystanie stanowiska obsługi, liczba zajętych stanowisk obsługi i inne.

4.3. OPERACJE NA ZGŁOSZENIACH

Przebieg procesu symulacji jest ściśle związany ze zmianami, których doświadczają zgłoszenia. Kluczowym pojęciem jest tu *zdarzenie*, które oznacza *zmianę w stanie systemu* (por. [Fishman 1981]). Zmiana może dotyczyć albo wartości wyróżnionych atrybutów (zgłoszeń lub zasobów), albo wprowadzenia/zniszczenia zgłoszenia. Zdarzeniem będzie zatem np. wejście nowego klienta do banku (wprowadzenie obiektu), rozpoczęcie obsługi przy okienku kasowym (zmiana atrybutu zasobu z *wolny* na *zajęty*), opuszczenie banku przez klienta (zniszczenie obiektu). Zdarzenie zachodzi zawsze w jednej chwili (w danym momencie) i dlatego w dyskretnym modelu symulacyjnym zmiana stanu systemu dokonuje się jedynie w dyskretnych momentach czasu.



Rys. 4.2. Zdarzenia, procesy jednostkowe, proces symulacji
Źródło: [Fishman 1981]

Działanie jest wynikiem oddziaływania pomiędzy zgłoszeniami i zasobami, które doprowadza do pojawienia się zdarzenia. Działanie rozpoczyna się i kończy zdarzeniem, a jego długość jest najczęściej określona poprzez wartość uzyskaną z rozkładu

losowego. Inaczej – *działanie* jest zbiorem operacji, w wyniku których ulegają zmianie stany systemu (mają miejsce zdarzenia). Działanie trwa przez pewien czas, np. pobieranie pieniędzy z bankomatu, skanowanie produktów przy kasie w supermarkecie, podpisywanie wniosku kredytowego w banku itd.

Proces to chronologicznie uporządkowany zbiór zdarzeń i działań związanych z każdym pojedynczym zgłoszeniem od chwili jego pojawienia się w systemie do chwili zaniku (por. rys. 4.2). W modelu terminalu lotniczego proces związany z jednym pasażerem będzie rozpoczynał się pojawieniem się podróżnego na lotnisku, a następnie będzie obejmował oczekiwanie w kolejce w celu nadania bagażu, przejście przez bramkę bezpieczeństwa, zweryfikowanie dokumentu podróznego, kontrolę bagażu podręcznego, oczekiwanie w hali odlotu na wezwanie do zajmowania miejsc w samolocie. Modelowanie pojedynczego procesu nie wydaje się skomplikowane, jednak przez lotnisko przemieszcza się setki, jeśli nie tysiące pasażerów, którzy oczekują na te same usługi i są obsługiwani na tych samych stanowiskach obsługi. Przemieszczanie się jednego pasażera zależy od tego, co dzieje się z pozostałymi podróżnymi.

4.4. REALIZACJA PROCESU SYMULACJI

4.4.1. ZEGAR SYMULACJI

W symulacji dynamicznej, tzn. takiej, w której uwzględnia się wpływ czasu na zmiany zachodzące w systemie (w omawianej w rozdziale 7 metodzie Monte Carlo wpływ czasu jest najczęściej pomijany), kluczowym zagadnieniem jest odwzorowanie w modelu upływu czasu. Czas jest rejestrowany jako zmienna zwana *czasem symulacji* i zapamiętywany w obiekcie zwanym *zegarem symulacji* (lub po prostu *zegarem*). Na początku symulacji *zegar* ustawia się na wartość zero i od tej pory wskazuje on liczbę jednostek czasu symulowanego, które upłynęły od chwili początkowej. Określenie *czas symulacji* oznacza czas zegarowy według wskazań zegara symulacji, a nie rzeczywisty czas wykonywania obliczeń. Czynnikiem decydującym o czasie obliczeń jest liczba zdarzeń występujących w trakcie symulacji.

Uaktualnianie *zegara* (czyli sterowanie upływem czasu) jest bardzo ściśle związane ze zdarzeniami, które pojawiają się w modelu. Istnieją dwie podstawowe metody uaktualniania czasu zegarowego (por. [Gordon 1974]). Według pierwszej, czas zegarowy jest ustawiany na chwilę, w której ma wystąpić zdarzenie (tzw. metoda *kolejnych zdarzeń*). W drugiej metodzie, zegar zwiększa się o stały przyrost czasu i każdorazowo sprawdza się, czy jest to czas wystąpienia jakiegoś zdarzenia (metoda *stałego kroku*). W obu podejściach zegar wymusza chronologiczne porządkowanie zdarzeń.

Metoda kolejnych zdarzeń pozwala na nieuwzględnianie okresów braku aktywności systemu. Po zajściu wszystkich zmian stanu w czasie odpowiadającym danemu zdarzeniu, czas symulacyjny zostaje przesunięty do chwili zajścia kolejnego zdarzenia, w której zachodzą następne zmiany stanu (odpowiadające kolejnemu zdarzeniu). W ten sposób okres braku aktywności systemu zostaje całkowicie pominięty.

Symulację systemów dyskretnych wykonuje się zwykle według metody kolejnych zdarzeń, natomiast symulację systemów ciągłych – metodą stałego kroku. Ogólnie rzecz biorąc, system ciągły modelujemy za pomocą ciągłych równań, opisujących zmiany atrybutów w czasie. Opis systemów dyskretnych związany jest ze zdarzeniami wywołującymi zmiany jego stanu.

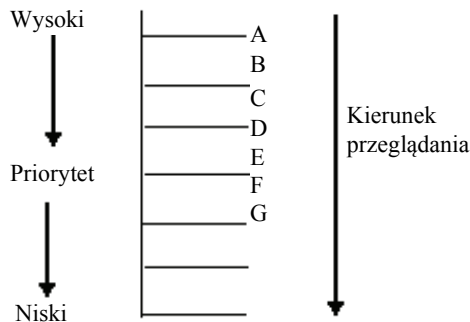
4.4.2. ZDARZENIA WARUNKOWE I BEZWARUNKOWE, ZAWIESZONE I RÓWNOCZESNE

Istnieją dwa typy zdarzeń: *zdarzenia bezwarunkowe*, które są bezpośrednio zależne od czasu oraz *zdarzenia warunkowe*, które pośrednio zależą od czasu i wyznaczone są stanami systemu. Pojawienie się zdarzenia bezwarunkowego nie jest uzależnione od jakichkolwiek warunków panujących w modelu – zdarzeniem bezwarunkowym jest np. nadejście nowego klienta. Zdarzenia warunkowe zaistnieją natomiast tylko wtedy, gdy pozwoli im na to stan systemu – np. rozpoczęcie obsługi na stanowisku jest możliwe tylko wtedy, gdy stanowisko jest wolne, a inne zgłoszenia nie oczekują w kolejce.

W dyskretnych modelach symulacyjnych należy wziąć pod uwagę możliwość jednoczesnego zajścia dwóch lub więcej zdarzeń (tzw. *zdarzenia równoczesne*). Jakkolwiek oznacza to, że zdarzenia te zachodzą w systemie w tej samej chwili, to jednak program symulacyjny będzie obsługiwał je kolejno. Kolejność obsługi może zostać wymuszona priorytetami, według których zdarzenia będą wybierane. Jeżeli nie ustalono natomiast odrębnej zasady porządkowania zdarzeń, wybór zdarzeń dokonany zostanie losowo.

Jeżeli zdarzenie nie może być wykonane w zaplanowanym czasie, to określa się je jako *zawieszona*. Zapis dotyczący takiego zdarzenia musi być przechowywany aż do momentu zmiany blokujących warunków. Zapisy takie umieszcza się często na specjalnej liście zwanej *listą zdarzeń zawieszonych*. Program powinien przeglądać tę listę po każdej zmianie stanu systemu w celu określenia, czy nowe warunki sprzyjają uaktywnieniu któregoś ze zdarzeń. Może tu zajść reakcja łańcuchowa: przetworzenie jednego zdarzenia może odwieść w tej samej chwili czasu zegarowego jedno lub więcej zdarzeń, które z kolei odwieszają następne. Prowadzi to do omówionego zagadnienia zdarzeń równoczesnych.

Rozważmy sytuację przedstawioną na rysunku 4.3, prezentującą listę zdarzeń zawieszonych. Zapisy rozmieszczone są w porządku priorytetów i przeglądane są od priorytetu najwyższego do najniższego.



Rys. 4.3. Przeglądanie zdarzeń zawieszonych
Źródło: [Gordon 1974]

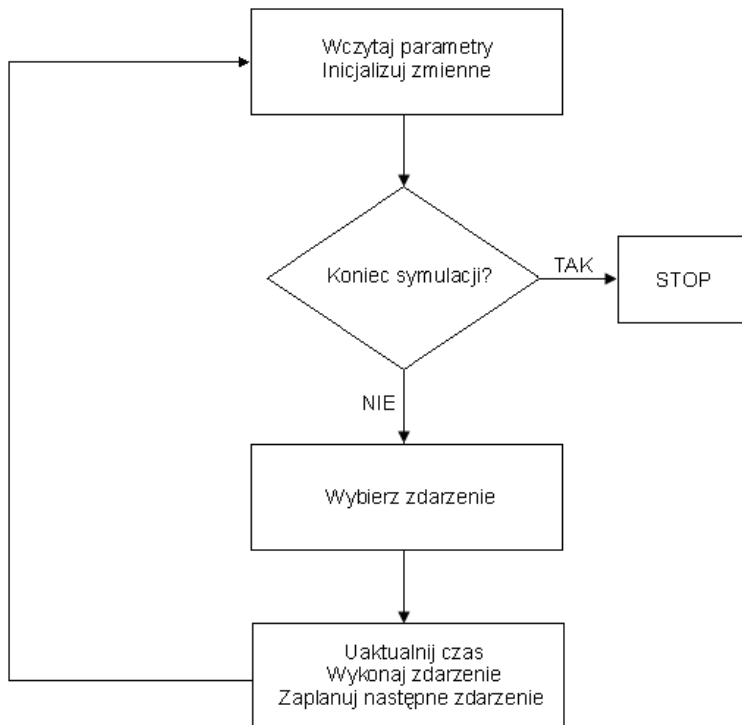
Gdy w danej chwili zostały zakończone wszystkie możliwe czynności, uaktualniany jest czas zegarowy, który zaczyna wskazywać czas następnego (przyszłego) zdarzenia. Po aktualizacji czasu zegarowego do listy dołącza się nowe zdarzenie według priorytetu. Przeglądanie listy zdarzeń zawieszonych rozpoczyna się od zapisu o najwyższym priorytecie. Po napotkaniu informacji o zdarzeniu, które może być zrealizowane, wykonuje się wszystkie czynności o zerowym czasie trwania związane z danym zdarzeniem. Przeglądanie ponawiane jest od początku listy. Po przetworzeniu wszystkich zdarzeń możliwych w danej chwili program znowu uaktualnia czas zegarowy.

Wielokrotne przeglądanie zdarzeń bywa bardzo czasochłonne. Proces ten można usprawnić, jeżeli istnieje sposób rozpoznawania działania, którego wynikiem będzie odwieszenie zdarzenia. Takie zdarzenie może być wtedy uczynione nieaktywnym albo przez usunięcie go z listy zdarzeń zawieszonych, albo przez wprowadzenie odpowiedniej informacji bezpośrednio do zapisu zdarzenia. Zdarzenie będzie wtedy podczas przeglądania pomijane aż do zmiany wskaźnika.

4.4.3. KALENDARZ ZDARZEŃ I OBSŁUGA PROCESU SYMULACJI

Informacje o wszystkich planowanych zdarzeniach przechowywane są w *kalendharzu zdarzeń*. Jest to obiekt dynamiczny, który steruje upływem czasu w trakcie symulacji. Terminy wystąpienia wszystkich zaplanowanych zdarzeń przechowywane są w nim w sposób uporządkowany według dnia i godziny. *Kalendarz* dopuszcza też możliwość przechowywania zdarzeń o tym samym terminie realizacji. Są one realizowane sekwencyjnie, w kolejności zapisu do kalendarza. W miarę przebiegu procesu symulacji, informacje zawarte w *kalendharzu zdarzeń* są uaktualniane, tzn. wykreślone są z niego dane dotyczące zdarzeń już zrealizowanych (lub będących w trakcie realizacji), a dopisywany jest czas wystąpienia nowych zdarzeń. Pobranie zdarzenia z *kalendharza* powoduje przesunięcie zegara symulacji na godzinę odpowiadającą terminowi realizacji zdarzenia. W zależności od typu pobranego zdarzenia inicjowany jest odpowiedni proces jednostkowy.

Proces symulacji dyskretnych przebiega od zdarzenia do zdarzenia, a jego kluczowym elementem jest właśnie *kalendarz*, którego zadaniem jest przechowywanie planowanych zdarzeń i sterowanie zegarem symulacji. Sposób pobierania zdarzeń z kalendarza może być jednak różnie zorganizowany i w odmienny sposób mogą być wykonywane procedury logiczne, będące wynikiem realizowania zdarzeń w modelu. Wyróżnia się (por. [Gordon 1974], [Seila i in. 2003], [Kelton i in. 2007]) *metodę planowania zdarzeń*, *metodę wyboru działań* i *metodę interakcji procesów*. Podstawą różnicowania metod symulacji dyskretnych jest wybór jednego z trzech typów elementów (zdarzenia, procesu, działania) i uczynienie z niego centralnego typu obiektów, z punktu widzenia którego planuje się strukturę modelu.



Rys. 4.4. Schemat logiczny procesu symulacji dyskretnych

Schemat logiczny procesu symulacji przedstawiono na rys. 4.4. Na początku następuje wczytanie lub zainicjowanie parametrów systemu oraz zmiennych symulacyjnych. Następnie rozpoczyna się symulacja realizowana w sposób iteracyjny. Najpierw określamy, czy symulacja powinna być dalej wykonywana. Jeżeli nie zaistniały warunki kończące symulację, to z *kalendarza zdarzeń* pobierane jest zdarzenie, które powinno być zrealizowane w pierwszej kolejności. Czas symulacji uaktualniany jest na czas realizacji zdarzenia pobranego z *kalendarza*. Zdarzenie jest następnie wykonywane, tzn. realizowane są czynności związane z wystąpieniem tego typu zdarzenia. Jeżeli z poja-

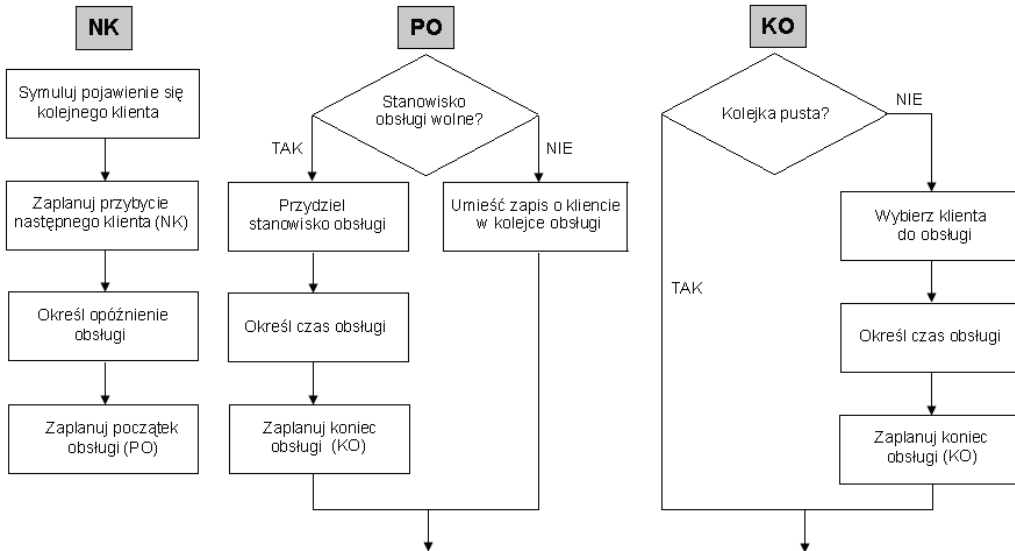
wieniem się zdarzenia wiążą się inne, przyszłe zdarzenia, czasy wystąpienia tych zdarzeń muszą zostać zaplanowane, a informacje o tym zapisane w *kalendariuszu zdarzeń*.

Decyzja o kontynuowaniu procesu symulacji może być związana z zaplanowanym czasem obserwacji, ale można też wybrać inne warunki końcowe: np. określić liczbę klientów, którzy będą obsłużeni lub maksymalną liczbę klientów, która może być odnotowana w systemie.

4.5. TECHNIKI MODELOWANIA ZDARZEŃ DYSKRETYCH

4.5.1. METODA PLANOWANIA ZDARZEŃ

W metodzie *planowania zdarzeń* określa się szczegółowy opis czynności, które są realizowane po zajściu każdego typu zdarzenia (por. rys. 4.5). Do kalendarza zdarzeń wprowadza się zdarzenia bezwarunkowe (bezpośrednio zależne od czasu) każdorazowo, przy rozpatrywaniu czynności przypisanych do zdarzenia poprzedzającego. Czynności zdarzeń warunkowych są natomiast rozpatrywane jedynie przy okazji realizowania czynności związanych z wystąpieniem zdarzeń bezwarunkowych.



Rys. 4.5. Koncepcja planowania zdarzeń. *NK* – zdarzenie Nadejście Nowego Klienta; *PO* – zdarzenie Początek Obsługi Klienta; *KO* – zdarzenie Koniec Obsługi

W przykładzie pokazanym na rysunku 4.5 zdefiniowano trzy typy zdarzeń: *NK* (Nadejście Nowego Klienta), *PO* (Początek Obsługi Klienta) i *KO* (Koniec Obsługi Klienta). Zdarzenie *NK* planowane jest w momencie nadejścia klienta wcześniejszego,

czyli wystąpienia poprzedniego zdarzenia *NK*; zdarzenie *PO* określane jest w momencie skierowania na stanowisko kolejnego klienta; *KO* planowane jest w momencie rozpoczęcia obsługi na stanowisku, czyli w momencie aktywacji działania obsługi klienta. Należy podkreślić, że sposób definiowania typów zdarzeń jest każdorazowo zależny od modelowanego systemu. Wyróżnianie zdarzeń *PO* i *KO* stosuje się zazwyczaj wtedy, kiedy między zwolnieniem stanowiska a rozpoczęciem obsługi obserwowane jest opóźnienie. W przypadku kiedy po zwolnieniu stanowiska obsługa kolejnego zgłoszenia może rozpocząć się natychmiast, można zdefiniować zdarzenie typu *KO* (por. przykład 2 w podrozdziale 4.5.5).

4.5.2. METODA PRZEGLĄDU I WYBORU DZIAŁAŃ

W metodzie *przeglądu i wyboru działań* uwaga skupia się na działaniach. Ponieważ jednak rozpoczęcie działania uwarunkowane jest wystąpieniem zdarzenia, w metodzie tej rozpatruje się jednocześnie zdarzenia i działania.

Zdarzenia bezwarunkowe i warunkowe (por. podrozdz. 4.4.2) obsługiwane są w metodzie przeglądu i wyboru działań przez odrębne procedury logiczne. Zdarzenia warunkowe (*zd-W*) mogą zaistnieć tylko wtedy, gdy spełnione zostaną określone warunki. Zdarzenia bezwarunkowe (*zd-B*) muszą zaistnieć w przewidzianym czasie. Najczęściej *zd-B* związane są z przybyciem nowego zgłoszenia i zakończeniem obsługi, natomiast *zd-W* z rozpoczęciem obsługi, co może być wykonane tylko wtedy, gdy spełnione zostaną warunki pozwalające na uruchomienie działania.

Proces symulacji polega na realizacji trzech faz (por. rys. 4.6). W fazie A następuje pobranie zdarzenia (bezwarunkowego) z Kalendarza i uaktualnienie zegara symulacji. W tej fazie pobrane z kalendarza zdarzenie **nie jest jeszcze** wykonywane. W fazie B pobrane zdarzenie *zd-B* jest wykonywane, czyli realizowane są wszystkie czynności przewidziane dla tego zdarzenia. W fazie C przeglądane są tylko zdarzenia warunkowe (*zd-W*), w celu sprawdzenia, czy na skutek zmiany warunków możliwe jest wykonanie jednego lub kilku z nich.

```
Dopóki nie nastąpi koniec symulacji wykonuj
  Faza A: Pobierz kolejne zdarzenie z Kalendarza
          Uaktualnij czas
  Faza B: Wykonaj wszystkie zdarzenia bezwarunkowe zapla-
          nowane na ten moment
  Faza C: Przeglądaj kolejno wszystkie zdarzenia warunkowe
          Jeżeli spełnione są warunki pozwalające na wyko-
          nane jakiegoś Zd-W, wykonaj to zdarzenie i po-
          wtórz Fazę C
Koniec symulacji
```

Rys. 4.6. Koncepcja wyboru działań

Metoda przeglądu i wyboru działań polega zatem na rozpatrywaniu wszystkich działań w celu określenia, które z nich, z chwilą zajścia określonego zdarzenia, powinny być rozpoczęte, a które zakończone. Z każdym działaniem związane są dwie procedury logiczne: *testowania warunków* i *realizacji działania*. Procedura testowania sprawdza, czy spełnione są warunki umożliwiające rozpoczęcie działania, a procedura realizacji opisuje zmiany, które zachodzą w systemie w następstwie realizacji działania.

Metoda przeglądu i wyboru działań jest prowadzona przez:

- wprowadzenie do *kalendacza* zdarzeń bezwarunkowych oraz zdarzeń warunkowych,
- rozpatrywanie czynności związanych ze zdarzeniami warunkowymi niezależnie od wystąpienia zdarzeń bezwarunkowych,
- sterowanie przebiegiem symulacji w oparciu o listę warunków, jakie powinny być spełnione, by zaszło w systemie określone zdarzenie.

Przewaga koncepcji wyboru działań nad koncepcją planowania zdarzeń pojawia się wtedy, gdy system charakteryzuje się małą liczbą zgłoszeń i dużą liczbą działań. Jeżeli liczba działań rośnie, to – w koncepcji planowania zdarzeń – rośnie liczba zaplanowanych zdarzeń a wraz z nimi rośnie czas pracy komputera wydatkowany na tworzenie zapisów zdarzeń, rejestrację w *kalendarzu*, wybieranie do realizacji oraz wymazywanie zapisów.

4.5.3. METODA INTERAKCJI PROCESÓW

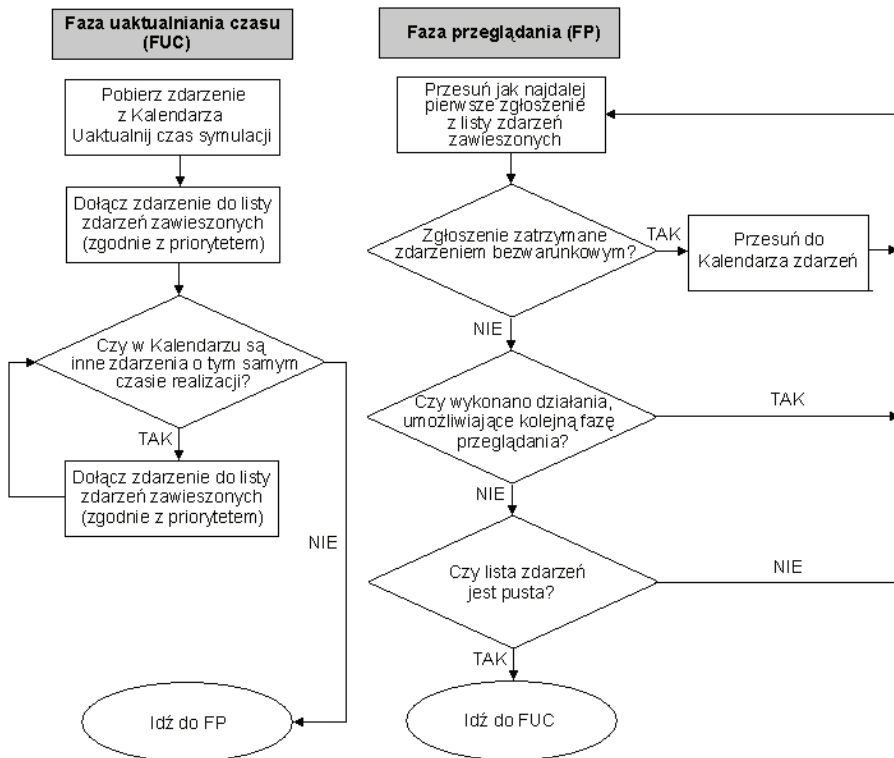
Metoda interakcji procesów polega na grupowaniu działań w procesy wykonywane na pojedynczych dynamicznych obiektach (zgłoszeniach) i rejestracji ich stanu od chwili pojawienia się w systemie do chwili zaniku (por. rys. 4.7). Zakłada się, że sposób zachowania się systemu jest opisany za pomocą zbioru procesów, z których niektóre mogą się pokrywać, przy czym *każdy z tych procesów dotyczy osobnego zgłoszenia*. W modelu jest zatem tyle procesów, ile pojawiających się w nim zgłoszeń. W omawianej koncepcji położony jest nacisk na interakcje między tymi procesami. Uwaga skupiona jest na zgłoszeniach i śledzi się przejście każdego poszczególnego zgłoszenia przez system. W trakcie przemieszczania się zgłoszenie może napotkać warunki blokujące typu bezwarunkowego (czekaj do chwili) lub warunkowego (czekaj, aż spełniony będzie warunek X). Termin czekania bezwarunkowego odpowiada planowaniu w koncepcji planowania zdarzeń.

Wszystkie zdarzenia zaplanowane (bezwarunkowe) oraz zdarzenia warunkowe umieszczane są w *kalendarzu zdarzeń*. Po wykonaniu (w określonym momencie) zdarzenia zaplanowanego badany jest wykaz zdarzeń warunkowych, z którego wybierane są zdarzenia możliwe do wykonania. Czynność powrotu do obiegu schematu kontynuowana jest do chwili, gdy nie można już wykonać żadnego dalszego zdarzenia zaplanowanego lub warunkowego. Po tym fakcie czas symulacyjny zostaje przesunięty do czasu wystąpienia następnego zaplanowanego zdarzenia i cała procedura zostaje

powtórzona. W rzeczywistości tworzone są dwie uporządkowane chronologicznie listy zdarzeń: dla zdarzeń *zaplanowanych (bezwarunkowych)*, ta lista zwana jest *kalendarem zdarzeń* i *zawieszonych (warunkowych)*. Proces symulacji polega na wykonywaniu dwóch faz: *fazy przeglądania (FP)* i *fazy uaktualniania czasu (FUC)*.

Metoda interakcji procesów polega na:

- grupowaniu zdarzeń bezwarunkowych i warunkowych w procesy równoległe związane ze zgłoszeniami,
- synchronizacji poszczególnych procesów jedynie poprzez zdarzenia,
- rozpatrywaniu zdarzeń warunkowych niezależnie od zdarzeń bezwarunkowych,
- sterowaniu procesem symulacji w oparciu o kalendarz zdarzeń zawierający zapisy o zdarzeniach bezwarunkowych.



Rys. 4.7. Koncepcja interakcji procesów

4.5.4. REALIZACJA PROCESU SYMULACJI WG METODY KOLEJNYCH ZDARZEŃ – PRZYKŁAD NR 1

Do punktu obsługi XYZ wchodzi klient i albo podchodzi natychmiast do pracownika (gdy ten aktualnie nikogo nie obsługuje), albo siada na krzeselkach i oczekują

na swoją kolej. Do przeprowadzenia symulacji będą nam potrzebne dane na temat momentów pojawienia się kolejnych klientów oraz czasu obsługi klientów przez pracownika punktu (por. tab. 4.1). Pierwszy rodzaj informacji jest podawany w postaci czasu *między nadejściem kolejnych klientów* (por. podrozdz. 3.7).

Tabela 4.1. Dane do przykładu

Kolejni klienci	Czas pomiędzy nadejściem kolejnych klientów [min]	Czas obsługi [min]
1	4	3
2	6	5
3	3	6
4	6	4
5	1	7
6	9	3
7	2	6
8	3	7
9	9	2
10	4	6
11	4	3
12	7	8

W dalszym kroku musimy wyróżnić zmienne stanu i typy zdarzeń:

Zmienne stanu:	n	= liczba klientów w systemie
	nq	= liczba klientów oczekujących na obsługę
Zdarzenia:	NK	= nadejście klienta
	PO	= rozpoczęcie obsługi
	KO	= zakończenie obsługi
Opóźnienia:	$t(a)$	= czas do momentu nadejścia następnego klienta
	$T(b)$	= czas obsługi klienta

Symulacja rozpoczyna się w czwartej jednostce czasu (tab. 4.2, wiersz nr 1), ponieważ jak wynika z tabeli z danymi (tab. 4.1, klient nr 1), od momentu startu do momentu nadejścia pierwszego klienta upływa 4 minuty. Pierwsze zdarzenie, czyli $NK1$ (Nadejście Klienta nr 1) pojawia się zatem w czwartej minucie, zmienne stanu n , nq przyjmują wartość 1. Ponieważ stanowisko obsługi jest wolne, w tej samej minucie (czwartej) pojawia się zdarzenie $PO1$ (Początek Obsługi Klienta nr 1) i zmienna stanu $nq = 0$. Z tabeli z danymi (tab. 4.1) wynika, że kolejny klient (klient nr 2) przybędzie dopiero w dziesiątej minucie, natomiast wcześniej (w siódmej minucie) zakończy się obsługa klienta nr 1 (czas obsługi klienta nr 1 wynosi 3 minuty). Pierwsze pojawi się zatem zdarzenie $KO1$ (Koniec Obsługi klienta nr 1), powodując zmianę wartości zmiennej stanu n ($n = 0$). W dziesiątej minucie przybywa klient nr 2 (zdarzenie $NK2$), a symulacja trwa aż do spełnienia warunku zakończenia (w przykładzie jest to obserwacja przez 35 minut).

Zauważmy (por. tab. 4.2), że zdarzenie (*NK*, *PO* lub *KO*) następuje zawsze, kiedy zmienia się co najmniej jedna ze zmiennych stanu (*n* lub *n_q*).

Tabela 4.2. Realizacja procesu symulacji

Numer wiersza	Wskazanie zegara [min]	Zdarzenie	Zmienna stanu <i>n</i>	Zmienna stanu <i>n_q</i>
1	4	<i>NK</i> 1	1	1
2	4	<i>PO</i> 1	1	0
3	7	<i>KO</i> 1	0	0
4	10	<i>NK</i> 2	1	1
5	10	<i>PO</i> 2	1	0
6	13	<i>NK</i> 3	2	1
7	15	<i>KO</i> 2	1	1
8	15	<i>PO</i> 3	1	0
9	19	<i>NK</i> 4	2	1
10	20	<i>NK</i> 5	3	2
11	21	<i>KO</i> 3	2	2
12	21	<i>PO</i> 4	2	1
13	25	<i>KO</i> 4	1	1
14	25	<i>PO</i> 5	1	0
15	29	<i>NK</i> 6	2	1
16	31	<i>NK</i> 7	3	2
17	32	<i>KO</i> 5	2	2
18	32	<i>PO</i> 6	2	1
19	34	<i>NK</i> 8	3	2

4.5.5. REALIZACJA PROCESU SYMULACJI WG METODY KOLEJNYCH ZDARZEŃ – PRZYKŁAD NR 2

W telefonicznym biurze informacji pracuje jeden pracownik odbierający telefony i udzielający informacji. Model dyskretny zostanie zdefiniowany za pomocą dwóch typów zdarzeń *Przyb* (Przybycie nowego zgłoszenia telefonicznego) oraz *Zak* (Zakończenie obsługi). W modelu może tworzyć się kolejka zgłoszeń oczekujących na połączenie z konsultantem. Dalej prześladzony zostanie proces planowania zdarzeń i realizacji symulacji według metody kolejnych zdarzeń.

Zdarzenie: Przybycie (*Przyb*) nowego klienta do systemu

1. Zapamiętaj czas (moment) przybycia klienta
2. Jeżeli Telefonistka jest „wolna”:
 - 2.1. rozpocznij obsługę (określ czas zakończenia obsługi – zdarzenie *Zakończenie*)
 - 2.2. oznacz Stanowisko jako *zajęte*
3. Jeżeli Telefonistka jest „zajęta”:
 - 3.1. wstaw klienta do kolejki
 - 3.2. zwiększ wartość zmiennej *liczba klientów w kolejce*

4. Określ czas przybycia następnego klienta – zdarzenie *Przybycie*

Zdarzenie: Zakończenie (Zak) obsługi i opuszczenie systemu

1. Wyznacz czas pobytu w systemie
2. Jeżeli kolejka nie jest pusta:
 - 2.1. weź pierwszego klienta z kolejki,
 - 2.2. oblicz czas pobytu tego klienta w kolejce,
 - 2.3. rozpocznij obsługę klienta (określ czas zakończenia – zdarzenie Zakończenie),
3. Jeżeli kolejka jest pusta:
 - 3.1. oznacz Stanowisko jako wolne (BRAK zdarzenia Zakończenie).

Dwie podstawowe zasady generowania zgłoszeń:

1. Gdy pojawia się Nowe Zgłoszenie, *WTEDY I TYLKO WTEDY* generujemy zdarzenie przybycia kolejnego zgłoszenia (*Przyb*).
2. Gdy rozpoczyna się obsługa zgłoszenia na stanowisku, *WTEDY I TYLKO WTEDY* generujemy zdarzenie Zakończenia Obsługi (*Zak*).

Objaśnienia do przykładu:

Kalendarz zdarzeń:

- * [Czas zdarzenia; Typ zdarzenia; Numer klienta],
- * Zdarzenia układane są *rosnąco* wg czasu,
- * Kolejno mające nastąpić zdarzenie umieszczone jest na górze kalendarza,
- * W momencie startu symulacji w kalendarzu umieszczone są dwa zdarzenia: przybycie pierwszego klienta [Przyb: 1] i koniec symulacji [Koniec].

Zmienne stanu:

- * Status Stanowiska $S(t) = 1$; gdy jest zajęte, 0; gdy wolne,
- * Liczba zgłoszeń w kolejce.

Dane liczbowe:

Losowo wygenerowane momenty przybycia klientów i czas obsługi klientów:

- * Odstępy czasu pomiędzy przybyciem kolejnych klientów: 4,9; 4,5,1,2; 1,3; 5,1... [min]
- * Czas obsługi (czas trwania rozmowy): 5,0; 7,0; 11,2; 6,8; ... [min]

Warunek zakończenia symulacji:

Symulacja trwa 15 minut

Statystyki:

Z = liczba klientów, którzy opuścili system,

N = liczba klientów, którzy opuścili kolejkę,

$\sum WQ$ = suma czasu oczekiwania w kolejce,

$\int Q$ = statystyka konieczna do wyznaczenia średniej długości kolejki. Jest to pole obszaru pod krzywą przedstawiającą kształtowanie się długości kolejki w miarę upływu czasu.

Tabela 4.3. Przebieg symulacji „Obsługa klientów w Centrali Telefonicznej” wg metody kolejnych zdarzeń

Czas	Pobrane zdarzenie, Nr klienta	Zawartość kalendarza	Czasy ustawienia w kolejce	Numery czekających klientów	Stanowisko Wolne/Zajęte [Nr klienta]	Statystyki			
						Z	N	$\sum WQ$	$\int Q$
0,0	START, –	[0,0; Przyb; 1] [15,0; Koniec]			Wolne	0	0	0	0
0,0	PRZYB, 1	[4,9; Przyb; 2] [5,0; Zak; 1] [15,0; Koniec]			Zajęte [1]	0	1	0	0
4,9	PRZYB, 2	[5,0; Zak; 1] [9,4; Przyb; 3] [15,0; Koniec]	(4,9)	(2)	Zajęte [1]	0	1	0	0
5,0	ZAK, 1	[9,4; Przyb; 3] [12,0; Zak; 2] [15,0; Koniec]			Zajęte [2]	1	2	0,1	0,1
9,4	PRZYB, 3	[10,6; Przyb; 4] [12,0; Zak; 2] [15,0; Koniec]	(9,4)	(3)	Zajęte [2]	1	2	0,1	0,1
10,6	PRZYB, 4	[11,9; Przyb; 5] [12,0; Zak; 2] [15,0; Koniec]	(9,4; 10,6)	(3; 4)	Zajęte [2]	1	2	0,1	1,3
11,9	PRZYB, 5	[12,0; Zak; 2] [15,0; Koniec] [17,0; Przyb; 6]	(9,4; 10,6; 11,9)	(3; 4; 5)	Zajęte [2]	1	2	0,1	3,9
12,0	ZAK, 2	[15,0; Koniec] [17,0; Przyb; 6] [23,2; Zak; 3]	(10,6; 11,9)	(4; 5)	Zajęte [3]	2	3	2,7	4,2
15,0	KONIEC, –								

Symulacja rozpoczyna się, kiedy w systemie pojawia się pierwsze zgłoszenie. Standardowo przyjmuje się, że moment ten opisany jest czasem 0,0. Wszystkie kolejne wskazania zegara symulacji naliczane są względem godziny 0,0. Zanim to pierwsze zgłoszenie pojawiło się w systemie, stanowisko obsługi było wolne i kolejka oczekujących zgłoszeń była pusta.

Godzina 0,0: Przybyło pierwsze zgłoszenie. Sprawdzamy, czy telefonistka jest „wolna” (TAK), a zatem wprowadzamy zgłoszenie bezpośrednio na stanowisko (Telefonistka odebrała telefon). Ponieważ rozpoczęła się obsługa zgłoszenia, **planowane** jest zdarzenie zakończenia obsługi, które pojawi się, gdy *zegar* wskaże 5 minut (tak wynika z naszych danych: czas obsługi pierwszego zgłoszenia to 5 minut). Do *kalendarza* wstawiamy zatem zdarzenie [5,0; ZAK; 1]. Zgodnie z pierwszą zasadą generowania zdarzeń, musimy jeszcze zaplanować czas nadejścia drugiego zgłoszenia. Zdarzenie to pojawi się wtedy, gdy *zegar* wskaże 4,9 min (zgodnie z danymi, odstęp czasu między nadejściem pierwszego i drugiego zgłoszenia to właśnie 4,9 min), a zatem

w *kalendarzu* pojawi się zdarzenie [4,9; PRZYB; 2]. *Kalendarz* wygląda tak, jak w wierszu drugim tabeli. Zauważmy, że zgodnie z *kalendarem*, kolejne zdarzenie pojawi się dopiero, gdy *zegar* wskaże 4,9 min. Możemy zatem w procesie symulacji **pomiąć** czas od godz. 0,0 do 4,9 min i od razu ustawić *zegar* na czas 4,9 min (wiersz trzeciej tabeli).

Godzina 4,9: O tej godzinie mamy do czynienia ze zdarzeniem, które było na górze *kalendarza*, czyli przybyciem do systemu zgłoszenia nr 2 [4,9; PRZYB; 2]. Tym razem nie możemy natychmiast wprowadzić zgłoszenia na stanowisko, bo Telefonistka ciągle jeszcze prowadzi rozmowę z pierwszym klientem. Zgłoszenie nr 2 ustawiamy zatem w kolejce, zapisując godzinę wstawienia do kolejki oraz numer zgłoszenia oczekującego. Nie możemy zaplanować zdarzenia Koniec Obsługi Zgłoszenia nr 2 (bo obsługa jeszcze się nie zaczęła), ale musimy **zaplanować** przybycie kolejnego zgłoszenia (Zasada nr 1). W *kalendarzu* pojawia się zatem zapis: [9,4; PRZYB; 3]. Według naszych danych między nadejściem drugiego i trzeciego zgłoszenia upływa 4,5 min. Ponieważ *zegar* wskazuje nam czas = 4,9 min, zatem zgłoszenie nr 3 przybędzie w momencie $4,9 + 4,5 = 9,4$ minuty. Zgodnie z zapisem *kalendarza*, kolejne zdarzenie pojawi się o godz. 5,0 min (por. zapis *kalendarza* w trzecim wierszu tabeli).

Godzina 5,0: Zdarzenie, które musimy rozpatrzyć to Zakończenie Obsługi zgłoszenia nr 1. Telefonistka kończy rozmowę z pierwszym klientem (zgłoszenie opuszcza system) i może zająć się zgłoszeniem (nr 2) oczekującym w kolejce. Gdyby kolejka była pusta, Stanowisko Obsługi oznaczone byłoby jako Wolne i **żadne nowe zdarzenie** nie byłoby planowane. Zgłoszenie numer 2 opuszcza zatem kolejkę (por. wiersz nr 4 tabeli) i rozpoczyna się obsługa tego zgłoszenia przez Telefonistkę. Stanowisko Obsługi jest zatem ciągle zajęte (teraz obsługą zgłoszenia drugiego), a my możemy **zaplanować** zakończenie obsługi zgłoszenia drugiego (druga zasada planowania zdarzeń). Nie planujemy nadejścia nowego zgłoszenia, bo byłoby to niezgodne z pierwszą zasadą planowania zdarzeń.

Godzina 9,4: W systemie pojawia się nowe zgłoszenie – numer 3. Sytuacja wygląda podobnie jak w przypadku nadejścia zgłoszenia drugiego. Stanowisko obsługi jest zajęte (obsługa klienta nr 2), a zatem zgłoszenie nr 3 ustawia się w kolejce, gdzie czeka na zwolnienie stanowiska obsługi. **Planujemy** przybycie nowego zgłoszenia (zasada nr 1) i w *kalendarzu* pojawia się zapis [10,6; PRZYB; 4].

Godzina 10,6: Pojawia się nowe zgłoszenie – numer 4. Stanowisko obsługi jest nadal zajęte obsługą klienta nr 2, a w kolejce czeka już zgłoszenie numer 3. Nowe zgłoszenie dołącza do kolejki i ustawia się za zgłoszeniem numer 3. Ponieważ do systemu przybyło nowe zgłoszenie, **planujemy** przybycie kolejnego zgłoszenia (numer 5) i w *kalendarzu* pojawia się zapis [11,9; PRZYB; 5].

Godzina 11,9: W systemie pojawia się nowe zgłoszenie – numer 5. Stanowisko obsługi jest nadal zajęte obsługą klienta nr 2, a w kolejce czekają zgłoszenia numer 3 i 4. Nowe zgłoszenie dołącza do kolejki i ustawia się za zgłoszeniem numer 4. Ponieważ do systemu przybyło nowe zgłoszenie, **planujemy** przybycie kolejnego zgłoszenia (numer 6) i w *alendarzu* pojawia się zapis [17,0; PRZYB; 6].

Godzina 12,0: Kończy się obsługa zgłoszenia numer 2, które opuszcza system, natomiast Telefonistka odbiera zgłoszenie czekające najdłużej w kolejce (zgłoszenie numer 3). Kolejka liczy teraz dwa zgłoszenia (numer 4 i 5). Telefonistka ciągle jest zajęta, a my **planujemy** koniec obsługi zgłoszenia nr 3 (które właśnie dodzwoniło się do Telefonistki).

Godzina 15,0: Proces symulacji kończy się, bowiem pobieramy z *alendarza* zdarzenie [15,0; KONIEC]. Oczywiście, gdyby inaczej ustawione były warunki eksperymentu (czas symulacji planowalibyśmy np. na 1 godzinę), symulacja toczyłaby się dalej według przedstawionego schematu.

Przykładowe wyniki:

- *Średni czas pobytu klienta w kolejce:*

Całkowity czas pobytu w kolejce/liczba klientów, którzy opuścili kolejkę = $(0 + 0,1 + 2,6) = 2,7$ min/3 klientów = 0,9 min/klienta.

Zauważmy, że pierwsze zgłoszenie po wejściu do systemu nie musiało czekać w kolejce, ponieważ stanowisko obsługi było wolne. Musimy jednak uwzględnić to zgłoszenie w statystykach, przypisując mu zerowy czas oczekiwania w kolejce.

- *Średnia liczba klientów w kolejce:*

Całkowity czas pobytu w kolejce/końcowe wskazanie zegara symulacji = $(0 \cdot 4,9 + 1 \cdot 0,1 + 0 \cdot 4,4 + 1 \cdot 1,2 + 2 \cdot 1,3 + 3 \cdot 0,1 + 2 \cdot 3) / 15 = 10,2$ klient-min / 15 min = 0,68 klienta. Sposób obliczania tej statystyki przedstawiony jest na rysunku 4.8.

- *Wykorzystanie Stanowiska:*

Czas pracy/końcowe wskazanie zegara symulacji = $15 / 15 = 1$ (bezwymiarowe). W przykładzie stanowisko pracowało bez chwili przerwy, a zatem czas pracy wynosi 15 minut.

- *Średni czas pobytu w systemie:*

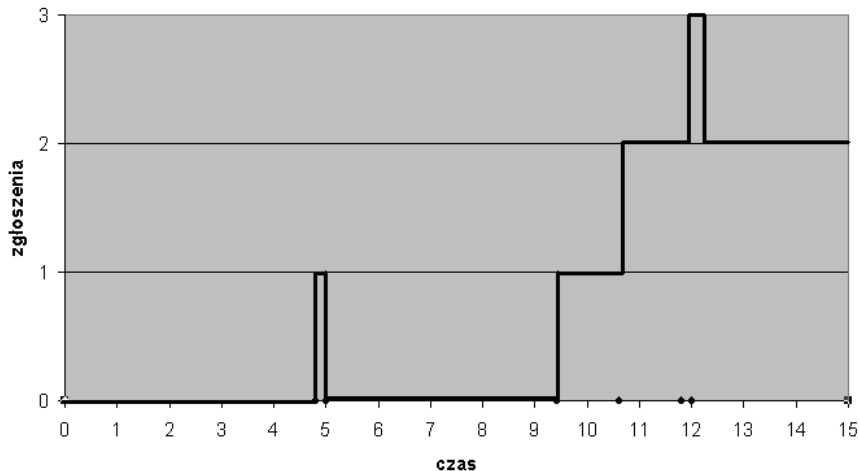
Całkowity czas pobytu w systemie/liczba klientów, którzy opuścili system = $(5,0 + 7,1) / 2 = 6,05$ min/klienta.

- *Liczba obsłużonych klientów:* 2 klientów

- *Maksymalny czas oczekiwania w kolejce:* 2,6 minuty

Pierwszy klient czekał 0 min, drugi klient 0,1 min, natomiast trzeci 2,6 min.

- *Maksymalna długość kolejki:* 3 klientów.



Rys. 4.8. Liczba zgłoszeń czekających w kolejce, w kolejnych minutach symulacji

Źródło: opracowanie własne

Poprawna interpretacja statystyk końcowych symulacji wymaga wiedzy na temat sposobu ich obliczania. Przykładowo, przy wyznaczaniu średniego czasu pobytu w systemie brane są pod uwagę wszystkie zgłoszenia, które opuściły system, natomiast obliczając średni czas pobytu w kolejce, uwzględnia się wyłącznie klientów, którzy opuścili kolejkę (niekoniecznie system).

4.5.6. REALIZACJA PROCESU SYMULACJI WG METODY PRZEGLĄDU DZIAŁAŃ – PRZYKŁAD NR 3

Posłużymy się przykładem 2 (podrozdz. 4.5.5).

Rozpatrywać będziemy dwa zdarzenia bezwarunkowe: *Przybycie zgłoszenia* (*Przyb*) i *Zakończenie obsługi* (*Zak*) oraz jedno zdarzenie warunkowe *Rozpoczęcie obsługi* (*Rozp*).

Zmiennymi stanu będą Nq liczba zgłoszeń oczekujących na obsługę i S stan aktywności stanowiska obsługi (wolne/zajęte).

Działanie uruchamiane zajściem zdarzenia warunkowego to *działanie obsługi klienta*.

Warunek, który musi być spełniony, aby mogło zaistnieć zdarzenie warunkowe *Rozp* to:

$Nq > 0$ AND $S == Wolne$ (w kolejce czeka co najmniej jeden klient i stanowisko obsługi jest wolne).

Zauważmy, że w fazie C (por. rys. 4.6), wykonanie zdarzenia *Rozp* może uwolnić zasoby (stanowisko obsługi), dlatego istotne jest powtarzanie fazy C aż do momentu, w którym żadne kolejne zdarzenie *Rozp* nie może zostać wykonane.

Uruchomienie symulacji (por. tab. 4.4) powoduje ustawienie warunków początkowych: $Nq:=0$, $S:=Wolne$. W kalendarzu umieszczane jest pierwsze zdarzenie [0,0; Przyb; 1] informujące o tym, że pierwszy klient przybędzie o godzinie 0,0 (zdarzenie bezwarunkowe).

Godzina 0,0

Faza A. Pobierane jest zdarzenie bezwarunkowe z kalendarza (usuwany jest zapis), zegar symulacji ustawiany jest na czas zdarzenia, zmienne stanu nie zmieniają wartości.

Faza B. Wykonywane jest zdarzenie pobrane z kalendarza w fazie A: zmienna stanu Nq zmienia wartość na 1, planowane jest kolejne zdarzenie *Przybycia* i odpowiedni zapis umieszczany jest w kalendarzu (przybycie klienta nr 2).

Faza C. Sprawdzany jest warunek $Nq > 0$ AND $S == Wolne$. Ponieważ warunek jest spełniony, wykonywane jest zdarzenie warunkowe, czyli uruchamiane jest *działanie obsługi klienta*: z kolejki usuwany jest klient, stanowisko obsługi staje się *zajęte*, planowane jest zdarzenie *Zakończenia* obsługi (klienta nr 1). W dalszym kroku ponawiana jest próba wykonania innych działań warunkowych, ponieważ jednak warunek zaistnienia zdarzenia warunkowego nie jest spełniony ($Nq == 0$ AND $S == Zajęte$), faza C jest zamykana.

Godzina 4,9

Faza A. Pobierane jest kolejne zdarzenie bezwarunkowe z kalendarza, czyli *Przybycie* klienta nr 2. Zegar symulacji wskazuje 4,9 min, zmienne stanu nie zmieniają wartości.

Faza B. Wykonywane jest zdarzenie bezwarunkowe *Przyb 2*: w kolejce umieszczany jest jeden klient ($Nq==1$) i planowane jest kolejne zdarzenie *Przybycia* (klienta nr 3).

Faza C. Nie jest uruchamiana, ponieważ nie jest spełniony warunek *działania* (stanowisko jest zajęte).

Godzina 5,0

Faza A. Z kalendarza pobierane jest zdarzenie *Zak 1* (koniec obsługi klienta nr 1). Zegar symulacji wskazuje 5,0 min.

Faza B. Wykonywane jest zdarzenie *Zak 1*: stanowisko obsługi zmienia status na *wolne*.

Faza C. Warunek *działania* jest spełniony (stanowisko jest wolne i kolejka nie jest pusta), wykonywane jest więc zdarzenie warunkowe *Rozp*: z kolejki usuwany jest klient, status stanowiska zmienia się na *zajęte*, rozpoczyna się obsługa klienta nr 2 i planowane jest zdarzenie bezwarunkowe *Zak 2*. Kolejny cykl fazy C się nie rozpoczyna, ponieważ warunek *działania* nie jest spełniony.

Symulacja toczy się dalej. Zauważmy, że w chwilach 9,4, 10,6 i 11,9 faza C nie jest uruchamiana, ponieważ nie jest spełniony warunek *działania*.

Zawartość *kalendacza zdarzeń* jest taka sama bez względu na wybór metody symulacji dyskretnej: kolejnych zdarzeń czy wyboru działań (por. podrozdz. 4.5.5 i 4.5.6). Identyczne będą też wyniki symulacji (pod warunkiem oczywiście, że będziemy korzystali dokładnie z tych samych liczb losowych). Różnica polega na przebiegu procesu i realizacji procedur logicznych.

Tabela 4.4. Przebieg symulacji „Obsługa klientów w Centrali Telefonicznej” według metody przeglądu działań

Aktualna faza	Pobrane zdarzenie	A Czas	B Zaplanowane zdarzenia bezwarunkowe	C Wykonywane zdarzenia warunkowe	Nq	S [Numer obsługiwanego klienta]
Uruchomienie		0,0	[0,0; Przyb; 1]	–	0	Wolne
A	Przyb, 1	0,0		–	0	Wolne
B	Przyb, 1	0,0	[4,9; Przyb; 2]	–	1	Wolne
C	Przyb, 1	0,0	[4,9; Przyb; 2] [5,0; Zak; 1]	[Rozp, 1]	0	Zajęte [1]
A	Przyb, 2	4,9	[5,0; Zak; 1]	–	0	Zajęte [1]
B	Przyb, 2	4,9	[5,0; Zak; 1] [9,4; Przyb; 3]	–	1	Zajęte [1]
A	Zak, 1	5,0	[9,4; Przyb; 3]	–	1	Zajęte [1]
B	Zak, 1	5,0	[9,4; Przyb; 3]	–	1	Wolne
C	Zak, 1	5,0	[9,4; Przyb; 3] [12,0; Zak; 2]	[Rozp, 2]	0	Zajęte [2]
A	Przyb, 3	9,4	[12,0; Zak; 2]	–	0	Zajęte [2]
B	Przyb, 3	9,4	[10,6; Przyb; 4] [12,0; Zak; 2]	–	1	Zajęte [2]
A	Przyb, 4	10,6	[12,0; Zak; 2]	–	1	Zajęte [2]
B	Przyb, 4	10,6	[11,9; Przyb; 5] [12,0; Zak; 2]	–	2	Zajęte [2]
A	Przyb, 5	11,9	[12,0; Zak; 2]	–	2	Zajęte [2]
B	Przyb, 5	11,9	[12,0; Zak; 2] [17,0; Przyb; 6]	–	3	Zajęte [2]
A	Zak, 2	12,0	[17,0; Przyb; 6]	–	3	Zajęte [2]
B	Zak, 2	12,0	[17,0; Przyb; 6]	–	3	Wolne
C	Zak, 2	12,0	[17,0; Przyb; 6] [23,2; Zak; 3]	[Rozp, 3]	2	Zajęte [3]

4.6. ANALIZA WYNIKÓW W DISKRETNYM MODELU SYMULACYJNYM

Za pomocą symulacji stochastycznej nie uzyskamy dokładnej odpowiedzi na postawione pytania. Będziemy mogli natomiast oszacować wartości średnie, prawdopodobieństwo zajścia lub niewystąpienia danego zdarzenia, długość przedziałów ufności. *Losowe wejście* oznacza również *losowe wyjście*, a to z kolei wymusza zaplanowanie i przeprowadzenie eksperymentów symulacyjnych w odpowiedni sposób.

W omawianym powyżej przykładzie *losowe wejście* występuje pod postacią losowo wygenerowanych momentów przybycia i czasu obsługi klientów. Zaprezentowane i wykorzystane w przykładzie wartości pochodzą z pewnych rozkładów losowych i oczywiście przy kolejnej próbie symulacji wygenerowane zostałyby inne (choćby pochodzące z tego samego rozkładu teoretycznego) liczby. Wyniki symulacji uzyskane w kolejnych powtórzeniach będą się zatem również różniły pomiędzy sobą (por. tab. 4.5).

Zazwyczaj wykonuje się wiele niezależnych powtórzeń, z których każde rozpoczyna się i kończy zgodnie z tymi samymi zasadami, przy zachowaniu tych samych warunków początkowych, ale dla różnych zestawów liczb losowych. Po każdym powtórzeniu zbierane są wyniki symulacji i informacja o typowym zachowaniu systemu jest przedstawiana w postaci średnich, odchyień oraz przedziałów ufności wyznaczonych na podstawie wyników zbiorczych ze wszystkich powtórzeń – por. podrozdz. 4.8.4.

Tabela 4.5. Wyniki symulacji „Centrala telefoniczna” uzyskane w pięciu powtórzeniach

Mierniki wyjściowe	Powtórzenie					Średnia	Odchylenie standardowe
	1	2	3	4	5		
Średni czas pobytu w kolejce	0,19	1,12	3,72	0,00	0,00	1,01	1,59
Średnia długość kolejki	0,27	0,30	0,99	0,00	0,00	0,31	0,41
Wykorzystanie stanowiska	0,85	0,93	1,00	0,32	0,37	0,69	0,32

4.7. NARZĘDZIA DO PROWADZENIA SYMULACJI DISKRETNEJ

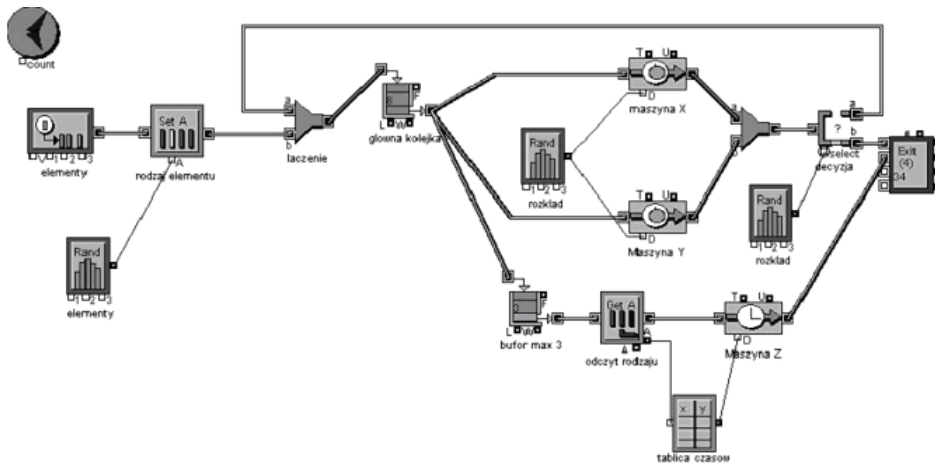
Począwszy od lat 60. XX wieku symulacyjne modele dyskretne opracowywane były za pomocą *symulacyjnych języków programowania*, takich jak: GPSS, SIMULA, SIMSCRIPT, SIMAN, SIMFACTORY, XCELL i inne. Wykorzystywano również ogólne języki programowania FORTAN, C, PASCAL. Najpierw powstawał model matematyczny, potem go oprogramowywano, kompilowano i uruchamiano symulację.

Wraz z gwałtownym rozwojem komputerów rozwijały się również komputerowe narzędzia ukierunkowane na wspomaganie procesów decyzyjnych. Symulacja nie była tu oczywiście wyjątkiem. *Interaktywna symulacja wizualna* (VIS – *Visual interactive simulation*) wykorzystuje interaktywne podejście oraz narzędzia projektowania graficznego we wszystkich etapach budowy i oprogramowania modelu symulacyjnego, począwszy od budowy modelu do zaplanowania i wykonania eksperymentów symulacyjnych.

Pierwszym komercyjnym pakietem VIS był SEE-WHY sprzedawany w Wielkiej Brytanii od 1979 roku. W dalszych latach opracowywano narzędzia dla określonych dziedzin, jak np. SIMFACTORY, XCELL czy OPTIK Process Line Simulator, które wykorzystują ikony graficzne charakterystyczne dla procesu produkcji. Czasami rozszerzano istniejące narzędzia do symulacji o elementy do wizualizacji. Tak było np. z GPSS/H, który rozszerzono o tzw. Proof Animation, czy z SIMSCRIPT II.5, któremu dodano SIMGRAPHICS.

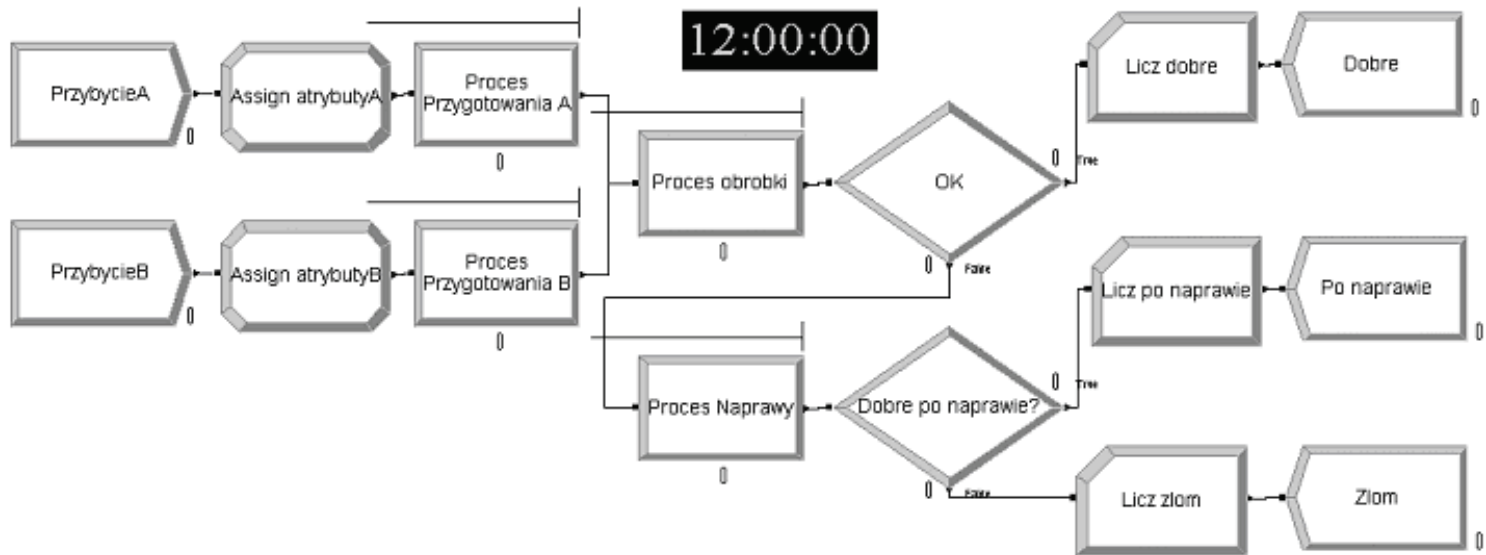
Powstawały również zupełnie nowe narzędzia, opracowane zgodnie z duchem VIS, jak np. ProModel dla systemów produkcyjnych, ServiceModel dla systemów obsługi czy MedModel do symulacji systemów opieki zdrowotnej.

Spośród obecnie dostępnych, najbardziej popularnych narzędzi VIS do symulacji dyskretnej wymienić należy pakiet ARENA (firmy Rockwell Automation) oraz Extend (firmy Imagine That).



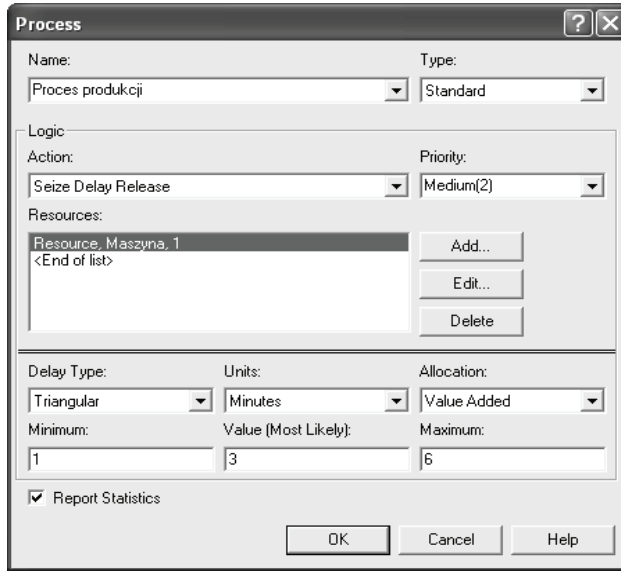
Rys. 4.9. Schemat modelu symulacyjnego procesu produkcyjnego opracowanego za pomocą pakietu Extend

Oba pakiety są przeznaczone do modelowania systemów dynamicznych i oferują interaktywny sposób budowy modelu, tworzenie animacji procesu symulacji, planowanie i wykonanie eksperymentów symulacyjnych, statystyczną analizę wyników oraz przeprowadzanie analizy danych wejściowych. Pakiety bazują na metodzie *interakcji procesów*. Użytkownik buduje model graficzny w postaci schematu złożonego z sieci połączonych ze sobą bloków (ikon), (por. rys. 4.9 i rys. 4.10), następnie

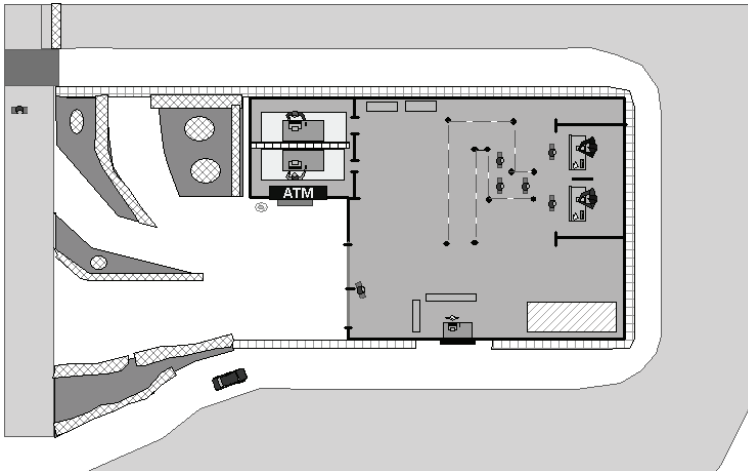


Rys. 4.10. Schemat modelu symulacyjnego procesu produkcyjnego opracowanego za pomocą pakietu Arena
 Źródło: opracowane własne

za pomocą okien dialogowych definiuje relacje, wprowadza warunki, ustala wartości zmiennych i parametrów (por. rys. 4.11) i – jeżeli uznaje to za konieczne – dodaje animację procesu symulacji (por. rys. 4.12). Sieć bloków może powstawać w sposób hierarchiczny – każdy blok może być zdefiniowany jako podmodel, który dalej również może składać się z pojedynczych bloków lub podmodeli. Porównanie implementacji modelu magazynowego w obu środowiskach graficznych można znaleźć w [Zabawa i Mielczarek 2007].



Rys. 4.11. Przykładowe okno dialogowe pakietu Arena

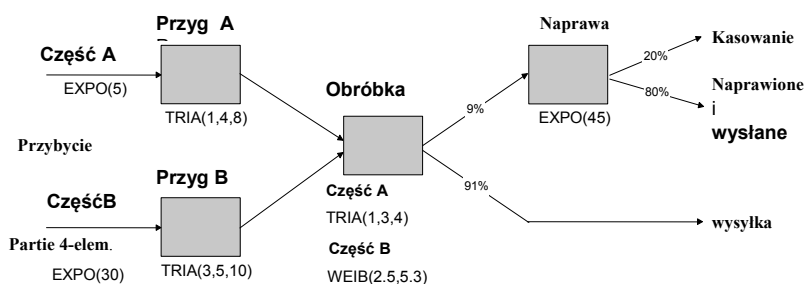


Rys. 4.12. Animacja procesu symulacji
Źródło: przykładowy model z biblioteki pakietu Arena

4.8. PRZYKŁADOWY MODEL DYSKRETNY OPRACOWANY ZA POMOCĄ PAKIETU ARENA

4.8.1. OPIS SYSTEMU

Rozważmy halę produkcyjną, na którą wprowadzane są dwa typy produktów metalowych stanowiące podstawy, na których instalowane będą części elektroniczne (por. rys. 4.13).



Rys. 4.13. Przepływ wyrobów w systemie produkcyjnym
Źródło [Kelton i in. 2004]

Produkt typu A (*Część A*) napływa z innej hali produkcyjnej w tempie, które można opisać za pomocą rozkładu wykładniczego o średniej 5 minut. Po przybyciu produkt jest transportowany na stanowisko przygotowawcze (*Przyg A*), gdzie poddaje się go wstępnej obróbce. Czas jej trwania opisany jest rozkładem trójkątnym o parametrach (1, 4, 8) minut. Po wstępnej obróbce realizowany jest transport produktu typu A na stanowisko *Obróbka*, gdzie realizowany będzie główny proces (umieszczanie części elektronicznych).

Produkt typu B (*Część B*) jest produkowany (podobnie jak produkt A) w innym budynku i napływa do hali w partiach czteroelementowych w tempie, które można opisać za pomocą rozkładu wykładniczego o średniej 30 minut. Po przybyciu partie są rozdzielane na pojedyncze wyroby i (już pojedynczo) produkt B kierowany jest na oddzielne stanowisko przygotowawcze (*Przyg B*). Czas trwania czynności na tym stanowisku może być również opisany rozkładem trójkątnym, jak w przypadku produktu A, ale o parametrach 3, 5, 10 minut. Po wstępnej obróbce produkt typu B transportowany jest na stanowisko *Obróbka* (to samo stanowisko, na które kierowane są produkty A), gdzie umieszczane będą na nim części elektroniczne.

Czas trwania obróbki na stanowisku *Obróbka* jest inny dla wyrobów A i B, wynosi odpowiednio 1, 3, 4 minuty w rozkładzie trójkątnym oraz (2,5; 5,3)* minut w rozkładzie Weibulla. Po zakończeniu obróbki wyroby są testowane i 91% wyrobów (wyroby, które pomyślnie przeszły kontrolę jakości) jest transportowanych na halę wysyłki.

* W pakiecie Arena liczby dziesiętne zapisywane są z kropkami, a nie z przecinkami, jak to jest przyjęte w języku polskim.

Pozostałe wyroby (z wadami) kierowane są na stanowisko naprawcze, gdzie są demontowane, czyszczone i ponownie składane. Ta operacja trwa około 45 minut w rozkładzie wykładniczym. 80% naprawianych wyrobów udaje się naprawić i są one kierowane do wysyłki, natomiast 8% jest złomowanych.

4.8.2. ZAŁOŻENIA DO MODELU

Należy przeprowadzić symulację, zebrać dane dotyczące procesu produkcji, wskazać wąskie gardło i określić czas produkcji z podziałem na wyroby dobre, naprawione i złomowane.

W dyskretnym modelu symulacyjnym musimy wskazać *zgłoszenia*. W naszym przykładzie wyróżnimy dwa typy zgłoszeń (*Część A* i *Część B*) napływające w dwóch niezależnych strumieniach do systemu i w różnym tempie (por. rys. 4.13).

Następnie musimy zdefiniować *Procesy*, w trakcie których będzie realizowana obsługa zgłoszeń. Wyróżnimy cztery procesy: *Proces Przygotowania A*, *Proces Przygotowania B*, *Proces obróbki* i *Proces Naprawy*.

Obsługa zgłoszeń będzie prowadzona na *stanowiskach obsługi*, przyporządkowanych do poszczególnych procesów. W modelu wyróżnimy cztery stanowiska obsługi: Stanowisko Przygotowawcze A (*Przyg A*), Stanowisko Przygotowawcze B (*Przyg B*), *Obróbka*, *Naprawa*.

Zgłoszenia będą przemieszczały się przez system, zajmując i zwalniając stanowiska obsługi, przy czym strumień *Część A* będzie kierowany przez stanowisko *Przyg A* do stanowiska *Obróbka*, a potem do stanowiska *Naprawa*, natomiast strumień *Część B* pokona podobną drogę, ale rozpoczynając od stanowiska *Przyg B*. W modelu będą mogły tworzyć się cztery kolejki: przed każdym stanowiskiem obsługi.

Istotnymi elementami modelu będą *dwa bloki decyzyjne*, w których podejmowana będzie decyzja o dalszym losie każdego zgłoszenia przemieszczanego przez blok. W pierwszym bloku decyzja będzie dotyczyła wszystkich zgłoszeń wprowadzonych do modelu, natomiast w drugim bloku – tych, które zostały skierowane do naprawy. Ponieważ znamy procentowy udział wyrobów dobrych i z wadami oraz wyrobów naprawionych i złomowanych, bloki decyzyjne będą bazowały na wyborze *prawda/falsz*, a nie na testowaniu warunku logicznego.

Zgłoszenia (wyroby) przemieszczane przez halę będą charakteryzowały się różnymi *atrybutami* (własnościami). Jednym z atrybutów będzie np. *czas pojawienia się na hali*. Znajomość tej wartości będzie nam potrzebna do wyznaczenia czasu trwania całego procesu produkcji.

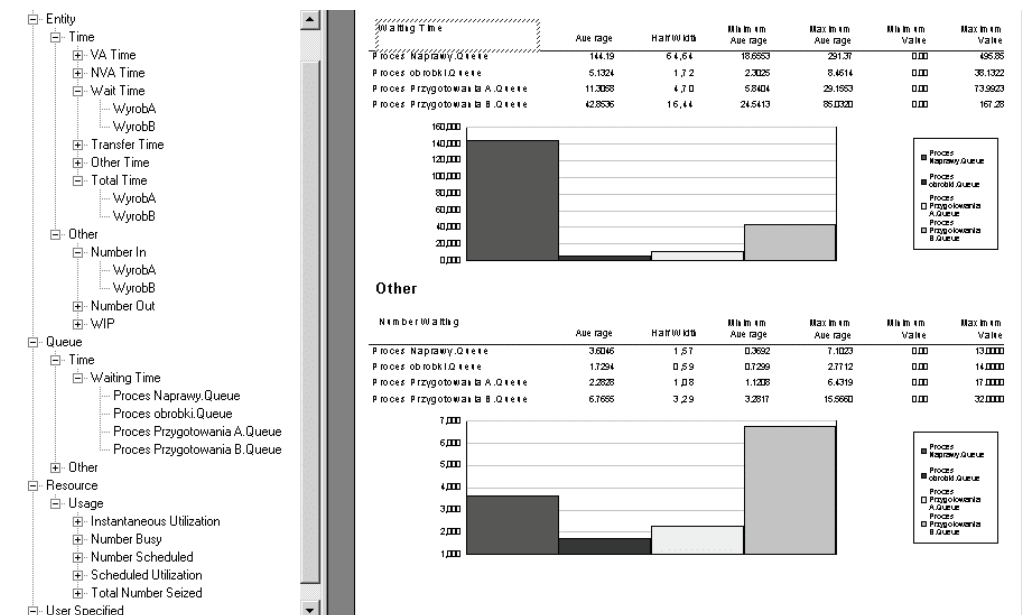
4.8.3. PRZEBIEG SYMULACJI

Proces symulacji rozpocznie się w chwili zero. Zaplanowane zostaną wtedy dwa zdarzenia (pojawienie się pierwszego zgłoszenia typu *Część A* oraz pierwszego zgłoszenia typu *Część B*), a następnie zegar symulacji zostanie ustawiony na pierwsze w kolej-

ności zdarzenie. Założmy, że jest nim pojawienie się wyrobu *Część A*. Wyrób typu *Część A* pojawia się zatem w modelu i rozpoczyna się przemieszczanie tego wyrobu zgodnie ze zdefiniowanym schematem aż do momentu, w którym wymuszone zostanie opóźnienie (np. obsługa na pierwszym stanowisku) lub pojawią się warunki blokujące (np. wyrób nie może być obsługiwany, ponieważ stanowisko jest zajęte). Zgłoszenie zostaje wtedy *zawieszona* i czeka na koniec opóźnienia lub na odblokowanie warunków, a z kalendarza pobierane jest kolejne zdarzenie (np. nadejście wyrobu typu *Część B*) i zegar symulacji przesuwany jest na tę chwilę. Podobnie jak w przypadku wyrobu A, wyrób B przemieszczany jest wzdłuż procesu tak daleko, jak to jest możliwe. Po napotkaniu warunków wstrzymujących (opóźnienie lub niedostępność stanowiska) jego przepływ jest wstrzymywany, a z kalendarza pobierane jest kolejne zdarzenie (np. przybycie kolejnego wyrobu A lub zakończenie opóźnienia dla pierwszego wyrobu A). Proces symulacji trwa tak długo, aż wypełni się zaplanowany czas powtórzenia.

4.8.4. WYNIKI SYMULACJI

W trakcie symulacji zbierane są szczegółowe informacje na temat zgłoszeń, kolejek, zasobów i innych zmiennych zdefiniowanych przez użytkownika. Po zakończeniu symulacji uśredniane są automatycznie wyniki z wszystkich powtórzeń, obliczane wartości średnie i przedziały ufności w różnorodnych przekrojach (por. rys. 4.14.). Możliwe jest ponadto zdefiniowanie własnych statystyk, niezbędnych do osiągnięcia celu symulacji.



Rys. 4.14. Przykładowe wyniki symulacji w pakiecie Arena

Źródło: raport uzyskany w pakiecie Arena

5. WERYFIKACJA I WALIDACJA MODELU

5.1. PODSTAWOWE DEFINICJE

Jednym z trudniejszych problemów, z którymi musi zmierzyć się twórca modelu symulacyjnego, jest sprawdzenie, czy zbudowany model w sposób właściwy opisuje modelowaną rzeczywistość. W teorii symulacji wprowadza się pojęcie *weryfikacji, walidacji i wiarygodności* modelu (por. [Fishman i Kowiat 1968]). *Weryfikacja modelu* to proces dowodzenia, że transformacja modelu z jednej postaci w drugą przebiegała z odpowiednią dokładnością. *Walidacja modelu* to proces dowodzenia, że model, w zakresie zastosowań, dla których został skonstruowany, wykazuje się satysfakcjonującą dokładnością, spójną z założonymi celami badawczymi. Jeżeli model symulacyjny i jego wyniki zostały zaakceptowane przez użytkowników, do których model jest adresowany, to model ten można uznać za *wiarygodny*.

W procesie *walidacji* modelu dąży się do sprawdzenia, czy model symulacyjny poprawnie opisuje system rzeczywisty. Proces *weryfikacji* pozwala sprawdzić, czy konceptualny model symulacyjny został poprawnie przetłumaczony na program komputerowy. Weryfikacja dotyczy zatem wewnętrznej spójności modelu, natomiast walidacja odnosi się do związków pomiędzy modelem a rzeczywistością. Model, który pomyślnie przeszedł weryfikację i walidację, i jest wykorzystywany w procesach decyzyjnych, uznawany jest za *wiarygodny*.

Ustalenie zasadności modelu jest niezbędnym składnikiem procesu symulacji.

Walidację i weryfikację (WiW) definiuje się ogólnie jako proces dowodzenia, że model symulacyjny poprawnie opisuje bądź przybliża system rzeczywisty w zakresie tych celów, dla których został opracowany. Przeznaczenie modelu symulacyjnego określa zatem aspekty, które powinny zostać poddane walidacji, jak również ustala stopień szczegółowości przeprowadzanej analizy walidacyjnej. Przykładowe pytania, które można zadać na poszczególnych etapach budowy modelu symulacyjnego, przedstawiono w tabeli 5.1.

W ostatnich czterdziestu latach opublikowano wiele tekstów na temat weryfikacji i walidacji modeli symulacyjnych. O zagadnieniach tych pisali między innymi tacy autorzy, jak: [Law i Kelton 1991], [Balci 1994], [Balci 1997], [Friedrich i Goldberg

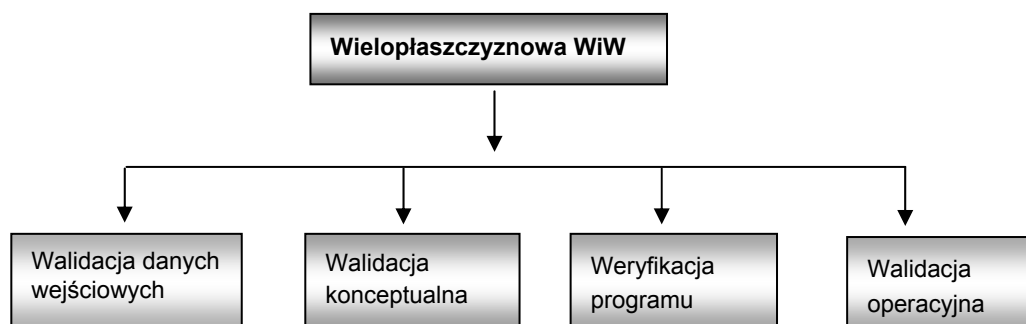
2000], [Sargent 2004], [Kleindorfer i Ganeshan 1993], [Kleijnen 1995] i wielu innych. Wszyscy oni podkreślają, że do chwili obecnej nie opracowano spójnej teorii walidacji ani nie przygotowano zestawu standardów metodologicznych. Zamieszczone poniżej rozważania należy zatem traktować jedynie jako pewien przyczynek do praktyki prowadzenia badań symulacyjnych.

Tabela 5.1. Przykładowe pytania zadawane w celu przeprowadzenia weryfikacji i walidacji modelu symulacyjnego

Model	Weryfikacja	Walidacja
Model konceptualny		Czy model zawiera wszystkie niezbędne elementy, zdarzenia, relacje? Czy za pomocą modelu będziemy w stanie odpowiedzieć na wszystkie pytania?
Model logiczny	Czy zdarzenia są właściwie przedstawione? Czy wszystkie formuły matematyczne i relacje są poprawne? Czy wszystkie funkcje zmiennych losowych są poprawnie określone?	Czy model zawiera wszystkie zdarzenia opisane w modelu konceptualnym? Czy model zawiera wszystkie relacje zawarte w modelu konceptualnym?
Model symulacyjny/model komputerowy	Czy wszystkie wielkości statystyczne są poprawnie naliczane?	Czy za pomocą modelu możemy odtworzyć sytuacje przeszłe? Czy model został zaakceptowany przez ekspertów?

5.2. WIELOPŁASZCZYZNOWA WERYFIKACJA I WALIDACJA

Ponieważ nie można za pomocą pojedynczego testu udowodnić poprawności modelu symulacyjnego, w literaturze przedmiotu wprowadza się pojęcie *wielopłaszczyznowej walidacji i weryfikacji* [Rao i in. 1998]. Proponuje się w niej przeprowadzanie procesu WiW na kilku płaszczyznach, z wykorzystaniem różnych technik i testów walidacyjnych. Ideę wielopłaszczyznowej walidacji i weryfikacji można zilustrować jak na rys. 5.1:



Rys. 5.1. Schemat wielopłaszczyznowej walidacji i weryfikacji modeli symulacyjnych

Walidacja danych wejściowych

Dane wejściowe są niezbędne do zrealizowania trzech podstawowych celów: budowy modelu konceptualnego, przeprowadzenia procesu weryfikacji i walidacji oraz wykonania eksperymentów na zweryfikowanym modelu. Na etapie konceptualizacji dane wykorzystuje się do sformułowania poprawnych relacji matematycznych i logicznych, jak również do opracowania teorii leżących u podstaw modelu i do sprawdzania przyjętych założeń. W procesie weryfikacji i walidacji dane są natomiast potrzebne do porównania zachowania systemu rzeczywistego i modelu. Proces zbierania danych wejściowych jest trudny, czasochłonny i kosztowny. Polecanym sposobem na uzyskanie wiarygodnych danych wejściowych jest zgromadzenie ich za pomocą właściwych metod statystycznych, na przykład za pomocą metody reprezentacyjnej.

Walidacja konceptualna

Walidacja konceptualna oznacza identyfikację i ewaluację założeń leżących u podstaw modelu oraz jakościową ocenę jego struktury i logiki. Założenia dotyczące liniowości, niezależności, stacjonarności czy poprawności dopasowania rozkładów teoretycznych testowane są za pomocą odpowiednich metod statystycznych. Ocenę poprawności struktury i logiki modelu dokonuje się natomiast poprzez tzw. *walidację zewnętrzną* (ang. *face validity*), czyli np. uzyskiwanie opinii osób niezwiązanych z projektem, ale posiadających znaczną wiedzę o funkcjonowaniu systemu rzeczywistego i/lub poprzez *technikę śledzenia zdarzeń*, której ideą jest „śledzenie” wybranych obiektów dynamicznych w trakcie ich wędrówki przez model i analizowanie w ten sposób logiki modelu i precyzji w opisie zdarzeń.

Weryfikacja programu

Weryfikacja programu ma na celu upewnienie się, że kod został opracowany w sposób właściwy oraz że model konceptualny został poprawnie przetłumaczony na język programowania. Zapewnieniem wysokiej jakości programu komputerowego zajmuje się inżynieria oprogramowania, która szczegółowo prezentuje wiele metod, technik i narzędzi służących do budowy odpowiedniego poziomu zaufania do produktu informatycznego. Najczęściej stosowane techniki weryfikacyjne programu komputerowego to m.in.: *inspekcje*, czyli techniki analizy wzrokowej produktów poszczególnych faz cyklu, *testowanie funkcjonalne*, czyli techniki eksperymentowania z kodem programu, traktowanym jak „czarna skrzynka” w celu sprawdzenia czy dany moduł ma przypisane mu cechy, *testowanie strukturalne*, czyli techniki eksperymentowania z kodem traktowanym jak „przezroczysta skrzynka” w celu sprawdzenia czy dany moduł wykonuje zaplanowane działania, *testowanie regresywne*, czyli powtórzenie uprzednio przeprowadzonych testów po dokonaniu w programie zmian, *pomiary osiągnięć*, czyli techniki eksperymentowania w celu pomiaru konkretnych parametrów fizycznych związanych z wykonywaniem programu i wiele innych. Ponieważ jednak nawet najbardziej starannie wykonane testy nie gwarantują wykrycia wszystkich błędów w programie, zaleca się zastosowanie kilku uzupełniających się strategii.

Walidacja operacyjna

Walidacja operacyjna odnosi się najczęściej do możliwości dokonania przez model predykcji zachowania systemu rzeczywistego. Najbardziej definitywnym testem walidacji operacyjnej jest udowodnienie, że dane wynikowe modelu są w wysokim stopniu zbieżne z wynikami, których oczekiwalibyśmy w systemie rzeczywistym przy założeniu wystąpienia identycznych parametrów wejściowych, dla dwóch różnych zestawów danych eksperymentalnych.

Trzy najczęściej stosowane metody walidacji operacyjnej to: (1) analiza porównawcza danych dostarczonych przez model i zaobserwowanych w systemie rzeczywistym, (2) badanie przedziałów ufności i (3) testowanie hipotez.

Metoda pierwsza polega na wyborze wskaźników operacyjnych, ważnych ze względu na cel budowy modelu, a następnie na przedstawieniu wartości tych wskaźników i/lub związków pomiędzy nimi w postaci numerycznej i/lub graficznej jednocześnie dla zbioru danych wynikowych pochodzących zarówno z modelu, jak i z systemu rzeczywistego.

Badanie przedziałów ufności umożliwia porównanie modelu z istniejącym systemem wtedy, gdy możliwe jest uzyskanie dużej liczby danych zarówno z modelu, jak i z systemu. Załóżmy, że zebraliśmy m niezależnych zbiorów danych z systemu i n niezależnych zbiorów danych z modelu. Dla każdego zebranego zbioru danych możemy określić średnią oraz wartość oczekiwaną analizowanego wskaźnika. Załóżmy, że $\mu_x = E(X_j)$ to wartość oczekiwana wskaźnika z j -tego zestawu danych systemu, natomiast $\mu_y = E(Y_j)$ to wartość oczekiwana z j -tego zestawu danych modelu. Chcemy porównać model z systemem budując przedział ufności dla $\zeta = \mu_x - \mu_y$. Budowanie takiego przedziału ufności wydaje się być lepsze niż testowanie hipotezy $H_0: \mu_x = \mu_y$, ponieważ:

- model jest tylko przybliżeniem rzeczywistości, a zatem hipoteza H_0 będzie fałszywa w większości przypadków,
- budowany przedział ufności dostarczy więcej informacji niż testowanie hipotezy H_0 . Jeżeli $\mu_x \neq \mu_y$, to przedział ufności dostarczy również informacji o wielkości zaobserwowanego odchylenia.

W przeprowadzaniu walidacji operacyjnej mogą być również pomocne takie techniki, jak: analiza wrażliwości, analiza wariancji, analiza regresji, analiza spektralna, analiza serii czasowych i inne.

Metoda trzecia stosowana jest do porównywania parametrów, rozkładów i serii czasowych dla danych z modelu i z systemu rzeczywistego, w celu zbadania, czy dany model generuje wyniki z wystarczającą dokładnością.

5.3. TRÓJSTOPNIOWA ANALIZA ZASADNOŚCI NAYLORA

Jedną z ważniejszych prac dotyczących problemu walidacji modeli symulacyjnych jest praca Naylora [1975], przedstawiająca koncepcję trójstopniowej oceny zasadności modeli symulacyjnych. Autor wskazuje na wagę problemu rozwinięcia możliwie uniwersalnych kryteriów akceptowania modelu symulacyjnego i proponuje procedurę zawierającą metodologię stanowiska racjonalistycznego, empiryzmu oraz ekonomii pozytywistycznej argumentując, że „każde z wymienionych powyżej stanowisk metodologicznych stanowi konieczną procedurę oceny zasadności [...], ale też, że żadne z nich nie jest procedurą wystarczającą”.

W *stanowisku racjonalistycznym* wychodzi się z założenia, że model lub teoria są zbiorem logicznych wniosków wysnutych z szeregu syntetycznych sądów o charakterze niepodważalnych prawd „niemogących być przedmiotem empirycznej weryfikacji bądź odwołania się do obiektywnego doświadczenia” [Emanuel Kant]. Tak więc cały problem zasadności redukuje się do poszukiwania zbioru założeń leżących u podstaw zachowania się interesującego nas systemu. Niestety, każda próba formalnego i szczegółowego odcyfrowania wszystkich podstawowych założeń, na których opiera się konkretny system, ujawnia wkrótce istnienie ograniczeń dotyczących ich oczywistości.

Empiryzm jest krańcowym przeciwstawieniem racjonalizmu. Za idealną postać wiedzy empirycy uważają nauki empiryczne, nie zaś matematykę. Empiryzm wzbrania się uznać te postulaty i założenia, które nie mogą być zweryfikowane w sposób od siebie niezależny.

Ekonomia pozytywistyczna twierdzi, że zasadność modelu nie zależy od zasadności założeń, na których ten model jest oparty, lecz przeciwnie, od możliwości dokonania przez model predykcji zachowania się zmiennych zawartych w modelu.

Trzy wymienione stanowiska zostały przez Naylora [1975] wykorzystane do zdefiniowania trzech stopni analizy zasadności modeli.

Pierwszy stopień oceny zasadności modeli polega na sformułowaniu zbioru postulatów dotyczących hipotez opisujących zachowanie się systemu (ang. *face validity*), który to zbiór powinien zostać zaakceptowany przez osoby związane z badanym systemem (eksperci, kierownicy, pracownicy) i nie powinien pozostawać w sprzeczności z obowiązującą teorią i wynikami wcześniej przeprowadzanych badań.

Drugi stopień polega na jakościowym sprawdzaniu założeń za pomocą dostępnych testów statystycznych.

Trzeci stopień dotyczy testowania możliwości dokonania przez model predykcji zachowania badanego systemu. W badaniu tym dąży się do sprawdzenia, czy wyniki procesu symulacji są porównywalne z danymi, których można spodziewać się, poddając obserwacji system rzeczywisty. Wykorzystuje się w tym celu dane historyczne (tzw. walidacja historyczna) pochodzące z systemu rzeczywistego oraz dane uzyskane w wyniku eksperymentów symulacyjnych. Można również przeprowadzić symulację

działania systemu w pewnej niedalekiej przyszłości i po upływie tego okresu porównać dane symulowane ze stanem faktycznym.

Podejście zaproponowane przez Naylora było rozwijane później przez innych autorów i obecnie dostępnych jest kilkadziesiąt technik, z których każda przeprowadza cząstkową walidację lub weryfikację modelu symulacyjnego. Niestety, nie jest dostępny algorytm ani procedura, które umożliwiłyby dokonanie wyboru właściwych technik, co sprawia, że – tak jak w przypadku pozostałych etapów procesu modelowania symulacyjnego – również przebieg oceny zasadności modelu zależy w dużej mierze od kompetencji i doświadczenia osób prowadzących badanie.

5.4. ZASADY PROCESU WALIDACJI I WERYFIKACJI MODELU

W literaturze przedmiotu (por. [Balci 1987]) wyróżnia się różne strategie prowadzenia procesu weryfikacji i walidacji modelu. Najczęściej stosowane podejście to prowadzenie procesu WiW przez twórców modelu symulacyjnego. Proces WiW jest wtedy włączany w proces budowy modelu symulacyjnego i polega na stosowaniu różnorodnych testów i metod oceny, które potwierdzają poprawność modelu. Inny sposób, to tak zwana *niezależna weryfikacja i walidacja* (NWiW) prowadzona przez osobę (osoby) niezwiązane ani z grupą twórców modelu, ani z grupą bezpośrednich odbiorców modelu. Jeszcze inne ze stosowanych podejść polega na wykorzystaniu modelu punktującego, który przyznaje różną liczbę punktów na poszczególnych etapach procesu WiW. Zsumowana liczba punktów jest podstawą do zaakceptowania lub odrzucenia modelu.

Balci [1994] wyróżnia kilkanaście zasad właściwie przeprowadzonego procesu walidacji i weryfikacji modelu:

Zasada 1. Walidacja i weryfikacja modelu symulacyjnego nie są procesami jednostkowymi i nie stanowią wyraźnie wyodrębnionego etapu badania symulacyjnego. Należy je traktować jako proces ciągły, zachodzący podczas całego cyklu modelowania.

Zasada 2. Wyników procesów symulacji nie należy odbierać w kategoriach *Prawda–Fałsz*. Nie można zatem dążyć do stwierdzenia, że model jest całkowicie fałszywy albo całkowicie poprawny. Ponieważ model jest tylko opisem rzeczywistości, całkowita poprawność budowanej repliki nie jest możliwa.

Zasada 3. Każdy model symulacyjny budowany jest w określonym celu i ocena jego wiarygodności odnosi się wyłącznie do sformułowanego celu. Sformułowany cel badania symulacyjnego określa, w jakim stopniu model powinien być wiarygodny – czasami wystarcza jedynie 90% dokładności, czasami jednak potrzeba aż 95%.

Zasada 4. Proces WiW powinien być prowadzony przez osobę niezależną. Twórca modelu symulacyjnego nie jest odpowiednią osobą do przeprowadzania procesu WiW, ponieważ najczęściej zależy mu na potwierdzeniu poprawności swoich badań i będzie (często podświadomie) dążył do potwierdzenia wiarygodności modelu.

Zasada 5. Proces WiW jest trudny i wymaga dużej kreatywności i umiejętności przeprowadzenia pogłębionej analizy. Niezbędna jest znajomość dziedziny, której dotyczy model, wcześniejsze doświadczenie w przeprowadzaniu WiW oraz doświadczenie w modelowaniu.

Zasada 6. Wiarygodność modelu może być potwierdzona jedynie dla warunków, których dotyczyły przeprowadzone testy. Dokładność transformacji „wejście–wyjście” zależy od charakterystyk parametrów wejściowych. Jeżeli transformacja jest poprawna dla jednego zestawu parametrów wejściowych nie oznacza to, że będzie również poprawna, gdy parametry wejściowe ulegną zmianie.

W modelu ruchu ulicznego można zbudować stacjonarny model symulacyjny zakładając, że w godzinach wieczornych częstotliwość napływu samochodów jest stała. Dla takich założeń przeprowadzony zostanie proces WiW, który wykaże poprawność modelu. Nie oznacza to jednak, że ten sam model będzie poprawny, gdy będziemy chcieli analizować cały dzień roboczy, od 6:00 do 19:00.

Zasada 7. Niemożliwe jest przeprowadzenie kompletnego testowania modelu. Kompletny proces testowania modelu wymaga przeprowadzenia procesu WiW we wszystkich możliwych warunkach. Kombinacja możliwych wartości zmiennych wejściowych może prowadzić do milionów zestawów parametrów wejściowych. Przetestowanie ich jest po prostu niemożliwe.

Zasada 8. Proces WiW powinien być zaplanowany i udokumentowany.

Zasada 9. Należy uchronić model przed błędami typu I, II i III.

Błąd typu I to błąd popełniony wtedy, gdy odrzucamy poprawne wyniki symulacji.

Błąd typu II to błąd popełniony wtedy, gdy akceptujemy fałszywe wyniki symulacji.

Błąd typu III to błąd polegający na rozwiązaniu niewłaściwego problemu lub na sformułowaniu problemu w taki sposób, że nie w pełni zawiera on cel badań.

Błąd typu I niepotrzebnie powiększa koszty badań. Skutki popełnienia błędu II mogą być poważne dla użytkowników modelu. Błąd III powoduje, że wyniki procesu symulacji są nieprzydatne.

Prawdopodobieństwo popełnienia błędu I nazywane jest *Ryzykiem Twórcy Modelu*. Prawdopodobieństwo popełnienia błędu II nazywane jest *Ryzykiem Użytkownika Modelu*.

Zasada 10. Błędy powinny być wykrywane tak wcześnie, jak to tylko jest możliwe.

Zasada 11. Należy rozpoznać możliwość wystąpienia tzw. wielokrotnej odpowiedzi modelu. W przypadku występowania kilku zmiennych wyjściowych nie można po kolei porównywać wyników symulacji z wartościami produkowanymi przez system rzeczywisty. Należy zastosować wielowariantową analizę statystyczną.

Zasada 12. Pomyślne testy wykonane na pod-modelach nie świadczą o poprawności całego modelu. Dopuszczalne błędy dla pod-modeli mogą się kumulować i w rezultacie cały model będzie obciążony znacznym błędem.

Zasada 13. Należy rozpoznać i rozwiązać problem podwójnej walidacji. Jeżeli możliwe jest pozyskanie danych zarówno wejściowych, jak i wyjściowych dla systemu, proces WiW może być przeprowadzony przez porównanie wyników modelu i systemu uzyskanych dla tych samych danych wejściowych. Problem leży jednak w określeniu poprawności danych wejściowych. Stąd nazwa – „podwójna walidacja”.

Zasada 14. Poprawność modelu symulacyjnego nie gwarantuje poprawności wyników symulacji. Proces WiW jest warunkiem koniecznym, ale niewystarczającym dla poprawności wyników symulacji. Jeżeli cel symulacji został niewłaściwie sformułowany i/lub system został niepoprawnie zdefiniowany, zbudujemy model, który będzie zgodny z (niewłaściwie) zdefiniowanym systemem, ale będzie produkował wyniki prowadzące do błędnych decyzji.

Zasada 15. Dokładność, z jaką sformułowany został cel, w dużym stopniu rzutuje na akceptowalność i poprawność wyników symulacji. Albert Einstein zauważył, że właściwie sformułowany problem jest ważniejszy od samego rozwiązania. Celem symulacji nie jest dostarczenie *jakiegoś* rozwiązania problemu, ale takiego rozwiązania, które zostanie zaakceptowane i wykorzystane przez decydentów.

Zasada 16. Złożoność procesu WiW zależy od złożoności modelu symulacyjnego. Weryfikacja modelu małego banku będzie łatwiejsza do przeprowadzenia niż weryfikacja modelu obrony marynarki wojennej.

Zasada 17. Do weryfikacji i walidacji modelu symulacyjnego należy użyć tych samych mierników działania, które będą wykorzystywane w procesie decyzyjnym.

5.5. TECHNIKI WALIDACJI I WERYFIKACJI

5.5.1. TAKSONOMIA TECHNIK WALIDACJI I WERYFIKACJI

Techniki procesu weryfikacji i walidacji można rozpatrywać w sześciu perspektywach (por. [Balci 1997]):

Nieformalne – są najczęściej wykorzystywane. Bazują na logicznym wnioskowaniu bez wykorzystania aparatu matematycznego (audyt, walidacja zewnętrzna, inspekcja, przegląd, test Turinga).

Statyczne – szacowanie dokładności na podstawie charakterystyk statycznego kodu źródłowego. Nie jest wymagane uruchomienie modelu, a jedynie przeprowadzenie symulacji mentalnej. Kompilator języka symulacyjnego jest sam w sobie narzędziem walidacji statycznej (kontrola spójności, analiza przepływu danych, analiza semantyczna, analiza strukturalna i analiza syntaktyczna).

Dynamiczne – wymagają uruchomienia modelu i przeznaczone są do oceny modelu na podstawie jego zachowania (test czarnej skrzynki, testowanie dół–górze, analiza wrażliwości, wizualizacja, testy statystyczne i inne).

Symboliczne – podobnie jak *dynamiczne* polegają na ocenie zachowania modelu w czasie jego wykonania. Dane wejściowe wprowadzane są w postaci symbolicznej natomiast wyniki w postaci wyrażań.

Z ograniczeniami – bada się poprawność modelu przy różnych ograniczeniach.

Formalne – oparte są na formalnych matematycznych dowodach poprawności modelu.

5.5.2. WYBRANE TECHNIKI WALIDACJI I WERYFIKACJI

Animacja – na ekranie wyświetlany jest przebieg procesu symulacji (np. graficzne przedstawienie przebiegu procesów w symulacji dyskretnej: przemieszczanie się zgłoszeń w systemie).

Porównanie z innymi modelami – różne wyniki procesu symulacji są porównywane z wynikami uzyskanymi z innych modeli (już zweryfikowanych). „Inne” modele mogą być modelami analitycznymi lub (również) symulacyjnymi. Czasami konieczne jest przeprowadzenie pewnych modyfikacji w weryfikowanym modelu, aby umożliwić takie porównanie.

Testy degeneracyjne – przeprowadza się je przez usunięcie z modelu pewnych jego fragmentów (ograniczenie modelu) lub przez odpowiednią selekcję danych wejściowych i wartości parametrów modelu. (Na przykład: czy długość kolejki przed stanowiskiem obsługi maleje, jeżeli odstęp czasu między przybyciem kolejnych klientów są dłuższe niż czas obsługi).

Walidacja zdarzeniowa – „zdarzenia” modelu symulacyjnego są porównywane ze zdarzeniami systemu rzeczywistego. Na przykład – przerwanie obsługi zgłoszenia mniej pilnego na korzyść zgłoszenia pilniejszego (w systemie medycznej pomocy doraźnej).

Testy warunków ekstremalnych – model powinien być odporny na wszelkie ekstremalne i mało prawdopodobne kombinacje czynników. (Na przykład: jeżeli popyt na pewien produkt spada do zera, sprzedaż też powinna wynosić zero).

Walidacja zewnętrzna (ang. *Face Validity*) – uzyskiwanie opinii osób niezwiązanych z projektem (ale związanych z systemem) na temat oceny zachowania się modelu.

Test stałych wartości – stałe wartości są wprowadzane pod wszystkie zmienne i parametry modelu. Umożliwia to sprawdzenie uzyskanych wyników poprzez ich obliczenie.

Historyczna walidacja – jeżeli istnieją dane historyczne, część z nich jest wykorzystana do budowy modelu, a część do przetestowania poprawności zachowania modelu.

Trójstopniowa analiza zasadności – (1) ustalanie założeń do modelu na podstawie teorii, obserwacji, wiedzy ogólnej, (2) empiryczne testowanie założeń, (3) sprawdzenie możliwości dokonania predykcji przez model.

Walidacja wewnętrzna – przeprowadzanych jest kilka przebiegów symulacyjnych w celu określenia wewnętrznej zmienności stochastycznej modelu. Duża jej wartość sprawia, że uzyskane wyniki mogą być poddane w wątpliwość.

Analiza wrażliwości – zmienianie wartości parametrów wejściowych i obserwowanie zmian zachodzących w zachowaniu się modelu. Te parametry, które powodują znaczące zmiany w zachowaniu się modelu, powinny być dobrane ze szczególną starannością. Oznacza to bowiem, że model jest *wrażliwy* na te parametry.

Walidacja predykcyjna – za pomocą modelu dokonywana jest predykcja zachowania się systemu i porównywane są uzyskane wyniki z danymi pochodzącymi z systemu. Dane z systemu można uzyskać na przykład poprzez przeprowadzenie specjalnych testów (badań).

Technika śledzenia – pewne obiekty są „śledzone” w trakcie ich wędrówki przez model i sprawdzana jest w ten sposób logika modelu.

Testy Turinga – eksperci (osoby związane z systemem) rozstrzygają, czy zaprezentowane wyniki pochodzą z modelu symulacyjnego, czy z systemu rzeczywistego.

Analiza wariancji – analiza ta stanowi zbiór metod analizy danych, który może być użyty do testowania hipotezy, że średnia wartość (lub wariancja) szeregów generowanych w komputerowych eksperymentach symulacyjnych jest równa średniej wartości (lub wariancji) odpowiadających im szeregów obserwowanych. Zastosowanie metody opiera się na trzech podstawowych założeniach: o normalności rozkładu, o niezależności w sensie statystycznym oraz o ogólnej wariancji.

Test chi-kwadrat – klasyczny test, który może służyć do testowania hipotezy, że zbiór danych generowanych przez model symulacyjny ma taki sam rozkład częstości, jak zbiór obserwowanych danych historycznych.

Test Kolmogorowa–Smirnowa – jest to klasyczny test *nieparametryczny*, dotyczący stopnia zgodności między rozkładem zbioru wartości z próby (szeregów symulowanych) a pewnym określonym rozkładem teoretycznym (rozkładem danych rzeczywistych). Test ten dotyczy określenia skumulowanego rozkładu częstości dla danych symulowanych oraz dla danych rzeczywistych. Traktuje on oddzielnie poszczególne obserwacje i – inaczej niż test chi-kwadrat – nie gubi informacji przez łączenie ze sobą różnych kategorii.

Inspekcje – techniki analizy wzrokowej produktów poszczególnych etapów.

Interpretacje – techniki polegające na zastąpieniu instrukcji programu operacjami symbolicznymi.

Imitacje – techniki imitowania akcji i zdarzeń w modułach programu.

Testowanie typu „czarna skrzynka” – techniki eksperymentowania z kodem programu traktowanym jak „czarna skrzynka” w celu sprawdzenia, czy cały program posiada przypisane mu cechy. Jest to testowanie w skali „makro”, które powinno dawać

odpowieź na pytanie: czy cały model odzwierciedla rzeczywistość z wystarczającą dokładnością?

Testowanie typu „przezroczysta skrzynka” – techniki eksperymentowania z kodem traktowanym jak „przezroczysta skrzynka” w celu sprawdzenia, czy moduł, grupa modułów lub cały program wykonują zaplanowane działania. Jest to testowanie w skali mikro, które powinno dawać odpowiedź na pytanie: czy każdy fragment modelu odzwierciedla rzeczywistość z wystarczającą dokładnością?

Testowanie regresywne – powtórzenie uprzednio przeprowadzonych testów, po dokonaniu w programie zmian, w celu potwierdzenia faktu całkowitego usunięcia poprzednio ujawnionego i zlokalizowanego błędu.

Pomiary osiągnięć – techniki eksperymentowania w celu pomiaru konkretnych parametrów fizycznych związanych z wykonywaniem programu.

6. PLANOWANIE EKSPERYMENTÓW SYMULACYJNYCH I ANALIZA WYNIKÓW

6.1. ASPEKTY PLANOWANIA EKSPERYMENTÓW SYMULACYJNYCH

Za pomocą symulacji stochastycznej możemy określić *typowe* zachowanie systemu, czyli takie zachowanie, które ma największą szansę zaistnienia. Równie często jesteśmy nie tyle zainteresowani poznaniem najbardziej prawdopodobnego scenariusza, ile chcielibyśmy określić prawdopodobieństwo zajścia najbardziej niekorzystnego przypadku. Bez względu jednak na to, jaki cel badań stawiamy przed sobą, za pomocą symulacji nie uzyskamy odpowiedzi na pytanie *jak zachowa się system?* Będziemy jednak mogli oszacować wartości średnie, prawdopodobieństwo zajścia lub niewystąpienia danego zdarzenia, długości przedziałów ufności. Czyli uzyskamy obraz systemu, z którym mielibyśmy najczęściej do czynienia, zakładając, że moglibyśmy powtarzać obserwację systemu bardzo wiele razy w identycznych warunkach.

Niewłaściwym, ale niestety często spotykanym podejściem jest jednokrotne uruchomienie symulacji o dowolnie wybranym czasie trwania, a następnie potraktowanie uzyskanych wyników jako poprawnych charakterystyk badanego systemu. Tymczasem, uzyskany komplet wartości jest wyłącznie pojedynczym przykładem realizacji zmiennych losowych. Ponowne uruchomienie symulacji może doprowadzić do wyników krańcowo odmiennych. Który komplet wartości wynikowych odzwierciedla *typowe* zachowanie badanego systemu? *Losowe wejście* oznacza również *losowe wyjście*. Aby wyniki symulacji można było traktować jako wiarygodny opis zachowania badanego systemu, należy w sposób właściwy zaplanować eksperymenty symulacyjne oraz przeprowadzić poprawną analizę statystyczną wyników symulacji.

Probabilistyczny charakter stochastycznych modeli symulacyjnych sprawia, że niezbędny jest taki sposób uruchomienia modelu, aby można było wychwycić *nie* jednostkowe (czyli przypadkowe) zachowanie, ale *typowy* obraz działania systemu. Planowanie eksperymentów symulacyjnych w sposób pozwalający uwzględnić losowość modelu zostanie omówione w podrozdziałach 6.2 i 6.3.

Osiągnięcie merytorycznego celu badań symulacyjnych to z kolei podejmowanie decyzji *przed* uruchomieniem symulacji, odnośnie do wyboru takiej konfiguracji parametrów modelu, która pozwoli uzyskać potrzebne informacje przy najmniejszej możliwej liczbie badań. Modele symulacyjne są najczęściej modelami złożonymi o dużej liczbie parametrów i zmiennych decyzyjnych. Przeprowadzenie wszystkich możliwych eksperymentów, badających wszystkie możliwe zestawy wartości wejściowych, jest najczęściej niemożliwe. Z kolei dobrze zaplanowany eksperyment symulacyjny jest o wiele bardziej efektywny niż metoda „prób i błędów”, która polega na realizacji przypadkowo zaplanowanych badań i obserwowaniu, co może się wydarzyć. Planowanie eksperymentów symulacyjnych w sposób pozwalający osiągnąć postawiony cel poznawczy zostanie omówione w podrozdziałach 6.4 i 6.5.

6.2. STRATEGIE EKSPERYMENTÓW SYMULACYJNYCH DLA MODELI STOCHASTYCZNYCH

Z punktu widzenia strategii prowadzenia badań modeli stochastycznych, większość symulacji można zaklasyfikować jako *symulacje pulsacyjne* (ang. *terminating simulations*), czyli takie, w których to model wskazuje warunki rozpoczynania i kończenia eksperymentu jako naturalne odzwierciedlenie sposobu działania systemu (np. każda instytucja czynna dla klientów w określonych godzinach doby) oraz *symulacje niezmiennicze* (ang. *steady-state simulations*), badające systemy pracujące bez przerwy przez (teoretycznie) nieskończony czas (przykładem może być pogotowie ratunkowe, które odbiera wezwania przez całą dobę, czy system produkcyjny pracujący w systemie trzyzmianowym). Rozróżnienie obu typów symulacji ma kluczowe znaczenie przy planowaniu strategii dla eksperymentów symulacyjnych modeli stochastycznych.

Symulacje pulsacyjne (terminating simulations)

Cechą charakterystyczną systemów pulsacyjnych jest naturalne niewypełnienie w momencie rozpoczynania pracy, tj. brak zgłoszeń w chwili startu, niezajęte i dostępne stanowiska obsługi, brak kolejek. Dokładnie taki stan odzwierciedla model symulacyjny wprawiony w ruch bez uprzedniego definiowania warunków początkowych. Pozyskanie statystycznie poprawnej próby z wynikami symulacji jest stosunkowo proste – wystarczy wykonać n niezależnych powtórzeń, zebrać wyniki z każdego powtórzenia i traktując uzyskane wartości badanego miernika (np. czasu pobytu w systemie czy całkowitego kosztu obsługi) jako realizacje zmiennej losowej, wyznaczyć wartości oczekiwane. Kluczowe znaczenie ma ustalenie właściwej liczby powtórzeń symulacyjnych. Przedział ufności, często obliczany automatycznie w pakietach symulacyjnych, może nas poinformować o jakości próby losowej. Dla przyjętego poziomu

ufności (najczęściej 95%) obliczony według wzoru (6.5) (por. podrozdział 6.6) przedział liczbowy jest „losowym” przedziałem, który z założonym prawdopodobieństwem (tj. 0,95 dla 95% poziomu ufności) zawiera wartość oczekiwaną szacowanego miernika wyjściowego. Im przedział ten jest węższy, tym precyzja oszacowania jest większa. Zwiększanie liczby powtórzeń powoduje kurczenie się przedziału ufności. Gdybyśmy byli w stanie wykonać nieskończenie wiele powtórzeń, przedział ufności skurczyłby się do zera i uzyskalibyśmy wartość punktową szacowanego parametru. Niestety, tempo kurczenia się przedziału ufności nie jest wprost proporcjonalne do tempa przyrostu liczby powtórzeń.

Tabela 6.1. Długość 95% przedziałów ufności dla średniego czasu produkcji wyrobów niewadliwych oraz braków, przy różnej liczbie powtórzeń symulacyjnych

Liczba powtórzeń symulacyjnych	Długość 95% przedziału ufności dla średniego czasu produkcji wyrobów niewadliwych	Długość 95% przedziału ufności dla średniego czasu produkcji braków
5	14,66	16,8
10	10,95	11,59
20	5,92	6,56
30	4,27	5,43
50	3,52	4,31
100	2,25	2,98

Średni czas procesów produkcji dla wyrobów niewadliwych i braków, uzyskany w pewnym eksperymencie symulacyjnym, dobrze ilustruje wyraźną tendencję zmniejszania się wartości przedziału ufności w miarę wzrostu liczby powtórzeń (por. tabela 6.1). Coraz intensywniejsze zwiększanie n skutkuje malejącym tempem pomniejszania długości przedziału ufności. W pewnym momencie nawet znaczące zwiększenie liczby powtórzeń (np. z 20 000 do 30 000) nie przyniesie istotnej poprawy precyzji oszacowania. Modelujący podejmuje zatem arbitralną decyzję o satysfakcjonującej go liczbie przebiegów symulacyjnych.

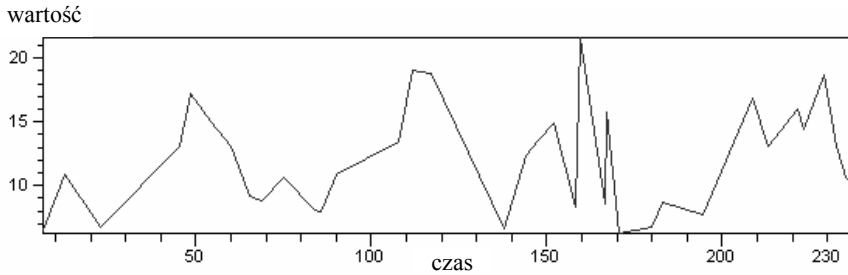
Więcej na temat analizy wyników symulacji Czytelnik znajdzie w podrozdziale 6.6.

Symulacje niezmiennicze (steady-state simulations)

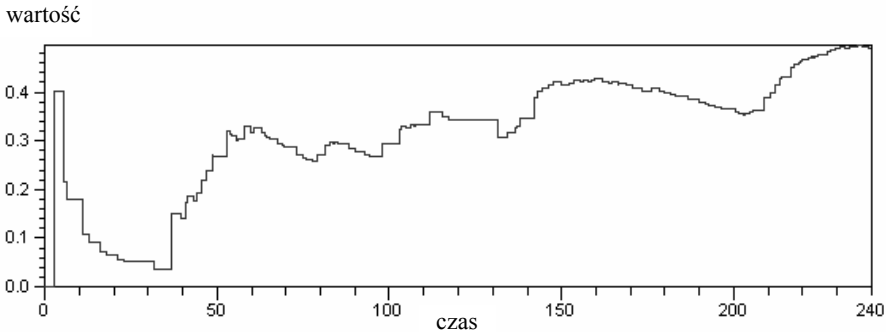
W symulacjach *niezmienniczych* planowanie eksperymentów jest trudniejsze, ponieważ musimy zmierzyć się z problemem warunków początkowych. W przeciwieństwie do symulacji pulsacyjnych, w których stan modelu w chwili uruchamiania eksperymentu idealnie odpowiadał pustemu i nieobciążonemu systemowi rozpoczynającemu swój kolejny cykl pracy, w przypadku symulacji niezmienniczej należy poradzić sobie z warunkami startu i zatrzymania oraz z niezgodnością między pustym modelem w chwili uruchamiania eksperymentu i wypełnionym (choć niekoniecznie zawsze w tym samym stopniu) systemem. Ta niezgodność powoduje, że przez pewien czas wyniki symulacji będą *niedoszacowane*. Aby uniknąć błędów

w interpretacji wyników, wskazane jest *rozgrzanie modelu* (ang. *warm-up period*) (czyli uruchomienie go, odczekanie aż wypełni się zgłoszeniami i zaniknie wpływ warunków początkowych), a następnie przystąpienie do zbierania statystyk.

Długość okresu rozgrzewania ustala się arbitralnie, obserwując wybrane wskaźniki, takie jak np. liczbę zgłoszeń przebywających w systemie, długość kolejek, czas pobytu w systemie. Zaleca się wybór takich mierników, dla których okres osiągnięcia stabilności jest stosunkowo długi. Dla pewnego przykładu założono, że system produkcyjny pracuje w ruchu ciągłym na trzy zmiany, spełniając warunki systemu o charakterze niezmienniczym. Symulację wykonano dla okresu 4 godzin, obserwując liczbę wyrobów przebywających w danej chwili w systemie (rys. 6.1) oraz średnią długość wszystkich kolejek (rys. 6.2). Wartości drugiego wskaźnika wyraźnie sugerują, że okres rozgrzewania modelu powinien wynosić około 60 minut.



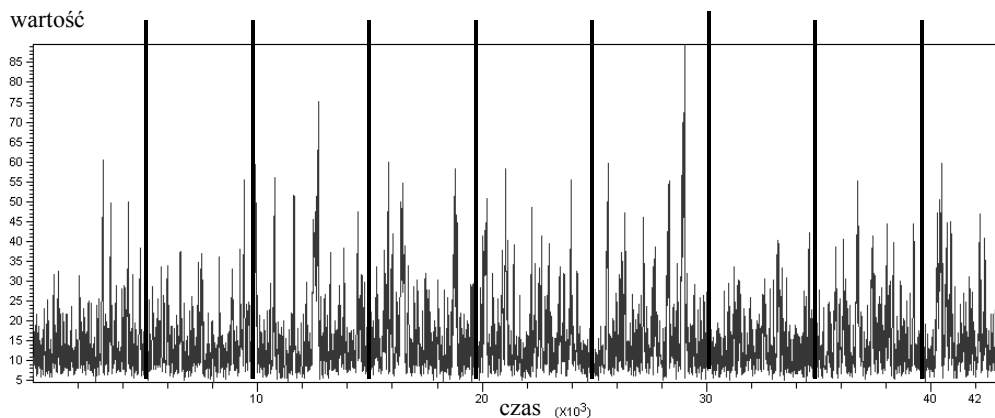
Rys. 6.1. Liczba zgłoszeń w systemie w ciągu 4 godzin (240 minut) symulacji
Źródło: opracowanie własne



Rys. 6.2. Średnia długość wszystkich kolejek w ciągu 4 godzin (240 minut) symulacji
Źródło: opracowanie własne

Jeżeli wyznaczony okres rozgrzewania się modelu jest stosunkowo krótki, można przyjąć dalszą strategię planowania eksperymentów taką, jak w przypadku symulacji pulsacyjnych, tzn. wykonać n niezależnych powtórzeń (każde powtórzenie należy rozpocząć okresem rozgrzewania modelu), a następnie przeprowadzić odpowiednią analizę statystyczną.

W przypadku modeli o długich i bardzo długich okresach rozgrzewania, przedstawiona strategia nie rokuje jednak pomyślnie. Zaleca się wtedy wykonanie jednego *niezwykle długiego* powtórzenia i tym samym tylko jednokrotnego poświęcenia czasu symulacji na rozgrzewanie modelu, a następnie podzielenie całego okresu symulacji na kilka większych odcinków* (por. rys. 6.3), wyznaczenie średnich w obrębie każdego odcinka i potraktowanie tak uzyskanych wartości jako próby losowej, którą można poddać właściwej analizie statystycznej. Średnie wyliczone dla każdego odcinka będą pełniły rolę średnich wyznaczonych dla każdego powtórzenia w przypadku wielokrotnego powtarzania symulacji. Obserwacje pochodzące z okolicy miejsc podziału będą niestety ze sobą skorelowane, aby więc wyznaczone średnie odcinkowe wykazywały jak najmniejszą korelację, odcinki na które dzielone jest powtórzenie powinny być jak największe. Zwiększanie długości odcinków skutkuje jednak wzrostem wariancji, dlatego często rozważanym problemem jest znalezienie optymalnej długości odcinków, czyli takiej długości, która zapewni rozsądną równowagę pomiędzy obciążeniem i wariancją.



Rys. 6.3. Podział jednego długiego powtórzenia (20 dni) na mniejsze odcinki.

Ilustracja zastosowania metody średnich odcinkowych

Źródło: opracowanie własne

6.3. METODY REDUKCJI WARIANCJI

W trakcie budowy modelu symulacyjnego wyróżnić można cztery fazy, w których szczególnie silnie uwidacznia się wpływ losowości zjawisk rzeczywistych na

* Ang. *batch means method*: metoda średnich odcinkowych lub inaczej metoda podziału szeregu czasowego.

proces modelowania. Są to etapy: (1) gromadzenia i przygotowania danych źródłowych, (2) uzyskania losowych charakterystyk zmiennych w modelu symulacyjnym, (3) weryfikacji i walidacji modelu, (4) planowania eksperymentów symulacyjnych i analizy wyników.

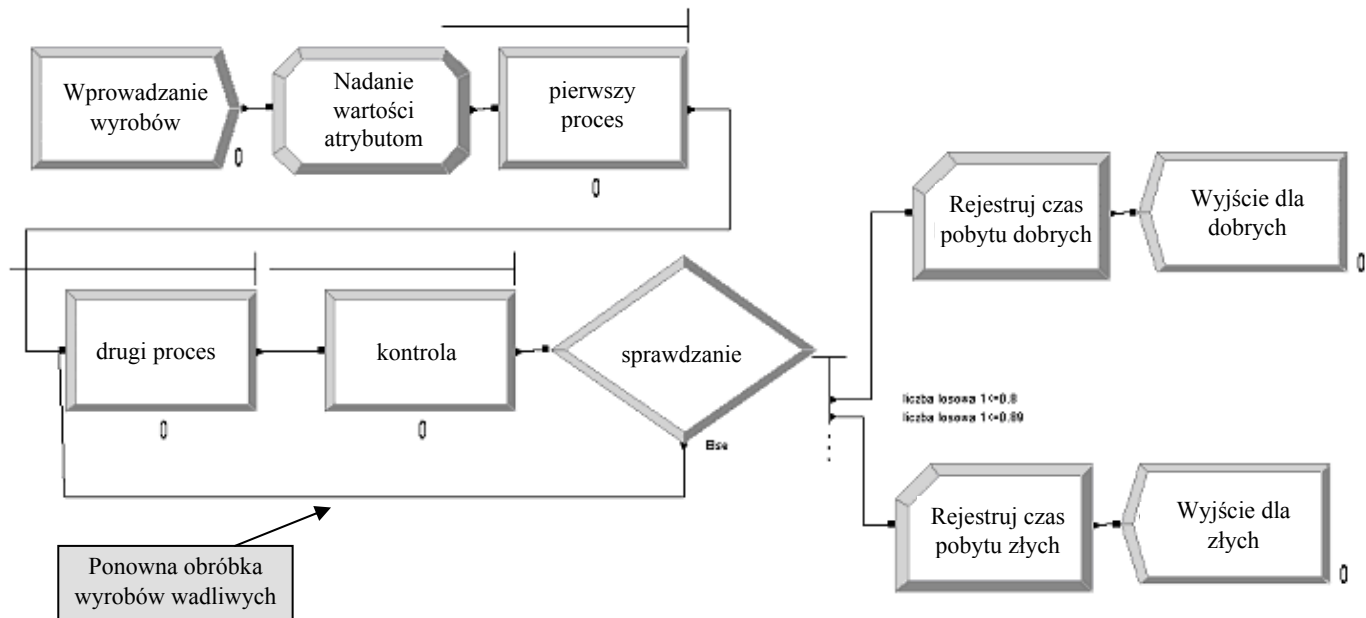
Pobieranie wartości z rozkładów losowych skutkuje losowym charakterem wyników symulacji i automatycznie wprowadza wariancję do pozyskiwanego na wyjściu zbioru wartości. Im wyższa wariancja, tym mniej precyzyjne będą wyniki, tym szersze przedziały ufności, tym większa niepewność rezultatów symulacji.

Bardzo pomocne w częściowym przynajmniej niwelowaniu wariancji są *metody redukcji wariancji* (ang. VRT: *variance-reduction technique*), z których na szczególną uwagę zasługuje metoda *jednakowych ciągów losowych* (ang. CRN: *common random numbers*). CRN, poprawnie zastosowana, pozwala na uzyskanie precyzyjniejszych rezultatów symulacji niejako „przy okazji”, bez dodatkowego inwestowania w proces modelowania. Jest to możliwe dzięki kontroli, którą za pomocą generatorów liczb losowych sprawujemy nad losowością odwzorowywanych zjawisk. Świadome i umiejętne manipulowanie mechanizmem procesu generowania zmiennych losowych na etapie budowy modelu może być przydatnym narzędziem ułatwiającym późniejsze zmniejszenie wariancji w wynikach symulacji. Ta niezwykle cenna, chociaż nie zawsze doceniana, własność stanowi jedną z istotniejszych zalet symulacji stochastycznej. Nie chcemy przecież pozbyć się losowości z budowanego modelu, skoro model ten ma poprawnie opisać zjawisko fizyczne, będące w samej swojej naturze losowym. Wysoka wariancja może nam jednak zaburzyć właściwą ocenę. Wysiłek, jaki wkładamy w projektowanie eksperymentów symulacyjnych, ma na celu właśnie zmniejszenie wariancji. Metody redukcji wariancji pozwalają dodatkowo, poprzez umiejętne korzystanie z generatorów liczb losowych, poprawić jakość danych wynikowych.

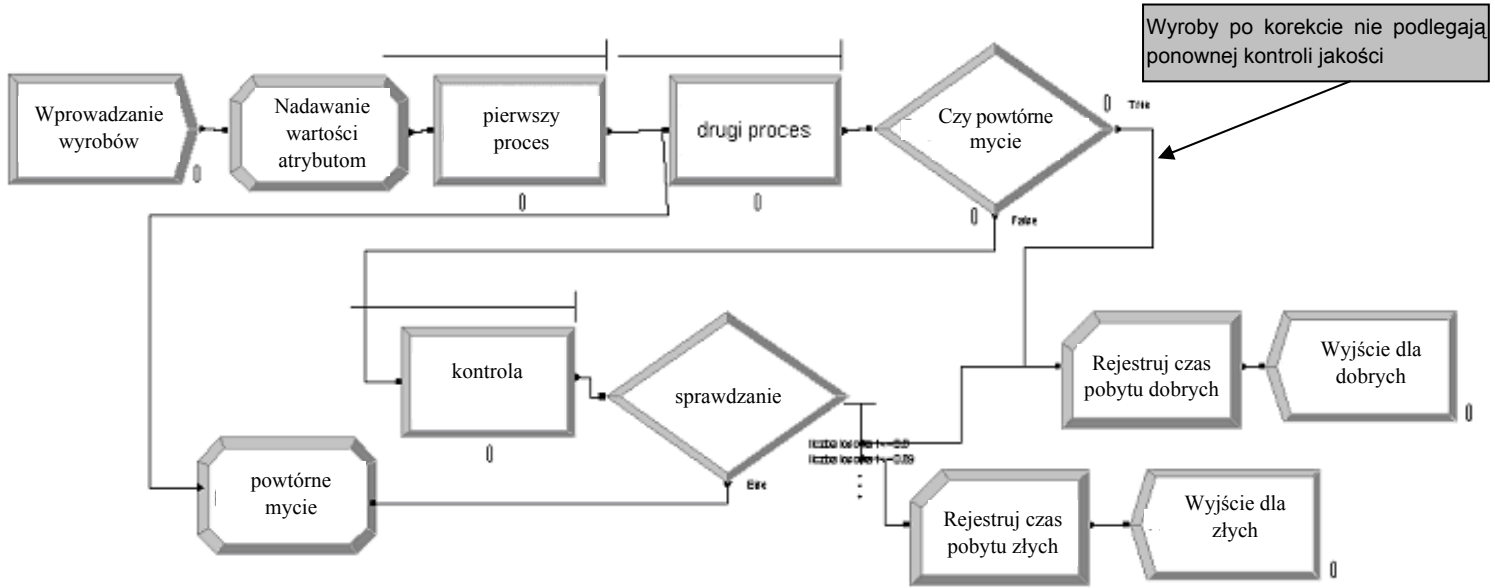
Metoda jednakowych ciągów losowych

Celem stosowania techniki CRN jest zapewnienie alternatywnym konfiguracjom modelu podobnych warunków eksperymentalnych. W przypadku zaobserwowania istotnych różnic w wartościach wskaźników będziemy wtedy mieli pewność, że zmiany te wynikają z różnic w konfiguracji modelu, nie są natomiast rezultatem fluktuacji w „warunkach eksperymentu”. Metoda CRN zakłada stosowanie jednakowych ciągów liczb losowych do generowania tych samych wielkości w różnych eksperymentach. Technicznie oznacza to konieczność synchronizacji, czyli wykorzystania tych samych liczb losowych do nadania wartości tym samym atrybutom w obu badanych konfiguracjach.

Metodę CRN zastosowano w prostym modelu systemu produkcyjnego (por. rys. 6.4). Wyroby, po wprowadzeniu na halę produkcyjną, poddawane są dwóm procesom obróbki, po których następuje kontrola jakości (wersja A modelu).



Rys. 6.4. Model prostego systemu produkcyjnego. Wersja A
 Źródło: opracowanie własne za pomocą pakietu Arena v.10.0



Rys. 6.5. Model prostego systemu produkcyjnego. Wersja B
Źródło: opracowanie własne za pomocą pakietu Arena v.10.0

Większość wyrobów, tzn. te, u których nie stwierdzono usterek, opuszcza halę. Wyroby, które nie spełniają wymogów, ale możliwa jest ich korekta, są ponownie poddawane obróbce w ramach drugiego procesu. Niewielki procent braków opuszcza halę w postaci osobnego potoku.

Model z rysunku 6.4 zmodyfikowano (wersja B modelu), wprowadzając założenie, że wyroby poddane korekcie nie podlegają już ponownej kontroli jakości i automatycznie uznane zostają za wyroby bez usterek (por. rys. 6.5).

Zastosowanie metody CRN wymaga określenia wszystkich losowych elementów modelu, a następnie uzależnienia ich losowego charakteru od odrębnych ciągów liczb losowych. W tabeli 6.2 zebrano losowe elementy modelu z rys. 6.4 (wersja A modelu) i przypisano im te same (model A_1) oraz różne (model A_2) ciągi liczb losowych. Dla zmienionej konfiguracji modelu również zdefiniowano wersję B_1 (jeden ciąg losowy dla wszystkich zmiennych losowych) oraz wersję B_2 (różne ciągi dla różnych elementów losowych).

Tabela 6.2. Wykaz losowych elementów modelu z rys. 6.4 wraz z przyporządkowanymi strumieniami liczb losowych dla badanych konfiguracji (A i B). Indeks 1 przy literach określających konfigurację modelu oznacza wersję z jednym ciągiem liczb losowych. Indeks 2 oznacza wersję z oddzielnymi ciągami liczb losowych dla poszczególnych atrybutów

Element losowy modelu	Rozkład losowy	Numer strumienia liczb losowych dla wersji A_1 i B_1	Numer strumienia liczb losowych dla wersji A_2 i B_2
Wprowadzanie wyrobów	WYKŁADNICZY(5)	10	10
Czas obsługi w pierwszym procesie	TRÓJKĄTNY(1, 3, 6)	10	1
Czas obsługi w drugim procesie	TRÓJKĄTNY(1, 3, 6)	10	2
Czas kontroli jakości	JEDNOSTAJNY(1, 2, 3)	10	3
Wynik kontroli jakości	Podział procentowy	10	4

Badania symulacyjne miały na celu porównanie wersji A i B modelu poprzez określenie średniego czasu pobytu w systemie wyrobów uznanych ostatecznie jako braki. Eksperyment przeprowadzono dwuetapowo, porównując (1) wersje A_1 oraz B_1 , czyli dwie alternatywne konfiguracje modelu, w których wszystkie elementy losowe są generowane z wykorzystaniem jednego ciągu liczb losowych oraz (2) wersje A_2 oraz B_2 , w których za każdy losowy element modelu odpowiedzialny jest inny (ale ten sam w obu konfiguracjach) ciąg liczb losowych. W przypadku etapu numer 2 zadbano o to, aby te same atrybuty w modelach A_2 i B_2 były generowane z wykorzystaniem tego samego ciągu liczb losowych.

Wyniki eksperymentu przedstawiono w tabeli 6.3. W celu wyraźniejszego zdemontowania wpływu synchronizacji ciągów liczb losowych na wariancję wyników symulacji, oba etapy eksperymentu powtórzono, zwiększając liczbę powtórzeń przebiegów symulacyjnych (wykonano 10, 20, 30 i 50 powtórzeń). Badanym miernikiem

był średni czas pobytu w systemie wyrobów uznanych ostatecznie jako braki. Skuteczność metody CRN obserwowano za pomocą 95% przedziału ufności obliczanego dla różnicy pomiędzy wartością oczekiwaną średniego czasu pobytu w systemie braków w konfiguracji A oraz wartością oczekiwaną średniego czasu pobytu w systemie braków w konfiguracji B. Uzyskane wyniki wyraźnie pokazują, że w kategoriach precyzji oszacowania wielkości wynikowych symulacji, zastosowanie metody CRN zmniejsza przedział ufności przy porównywaniu różnicy pomiędzy dwiema konfiguracjami tego samego modelu. Należy podkreślić, że zwiększoną dokładność uzyskano *bez ponoszenia żadnych dodatkowych nakładów* ani w trakcie budowy modelu, ani w trakcie wykonywania eksperymentów symulacyjnych.

Tabela 6.3. Zestawienie wielkości 95-procentowych przedziałów ufności obliczanych dla różnicy pomiędzy wartością oczekiwaną średniego czasu pobytu w systemie wyrobów uznanych jako braki w konfiguracji A minus ta sama wielkość w konfiguracji B. Indeks 1 oznacza wersję z jednym ciągiem liczb losowych. Indeks 2 oznacza wersję z oddzielnymi ciągami liczb losowych dla poszczególnych atrybutów

Liczba powtórzeń symulacyjnych	Długość 95% przedziału ufności przy porównaniu czasu pobytu braków w konfiguracji A_1 i B_1	Długość 95% przedziału ufności przy porównaniu czasu pobytu braków w konfiguracji A_2 i B_2
10	6,46	5,18
20	3,97	3,87
30	3,11	3,08
50	2,55	2,48

Różnicowanie przeciwstawne

Celem tej metody jest wymuszenie negatywnej korelacji. Pierwszą symulację wykonujemy wykorzystując liczby losowe dostarczane przez generator liczb losowych, ale w kolejnym powtórzeniu zastępujemy liczby losowe U_1, U_2, U_3, \dots przez $1 - U_1, 1 - U_2, 1 - U_3, \dots$. Losowość nie zostaje w ten sposób zaburzona, ponieważ jeżeli U są rozłożone równomiernie w $(0, 1)$, to $1 - U$ również. Zastosowana koncepcja polega na tym, że skoro „duże” U definiuje „małe” $1 - U$ (i odwrotnie), wyniki tak powiązanych ze sobą powtórzeń będą ujemnie skorelowane. Do analizy bierzemy zatem nie wyniki pierwszego powtórzenia, ale wartość średnią z dwóch kolejnych powtórzeń. W tak skonstruowanych przeciwstawnych parach, korelacja spowoduje, że uzyskana średnia będzie bardziej zbliżona do wartości rzeczywistej

$$\text{Cov}(X_1, X_2) = E[(X_1 - E(X_1))(X_2 - E(X_2))] \quad (6.1)$$

Kowariancja (por. (wzór 6.1)) jest miarą liniowego związku pomiędzy zmiennymi losowymi X_1 i X_2 . Może być dodatnia, ujemna lub równa zero. Jeżeli łączny rozkład jest taki, że w przypadku, gdy X_1 jest powyżej średniej, to również X_2 jest powyżej średniej – kowariancja jest dodatnia; a zatem małe X_1 wymusza również małe X_2 . Gdy kowariancja jest ujemna – duże X_1 wymusza małe X_2 . Jeżeli pomiędzy X_1 i X_2 nie ma związku liniowego – kowariancja jest równa zero.

Za pomocą kowariancji nie jesteśmy jednak w stanie oszacować wielkości tego powiązania, ponieważ kowariancja mierzona jest w jednostkach zmiennych losowych. Współczynnik korelacji (wzór (6.2)) pozwala zmierzyć siłę związku pomiędzy X_1 i X_2

$$\text{Cor}(X_1, X_2) = \frac{\text{Cov}(X_1, X_2)}{\sigma_{x_1} \sigma_{x_2}} \quad (6.2)$$

Współczynnik korelacji ma taki sam znak jak kowariancja i nie jest mierzony w jednostkach (będzie miał taką samą wartość niezależnie od wybranych jednostek miary). Współczynnik przyjmuje wartości pomiędzy -1 a 1 . Przykładowo, wartość $0,96$ lub $-0,98$ oznacza silną korelację (odpowiednio dodatnią lub ujemną), natomiast $0,1$ lub $-0,08$ słabą (co oznacza, że nie ma liniowego związku pomiędzy dwiema zmiennymi losowymi).

Różnicowanie kontrolne

Metoda wykorzystuje pomocniczą „kontrolną” zmienną losową, aby skorygować wyniki w górę lub w dół. Załóżmy, że kontrolną zmienną losową w pewnym eksperymencie jest czas pomiędzy nadejściem kolejnych zgłoszeń. Jeżeli wartości wygenerowane z tego rozkładu są mniejsze od wartości oczekiwanej (którą znamy, ponieważ to my zdefiniowaliśmy ten rozkład), powinniśmy odpowiednio korygować również mierniki działania systemu (wyniki symulacji).

Estymacja pośrednia

Metoda ta polega na tym, że estymujemy inny miernik, ale związany z interesującym nas miernikiem. Załóżmy, że chcemy oszacować czas pobytu klienta w systemie, na który składa się czas pobytu w kolejce i czas obsługi na stanowisku. Zamiast estymować ten czas z wyników symulacji, estymujemy jedynie czas pobytu w kolejce. Czas obsługi określamy, wyznaczając jego wartość oczekiwaną ze znanego nam rozkładu wejściowego. Usuwamy w ten sposób z rozwiązania wariację czasu obsługi na stanowisku.

6.4. MERYTORYCZNE PROJEKTOWANIE UKŁADU EKSPERYMENTÓW

Projektowanie układu eksperymentów symulacyjnych to podejmowanie decyzji *przed* uruchomieniem symulacji, odnośnie do wyboru takiej konfiguracji parametrów modelu, która pozwoli uzyskać potrzebne informacje przy najmniejszej możliwej liczbie badań. Dobrze zaplanowany eksperyment symulacyjny jest o wiele bardziej efektywny niż metoda „prób i błędów”, która polega na realizacji przypadkowo zaplanowanych badań i obserwowaniu, co może się wydarzyć.

Jak wspomniano w rozdziale 1, dwa pojęcia ściśle związane z projektowaniem układu eksperymentów to *czynnik* oraz *reakcja* (por. [Naylor 1974]). *Czynnikami* określa się parametry wejściowe i założenia do modelu, natomiast *reakcje* odnoszą się do mierników wyjściowych. Decyzja o zaklasyfikowaniu danego parametru do grupy *czynników* bądź *reakcji* związana jest z celem prowadzonych badań, a nie z wewnętrzną postacią modelu.

Tabela 6.4. Przykłady czynników i reakcji

System	Przykładowe czynniki	Ilościowe	Jakościowe	Kontrolowane	Niekontrolowane	Przykładowe reakcje
Linia produkcyjna	Liczba stanowisk	√		√?	√?	Przepustowość
	Dyscyplina kolejki		√	√		Wykorzystanie maszyn
	Prędkość przesuwu taśmy	√		√		Czas produkcji
Biuro podróży	Liczba pracowników	√		√		Koszt
	Płace	√		√		Wskaźnik strat
Magazyn	Średni czas pomiędzy zgłoszeniami popytu	√			√	Koszt składowania Koszt braku
	Wielkość zgłoszenia popytu	√			√	Zysk Stracone zamówienia
	Moment zamawiania	√		√		
	Wielkość zamówienia	√		√		

Czynniki mogą być *jakościowe* lub *ilościowe*, *kontrolowane* lub *niekontrolowane*. Niektóre przykłady czynników ilościowych i jakościowych, kontrolowanych i niekontrolowanych podano w tabeli 6.4. Należy zauważyć, że czynniki **nie są z definicji** kontrolowane lub niekontrolowane. Przykładowo, dla decydenta niższego szczebla liczba stanowisk będzie wielkością, na którą nie ma wpływu, natomiast decydent wyższego szczebla potraktuje ten czynnik jako wielkość kontrolowaną.

Jeden czynnik w modelu

Przy jednym *czynnikiem* w modelu, symulacja uruchamiana jest dla różnych wartości czynnika, tzw. *poziomów*. Wykonujemy n powtórzeń dla każdego poziomu. Dla 2 poziomów mamy zatem $2n$ powtórzeń.

k czynników w modelu

Jeżeli w modelu występuje k *czynników*, naszym celem może być np. ocena siły wpływu *czynników* na wskaźnik *reakcji* lub określenie oddziaływań pomiędzy *czynnikami*. Wybieramy jeden czynnik, „zamrażamy” pozostałe $k - 1$ czynników na pewnym poziomie i wykonujemy symulację dla kilku poziomów wybranego czynnika. Strategię powtarzamy dla pozostałych czynników. Strategia ta jest niestety dosyć nieefektywna (np. dla $k = 3$ czynniki, $m = 3$ poziomy przyjmowane przez te czynniki mamy $3^3 n$ powtórzeń do wykonania) i można ją stosować przy niewielkiej liczbie czynników, z których każdy może przyjmować tylko kilka poziomów wartości.

Na rysunku 6.6 przedstawiono plan eksperymentów symulacyjnych dla modelu biura podróży. Symulacja ma na celu zbadanie skutków różnych scenariuszy zatrudnienia pracowników do działów informacji i sprzedaży.

Czynniki (ang. *cowrds*) definiujemy jako liczbę dodatkowo zatrudnionych osób w dziale informacji (*Czynnik DodInf* = 0, 1 lub 2 osoby) oraz liczbę dodatkowo zatrudnionych osób w dziale sprzedaży (*Czynnik DodSprz* = 0, 1, 2 osoby).

Reakcje (ang. *Responces*) mierzone są kosztem związanym z zatrudnieniem dodatkowych osób (*Koszt*) oraz stratami związanymi z rezygnacją klientów na skutek długich kolejek (*Wskaźnik strat*).

Ponieważ $k = 2$ czynniki w modelu, a $m = 3$ poziomy każdego czynnika, daje nam to $3^2 = 9$ eksperymentów, z których każdy powinien być wykonany n razy $n = 10$, (kolumna *Reps* = *Replications*). Na rysunku 6.6 przedstawiono wykaz sześciu eksperymentów z dziewięciu zaplanowanych.

	Scenario Properties				Controls		Responses	
	S	Name	Program File	Reps	DodInf	DodSprz	Koszt	Wskaźnik stra
1	☛	bazowy	14 : Model3W	10	0	0	800.000	34.928
2	☛	Dodaj1Inf	14 : Model3W	10	1	0	900.000	34.873
3	☛	Dodaj2Inf	14 : Model3W	10	2	0	1000.000	34.873
4	☛	Dodaj1Sprz	14 : Model3W	10	0	1	1000.000	13.290
5	☛	Doda2Sprz	14 : Model3W	10	0	2	1200.000	4.066
6	☛	DodajPo1	14 : Model3W	10	1	1	1100.000	10.940

Rys. 6.6. Fragment planu eksperymentów dla $k = 2$ czynników i $m = 3$ poziomów każdego czynnika

Źródło: opracowanie własne

Analiza 2^k -czynnikowa

Efektywniejszą metodą planowania eksperymentów przy wielu czynnikach jest tzw. *analiza 2^k -czynnikowa*. Każdemu czynnikowi przypisujemy tylko dwa poziomy („+” i „-”), a następnie prowadzimy symulację dla wszystkich możliwych 2^k -kombinacji typu „*czynnik-poziom*”. Nie sformułowano żadnych szczegółowych zaleceń dotyczących wyboru poziomów, pozostawiając tę decyzję analitykowi. Również sposób przypisania znaków „+” i „-” nie ma specjalnego znaczenia, chociaż zwyczajowo znak „-” przypisywany jest poziomowi o mniejszej wartości.

Rozważmy przykład dla $k = 3$ czynników. Wykaz $2^k = 8$ kombinacji typu „*czynnik-poziom*” przedstawiono w tab. 6.5. R_j to reakcja systemu dla j -tej kombinacji czynników; np. R_6 to *reakcja* wynikająca z ustawienia czynnika 1 i 3 na poziomie „+”, a czynnika 2 na poziomie „-”.

Główny efekt każdego czynnika j to średnia *zmiana* w reakcji systemu spowodowana przesunięciem czynnika j z poziomu „-” na „+” przy zamrożeniu pozostałych czynników. Na przykład *główny efekt czynnika 1* obliczymy, dodając znak z kolumny czynnika 1 do kolumny reakcji i dzieląc uzyskaną sumę przez $2^{3-1} = 4$, czyli:

Tabela 6.5. Układ eksperymentów dla $k = 3$ czynników w analizie 2^k -czynnikowej
 Źródło: [Law i Kelton 1991]

Kombinacja czynników	Czynnik 1	Czynnik 2	Czynnik 3	Reakcja
1	-	-	-	R_1
2	+	-	-	R_2
3	-	+	-	R_3
4	+	+	-	R_4
5	-	-	+	R_5
6	+	-	+	R_6
7	-	+	+	R_7
8	+	+	+	R_8

$$\begin{aligned}
 e_1 &= \frac{-R_1 + R_2 - R_3 + R_4 - R_5 + R_6 - R_7 + R_8}{4} \\
 e_2 &= \frac{-R_1 - R_2 + R_3 + R_4 - R_5 - R_6 + R_7 + R_8}{4} \\
 e_3 &= \frac{-R_1 - R_2 - R_3 - R_4 + R_5 + R_6 + R_7 + R_8}{4}
 \end{aligned} \tag{6.3}$$

Zatem, e_j to różnica między średnią reakcją, gdy czynnik j jest na poziomie „+” a średnią reakcją, gdy czynnik j jest na poziomie „-”.

Siła powiązań pomiędzy czynnikami i oraz j to różnica pomiędzy średnią reakcją, gdy czynniki i oraz j są na tym samym poziomie (oba na „+” lub oba na „-”) a średnią reakcją, gdy są na przeciwnych poziomach. Na przykład siłę powiązań pomiędzy czynnikami 1 i 3 uzyskamy po pomnożeniu wierszami znaków odpowiednich kolumn „czynnikowych” i dołączeniu tak otrzymanego znaku do kolumny reakcji:

$$e_{13} = \frac{+R_1 - R_2 + R_3 - R_4 - R_5 + R_6 - R_7 + R_8}{4} \tag{6.4}$$

6.5. PRZYKŁAD ZASTOSOWANIA ANALIZY 2^k -CZYNNIKOWEJ

Rozważmy pewien model (por. [Law i Kelton 1991]) odnawiania zapasów magazynowych. Firma sprzedająca pewien produkt chciałaby określić strategię zamawiania kolejnych dostaw, tj. ustalić s , czyli wysokość zapasu minimalnego oraz d , czyli wielkość składanego zamówienia.

Czynniki ($k = 2$) to: s = wysokość zapasu minimalnego oraz d = wielkość składanego zamówienia.

Reakcja to średni całkowity koszt utrzymywania zapasów i utrzymywania portfela zamówień.

Celem symulacji jest zbadanie wpływu obu czynników na średni całkowity koszt operacyjny. W tabeli 6.6 przedstawiono specyfikację poziomów dla obu czynników, natomiast w tabeli 6.7 wyniki eksperymentu symulacyjnego, przeprowadzonego według analizy 2^k -czynnikowej w pojedynczym przebiegu symulacyjnym liczącym 120 miesięcy.

Tabela 6.6. Czynniki i poziomy w modelu odnawiania zapasów magazynowych
Źródło: [Law i Kelton 1991]

Czynnik	Poziom „-”	Poziom „+”
Zapasy minimalny s	20	60
Wielkość zamówienia d	10	50

Tabela 6.7. Plan eksperymentów i wyniki symulacji dla modelu odnawiania zapasów magazynowych
Źródło: [Law i Kelton 1991]

Układ czynników	s	d	$s \times d$	Reakcja
1	-	-	+	141,86
2	+	-	-	141,37
3	-	+	-	112,45
4	+	+	+	146,52

Główny efekt czynników s i d :

$$e_s = \frac{-141,86 + 141,37 - 112,45 + 146,52}{2^{2-1}} = 16,79$$

$$e_d = \frac{-141,86 - 141,37 + 112,45 + 146,52}{2} = -12,13$$

Siła powiązania pomiędzy czynnikami:

$$e_{sd} = \frac{141,86 - 141,37 - 112,45 + 146,52}{2} = 17,38$$

Wnioski z analizy uzyskanych wartości można sformułować następująco:

1. Średni skutek podniesienia s od 20 do 60 spowoduje wzrost średniego kosztu całkowitego o 16,79 (e_s).

2. Średni skutek podniesienia d od 10 do 50 spowoduje spadek średniego kosztu całkowitego o 12,13 (e_d).

3. Wskazany dobór wielkości s i $d \rightarrow$ to mała wielkość s i duża d . Naszym celem jest bowiem minimalizacja kosztu.

4. Ponieważ związek między s i d jest dodatni (e_{sd}), jest to kolejna wskazówka do ustalenia obu czynników na przeciwnych poziomach. Pozostawienie obu czynników na tych samych poziomach spowoduje, że przesuając się z wartości s i d o określonym znaku w kierunku wartości o znaku przeciwnym uzyskujemy wartość dodatnią ($e_{sd} = 17,38$). Oznacza to, że wartość początkowa była większa od wartości końcowej.

Oczywiście, ponieważ R_j jest zmienną losową, jedno powtórzenie nie wystarczy. Należy przeprowadzić tyle powtórzeń, aby uzyskane przedziały ufności dla e_s , e_d i e_{sd} były satysfakcjonujące.

6.6. ANALIZA STATYSTYCZNA WYNIKÓW

Większość standardowych technik statystycznej obróbki danych wymaga dysponowania zbiorem niezależnych obserwacji. Ważne jest zatem, aby eksperymenty symulacyjne były zaplanowane i przeprowadzone zgodnie z procedurą omówioną w rozdziale 6.2.

Szacowanie średniej i przedziałów ufności

Analiza wyników symulacji rozpoczyna się zazwyczaj od wyznaczenia przedziału ufności dla średniej. Przyjmijmy, że X_1, X_2, \dots, X_n to obserwacje uzyskane w kolejnych n powtórzeniach symulacyjnych, które stanowią pewien wycinkowy obraz zachowania systemu, reprezentowany przez parametry populacji μ (średnia) i σ (odchylenie standardowe). Wnioskowanie o tym, jakie wartości przyjmują parametry populacji (których przecież nie znamy, ale chcielibyśmy za pomocą symulacji poznać) przeprowadza się, wyznaczając statystyki z próby, czyli korzystając z obserwacji X_1, X_2, \dots, X_n . Średnia z próby, \bar{X} , jest estymatorem średniej z populacji μ , a odchylenie standardowe z próby, S , służy jako estymator odchylenia standardowego w populacji, σ . Statystyka z próby \bar{X} jest oczywiście zmienną losową, ponieważ jej wartość zależy od tego, jakie konkretne wartości uzyskaliśmy w kolejnych powtórzeniach. Jeżeli będziemy przeprowadzali kolejne eksperymenty, każdy składający się z n powtórzeń, to otrzymamy wiele wartości średnich \bar{X} . Okazuje się, że *niezależnie* od tego, czy rozkład samej populacji był normalny, czy nie, gdy wartość próby wzrasta nieograniczenie, wówczas rozkład średniej z próby \bar{X} dąży do rozkładu normalnego o średniej μ i odchyleniu standardowym σ/\sqrt{n} .

Mówi o tym tzw. centralne twierdzenie graniczne (por. [Aczel 2005]). Na podstawie twierdzenia można ustalić ponadto, jaki jest zakres, w którym średnia z próby powinna się zmieścić oraz jakie jest prawdopodobieństwo odbiegania wartości \bar{X} na określoną odległość od średniej populacji μ . Przykładowo, prawdopodobieństwo, że odległość między \bar{X} i μ nie przekroczy σ/\sqrt{n} wynosi 0,68. Przedziałowa ocena nazywana jest *przedziałem ufności*.

Przedziałem ufności nazywamy przedział liczbowy, o którym przypuszczamy, że mieści się w nim nieznaną parametr populacji. Miara pewności, że ten przedział naprawdę zawiera interesujący nas parametr, zwana jest *poziomą ufnością*.

Przedział ufności $(1 - \alpha)100\%$ dla μ , gdy σ nie jest znane (a z taką sytuacją mamy do czynienia w trakcie analizy wyników symulacji), a n jest dostatecznie duże ($n > 30$), wyznaczamy za pomocą wzoru:

$$\bar{x} \pm z_{\alpha/2} \frac{s}{\sqrt{n}} \quad (6.5)$$

gdzie: \bar{x} jest konkretną wartością zmiennej \bar{X} , $z_{\alpha/2}$ jest wartością standaryzowanej zmiennej losowej normalnej Z , $1 - \alpha$ nazywane jest współczynnikiem ufności. Współczynnik przemnożony przez 100 daje *poziomą ufność* wyrażony w procentach.

W modelach symulacyjnych bezpieczniej jest korzystać z innego wzoru (por. wzór 6.6), ze względu na nie zawsze dostateczną liczbę wykonanych powtórzeń symulacyjnych. Przedział ufności $(1 - \alpha)100\%$ dla μ , gdy σ nie jest znane, wyznaczamy za pomocą wzoru:

$$\bar{x} \pm t_{\alpha/2} \frac{s}{\sqrt{n}} \quad (6.6)$$

gdzie: \bar{x} jest konkretną wartością zmiennej \bar{X} , $t_{\alpha/2}$ jest wartością rozkładu t o $n - 1$ stopniach swobody, $1 - \alpha$ nazywane jest współczynnikiem ufności.

Załóżmy, że w pewnym eksperymencie symulacyjnym prostego systemu obsługi wykonano $n = 10$ powtórzeń i uzyskano następujące wyniki (tab. 6.8).

Tabela 6.8. Obliczenie przedziału ufności przy próbie $n = 10$

Parametr	Koszt całkowity
Średnia z próby	55530
Odchylenie standardowe z próby	3223,82
Liczba obserwacji	10
Długość połowy 95% przedziału ufności	1998,11
Przedział ufności	[53531,89; 57528,11]

Dziewięćdziesięciopięcioprocentowy przedział ufności obliczony dla średniej wartości kosztu całkowitego wynosi $55530 \pm 1998,11$ lub $[53531,89; 57528,11]$, co daje 3,6% błędu oszacowania punktowego. Możemy mieć 95% pewności, że przeciętny zysk całkowity znajdzie się pomiędzy 53532 zł a 57528 zł.

Jeżeli chcemy uzyskać bardziej precyzyjne oszacowanie, musimy przeprowadzić więcej powtórzeń. Wyniki symulacji dla $n = 80$ powtórzeń zestawiono w tab. 6.9. Tym razem błąd oszacowania punktowego wynosi już tylko 1,1%, a my mamy 95% pewności, że przeciętny zysk całkowity znajdzie się pomiędzy 57121 zł a 58404 zł.

Tabela 6.9. Obliczenie przedziału ufności przy próbie $n = 80$

Parametr	Koszt całkowity
Średnia z próby	57763
Odchylenie standardowe z próby	2929,26
Liczba obserwacji	80
Długość połowy 95% przedziału ufności	641,89
Przedział ufności	[57121,11; 58404,89]

Porównanie dwóch scenariuszy

W wielu badaniach symulacyjnych jesteśmy zainteresowani porównaniem różnych wersji (zmodyfikowanych scenariuszy, alternatywnych rozwiązań) tego samego modelu. Różnica może być niewielka i dotyczyć tylko wartości jednego wybranego parametru, lub też może polegać na fundamentalnej zmianie w założeniach logicznych. Chcemy na przykład zbadać, czy zwiększenie liczby stanowisk kasowych wpłynie znacząco na sprawność obsługi klientów w hipermarkecie. Możemy być również zainteresowani porównaniem różnych strategii odnawiania zapasów w hurtowni. W tego typu analizach również będziemy się posługiwać przedziałami ufności.

Tabela 6.10. Wyniki symulacji zysku ze sprzedaży ozdób choinkowych

Parametr	Zysk dla 30 tys. sztuk	Zysk dla 50 tys. sztuk
Średnia z próby	51731,1	51391,0
Liczba powtórzeń	80	80
Odchylenie standardowe z próby	1778,6	3680,8
Długość połowy 95% przedziału ufności	389,76	806,6
Przedział ufności	[51341,34; 52120,86]	[50584,4; 52197,6]

Przypuśćmy, że pewna sieć handlowa zastanawia się nad zamówieniem ozdób choinkowych w ilości 30 tys. lub 50 tys. sztuk. W badaniach symulacyjnych uzyskano wartości średniego zysku takie jak w tab. 6.10. Czy istnieją podstawy do stwierdzenia, że przeciętny zysk ze sprzedaży 30 tys. sztuk jest różny od przeciętnego zysku ze sprzedaży 50 tys. sztuk?

Ponieważ próby losowe są duże (przekraczają 30 powtórzeń), możemy wyznaczyć przedział ufności dla różnicy pomiędzy średnimi dwóch populacji według wzoru:

$$(\bar{x}_1 - \bar{x}_2) \pm z_{\alpha/2} \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \quad (6.7)$$

Ze wzoru wynika, że granice przedziału ufności dla różnicy między średnimi w dwóch populacjach są równe różnicy między średnimi z próby \pm wartość krytyczna z dla $(1 - \alpha)100\%$ poziomu ufności pomnożona przez standardowy błąd różnicy między dwiema średnimi z próby. Z to standaryzowany rozkład normalny, który jest dobrym przybliżeniem rozkładu t przy dużej liczbie stopni swobody [Aczel 2005]. 95-procentowy przedział ufności będzie zatem następujący:

$$(51731,1 - 51391,0) \pm 1,96 \sqrt{\frac{(1778,6)^2}{80} + \frac{(3680,8)^2}{80}} = [-555,72; 1235,9]$$

Ponieważ wartość zero zawarta jest w wyznaczonym przedziale, nie możemy zatem odrzucić hipotezy, że obie średnie są równe. Oznacza to, że nie ma znaczącej różnicy w przeciętnym zysku między zamówieniem 30 tys. i 50 tys. sztuk ozdób choinkowych.

Porównanie wielu scenariuszy

Często zachodzi konieczność porównania więcej niż dwóch scenariuszy. Możemy wtedy budować przedziały ufności dla różnic pomiędzy każdą parą średnich lub też możemy jednocześnie porównywać średnie z kilku populacji. Zaawansowana metoda statystyczna, która rozstrzyga o istnieniu różnic między średnimi w kilku populacjach nosi nazwę analizy wariancji i często określana jest kryptonimem ANOVA. Nazwa tej metody jest nieco myląca. Jej celem jest wykrywanie różnic pomiędzy *średnimi*, ale metoda ta wymaga zbadania *wariancji*. Stąd nazwa podejścia.

ANOVA jest metodą złożoną, wymagającą wielu obliczeń. Zainteresowanych odsyłam do bardzo bogatej literatury statystycznej. Tu tylko wyjaśnimy ogólną ideę tego podejścia. Przedmiotem naszego badania jest kilka scenariuszy, dla których prowadzimy badania symulacyjne. W wyniku symulacji uzyskujemy zbiory niezależnych powtórzeń, dla których przeprowadzamy test sprawdzający F . Wartość tego testu oraz odczytana wartość krytyczna przy danym poziomie istotności pozwala nam rozstrzygnąć, czy możliwe jest przyjęcie, czy odrzucenie hipotezy o tym, że średnie we wszystkich populacjach są sobie równe. Gdy liczba porównywanych populacji wynosi dokładnie 2, test staje się po prostu testem na równość dwóch średnich.

Jeżeli zerowa hipoteza ANOVA o równości średnich zostanie przyjęta (brak podstaw do stwierdzenia, że pomiędzy średnimi występują różnice), to nasza analiza jest

zakończona. Jeżeli jednak zerowa hipoteza zostanie odrzucona, to przeprowadza się dalsze testy i wyznacza przedziały ufności w celu ustalenia, które średnie się różnią i jaka jest skala tych różnic.

Optymalizacja

Symulacja nie jest metodą optymalizacji, ale poprzez wykonanie kolejnych eksperymentów możemy dążyć do wybrania najlepszego scenariusza. W poszukiwaniu optymalnego scenariusza pomagają nam zazwyczaj specjalne narzędzia dołączane do środowiska komputerowego, w którym prowadzimy symulację: np. Solver w Excelu, OptQuest w Arenie czy Evolutionary Optimizer w Extendzie. Nie będą one przedmiotem rozważań w tej książce.

7. METODA MONTE CARLO

7.1. WPROWADZENIE

Założmy, że chcemy oszacować spodziewany przyszłoroczny zysk ze sprzedaży pewnego produktu. W tym celu skorzystamy ze znanej formuły:

$$\text{Zysk} = \text{Wielkość Sprzedaży} (\text{Cena Jednostkowa} - \text{Koszt Jednostkowy}) - \text{Koszty Stałe} \quad (7.1)$$

Przy próbie podstawienia danych do formuły zaczynamy zastanawiać się nad wartościami poszczególnych parametrów. Jeżeli działamy w stabilnym otoczeniu, *Cenę Jednostkową*, *Koszt Jednostkowy* oraz *Koszty Stałe* możemy określić z dużą dokładnością (choć nie mamy gwarancji utrzymania się tych wielkości na spodziewanym przez nas poziomie). Ostatni parametr formuły, tj. *Wielkość Sprzedaży* możemy starać się wstępnie oszacować, ale nasze obliczenia będą niedokładne. Wartość tego parametru jest związana z popytem, który z kolei zależy od wielu czynników i nie jesteśmy w stanie precyzyjnie go wyznaczyć.

Szacunkowe badania rynku pozwoliły określić ramy zmienności parametrów decyzyjnych:

Wielkość sprzedaży: od 1000 do 2000 sztuk.

Cena jednostkowa: od 21 do 25 zł/szt.

Koszt jednostkowy: od 15 do 19 zł/szt.

Koszty stałe: 5000 zł.

Założmy, że możliwe są trzy scenariusze (por. rys. 7.1): *Pesymistyczny*, *Optymistyczny* oraz *Neutralny*. Dla wszystkich wariantów wyznaczamy *Zysk*; w przypadku realizacji scenariusza pesymistycznego nie odnotujemy żadnego zysku (a nawet poniesiemy stratę), natomiast w scenariuszu optymistycznym nasz zysk wyniesie 15 000 zł. Ostrożny scenariusz pozwoli nam uzyskać 4000 zł zysku. Te trzy scenariusze nie wyczerpują jednak wszystkich przypadków, które mogą wystąpić. Ich liczba jest dużo większa i obejmuje nieskończenie wiele kombinacji wartości trzech zmiennych decyzyjnych. Chcielibyśmy wiedzieć, który scenariusz (nie tylko spośród trzech opisanych) jest najbardziej prawdopodobny i jaka jest szansa jego wystąpienia.

1 Metoda Monte Carlo - szacowanie zysku ze sprzedaży				
2				
3				
4	Scenariusz	Pesymistyczny	Neutralny	Optymistyczny
5				
6				
7	Wielkość sprzedaży	1000,00	1500,00	2000,00
8	Cena jednostkowa	21,00	23,00	25,00
9	Koszt jednostkowy	19,00	17,00	15,00
10				
11				
12	Koszty stałe	5000,00		
13				
14	Zysk	-3000,00	4000,00	15000,00
15				

Rys. 7.1. Możliwe scenariusze przy szacowaniu zysku
Źródło: opracowanie własne

W przykładzie trzy wielkości: *Wielkość sprzedaży*, *Cena jednostkowa* i *Koszt jednostkowy* zmieniają się w sposób losowy. Aby losowość tę ująć w modelu, możemy zastosować *metodę Monte Carlo*, która pozwala badać szansę wystąpienia danego zjawiska poprzez tzw. losowe próbkowanie.

Rozwój symulacji Monte Carlo został zapoczątkowany podczas II wojny światowej (por. podrozdział 1.1) najpierw w fizyce i chemii, a potem również w innych dziedzinach. Zieliński [1970] podaje, że metoda Monte Carlo oznacza *rozwiązywanie zadań numerycznych za pomocą eksperymentów statystycznych*. Niektórzy autorzy (por. [Wit 2004], [Fishman 1996]) piszą o *metodach* Monte Carlo, definiując je jako metody obliczeniowe, które polegają na rozwiązywaniu zadań probabilistycznych poprzez symulację komputerową lub po prostu jako metody, które wymagają użycia generatorów liczb losowych.

Termin *metoda Monte Carlo* jest najczęściej stosowany w odniesieniu do symulacji, w których czas nie odgrywa istotnej roli [Evans i Olson 2002]. Celem takiej symulacji jest zwykle określenie rozkładu losowego zmiennej wyjściowej, który zależy od pewnej liczby losowych rozkładów wejściowych. Istotną rolę w metodzie tej odgrywa losowanie (wybór przypadkowy) wielkości charakteryzujących proces poprzez tzw. *losowe próbkowanie*. Zamiennie używane w literaturze określenia to *symulacja statyczna* lub *symulacja stochastyczna w arkuszu kalkulacyjnym*, ponieważ właśnie arkusz kalkulacyjny jest najczęściej wykorzystywanym środowiskiem do prowadzenia symulacji metodą Monte Carlo.

Obok głównego, statycznego kierunku zastosowań, podejście Monte Carlo można jednak również zastosować do prowadzenia symulacji dynamicznej, w trakcie której obserwuje się zmiany systemu zachodzące w czasie i wykorzystuje się informacje zebrane podczas wcześniej symulowanych okresów. Pod koniec rozdziału zostaną zaprezentowane przykładowe modele kolejkowe zbudowane w arkuszu kalkulacyjnym z wykorzystaniem metody Monte Carlo, opracowane zgodnie z trzema technikami symulacji zdarzeń dyskretnych, które zostały omówione w podrozdziale 4.5.

7.2. ETAPY METODY MONTE CARLO

Symulację metodą Monte Carlo wykonuje się poprzez realizację kolejnych kroków, po uprzednim sformułowaniu deterministycznej postaci modelu matematycznego i zapisaniu go w postaci formuł logicznych w arkuszu kalkulacyjnym. W literaturze (por. [Powell i Baker 2004]) najczęściej wymienia się pięć głównych etapów, jednakże wydaje się, że dla większej czytelności procedury realizacji badania symulacyjnego należałoby wyodrębnić pięć faz głównych oraz sześć faz szczegółowych.

Etap 1. Określenie rozkładów losowych wybranych zmiennych wejściowych

1.1. Wybór zmiennych opisywanych losowo

Decyzja o wskazaniu losowych parametrów modelu wydaje się być oczywista, jednakże zaleca się przypisanie losowości tylko tym elementom, które mają największy wpływ na szacowaną zmienną wynikową. Wprowadzanie kolejnych zmiennych o charakterze losowym do modelu automatycznie zwiększa wariancję pozyskiwanego na wyjściu zbioru wartości (por. [Mielczarek 2005]), nie zawsze jest natomiast niezbędne ze względu na celowość budowy modelu.

1.2. Określenie postaci rozkładu losowego

W zależności od tego, czy losowy charakter wyróżnionej zmiennej przejawia się w sposób dyskretny czy ciągły, należy doprowadzić, poprzez analizę danych wejściowych, do sformułowania dystrybuanty skumulowanej dla każdej wybranej dyskretnej zmiennej losowej lub/oraz do zapisania odpowiedniej teoretycznej formuły funkcyjnej dla każdej ciągłej zmiennej losowej.

Etap 2. Wybór zmiennych wyjściowych

Symulacja może doprowadzić do wygenerowania wynikowych rozkładów losowych dla bardzo wielu zmiennych. Jedną z zalet symulacji jest między innymi możliwość koncentrowania się na więcej niż jednym parametrze wyjściowym.

Etap 3. Przeprowadzenie jednego pełnego eksperymentu symulacyjnego

3.1. Generowanie ciągów liczb losowych oddzielnie dla każdej zmiennej losowej

Uzyskanie w modelu losowych zmian zgodnych z wybranym rozkładem obejmuje dwa etapy: generowanie liczb losowych o rozkładzie równomiernym z przedziału $(0, 1)$, a następnie przekształcanie uzyskanych liczb losowych w dowolny rozkład zmiennej losowej czy w dowolny proces stochastyczny. Zaleca się uzależnienie losowego charakteru poszczególnych zmiennych od odrębnych ciągów liczb losowych. W powszechnie stosowanym arkuszu kalkulacyjnym *Excel* do generowania liczb losowych wykorzystuje się funkcję *LOS()*.

3.2. Wykorzystanie uzyskanych liczb losowych do wyznaczenia wartości zmiennych losowych

Formuły niezbędne do przekształcenia liczb losowych w zmienne losowe są w większości automatycznie dostępne w standardowej wersji arkusza kalkulacyjnego lub też – w kilku nielicznych przypadkach – wymagane jest samodzielne skonstruowanie prostych formuł.

3.3. Wprowadzenie zmiennych losowych do analitycznej postaci modelu

Model analityczny wiąże ze sobą dane wejściowe (deterministyczne i pozyskiwane z rozkładów losowych) z miernikami wyjściowymi poprzez formuły logiczne.

3.4. Wykonanie n powtórzeń ($n \rightarrow \infty$)

Wykonanie pojedynczego eksperymentu symulacyjnego wymaga pobrania n losowych wartości z rozkładu wejściowego, wprowadzenia tych wartości do formuł modelu, wyznaczenia wartości wynikowych i przechowania ich do późniejszych oszacowań. Im więcej powtórzeń zostanie wykonanych, tym bardziej precyzyjne będą analizy końcowe.

Etap 4. Przeprowadzenie pełnego badania symulacyjnego, tj. wykonanie k eksperymentów ($k \gg 1$)

Pełny eksperyment symulacyjny wymaga w niektórych przypadkach wielokrotnego powtórzenia eksperymentu pojedynczego, czyli wykonania k razy n powtórzeń. Takie postępowanie zalecane jest przy symulacji *typowego* zachowania systemu. Zwiększanie liczby powtórzeń n oraz liczby eksperymentów k zapewnia nam największą precyzję przy szacowaniu średniej z próby.

Etap 5. Analiza wyników

Wyniki symulacji przedstawiane są za pomocą wartości średnich oraz w postaci histogramów, które interpretuje się jako rozkłady losowe zmiennych wynikowych. Miarą precyzji wyników są najczęściej: wariancja, przedział ufności oraz średni błąd kwadratowy.

7.3. OKREŚLENIE ROZKŁADÓW LOSOWYCH WYBRANYCH ZMIENNYCH WEJŚCIOWYCH

O tym, jak przeprowadzić procedurę doboru rozkładu losowego do obserwowanego przez nas losowego zachowania zmiennej była mowa w rozdziale 2.8. W tym momencie musimy tylko uświadomić sobie, że losowy charakter zmiennej może przejawiać się w sposób *dyskretny* lub *ciągły*, a ponadto może on być opisany wzorem funkcyjnym (tzw. *rozkład teoretyczny*), np. funkcją gęstości rozkładu normalnego:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \text{ dla wszystkich } x \in R \quad (7.2)$$

lub za pomocą *rozkładu empirycznego* (podajemy wtedy tylko wartości i zakres zmiennej losowej). Rozkład empiryczny budujemy najczęściej wtedy, gdy: (a) nie potrafimy dopasować rozkładu teoretycznego, (b) zmienna losowa przyjmuje mało wartości lub (c) mamy zbyt mało danych do przeprowadzenia procedury doboru rozkładu.

7.3.1. PRZYKŁADY ZMIENNYCH O CHARAKTERZE DYSKRETNYM

Przykład 1. *Produkcja pieczarek*. Wzrost pieczarek jest ściśle uzależniony od temperatury otoczenia, a zatem w modelu musimy uwzględnić parametr opisujący wpływ pogody na produkcję. W tym celu zbieramy informacje z zeszłego roku o liczbie dni bardzo ciepłych (powyżej 25 °C), ciepłych (od 18 °C do 25 °C), chłodnych (od 10 °C do 18 °C) i zimnych. Uzyskujemy dane jak na rys. 7.2. Ponieważ interesują nas tylko cztery typy dni, możemy zbudować rozkład dyskretny, empiryczny. Zmienna losowa (*Pogoda*) przyjmuje cztery różne wartości, które dla ułatwienia zakodujemy jako 1, 2, 3, 4.

Metoda Monte Carlo – uprawa pieczarek	
Dane na temat pogody z zeszłego roku:	
Dni bardzo ciepłe (1)	32
Dni ciepłe (2)	78
Dni chłodne (3)	158
Dni zimne (4)	97

Rys. 7.2. Dyskretny charakter zmiennej *Pogoda*

Przykład 2. *Sprzedajemy rowery*. Prowadzimy sklep rowerowy. Ze względu na wysokie koszty magazynowania chcielibyśmy określić liczbę rowerów, które powinniśmy mieć każdego dnia w naszym sklepie. W tym celu badamy sprzedaż zeszłoroczną (por. rys. 7.3).

Metoda Monte Carlo – sprzedaż rowerów	
Dane na temat zeszłorocznej sprzedaży:	
Liczba dni	Sprzedaż
10	0
20	1
40	2
60	3
40	4
30	5
Razem	200

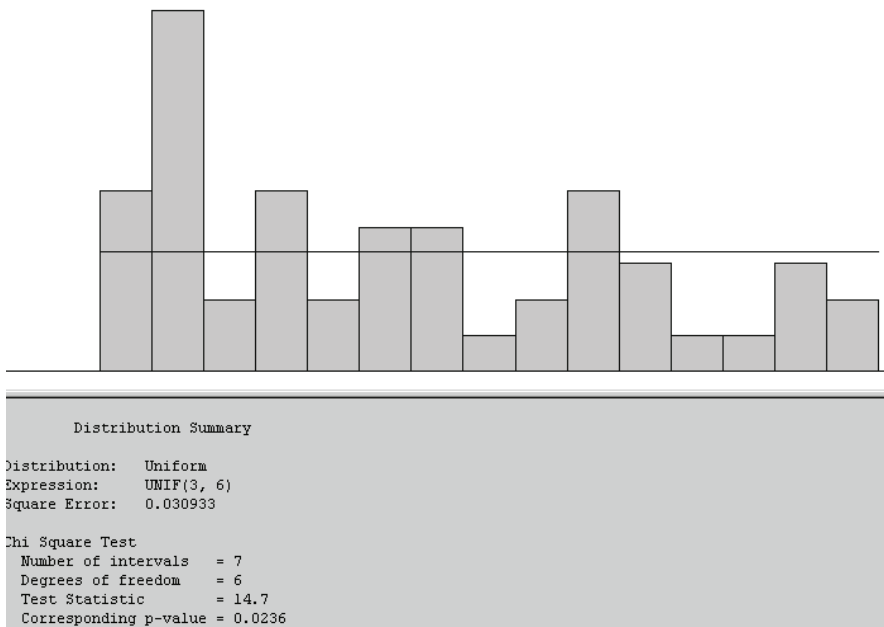
Rys. 7.3. Dyskretny charakter zmiennej *Sprzedaż*

I w tym wypadku powinniśmy zbudować rozkład empiryczny, ponieważ zmienna losowa przyjmuje zaledwie kilka wartości.

7.3.2. PRZYKŁADY ZMIENNYCH O CHARAKTERZE CIĄGLYM

Przykład 3. *Analizujemy pracę gniazda produkcyjnego.* W tym celu musimy określić czas trwania obsługi na stanowisku (w minutach). Zebraliśmy następującą próbkę danych:

3.34 3.96 4.10 4.21 4.87 3.79 3.04 5.84 3.70 3.18 3.52 3.66 5.05 4.21 4.36
5.89 4.81 3.27 5.74 3.86 3.03 3.29 3.26 3.25 3.10 4.82 3.30 5.14 4.09 3.10
3.54 4.09 4.73 3.39 3.26 4.25 5.55 3.27 4.48 5.08 3.39 5.74 4.97 4.84 5.28
4.13 3.69 3.66 5.78 4.75



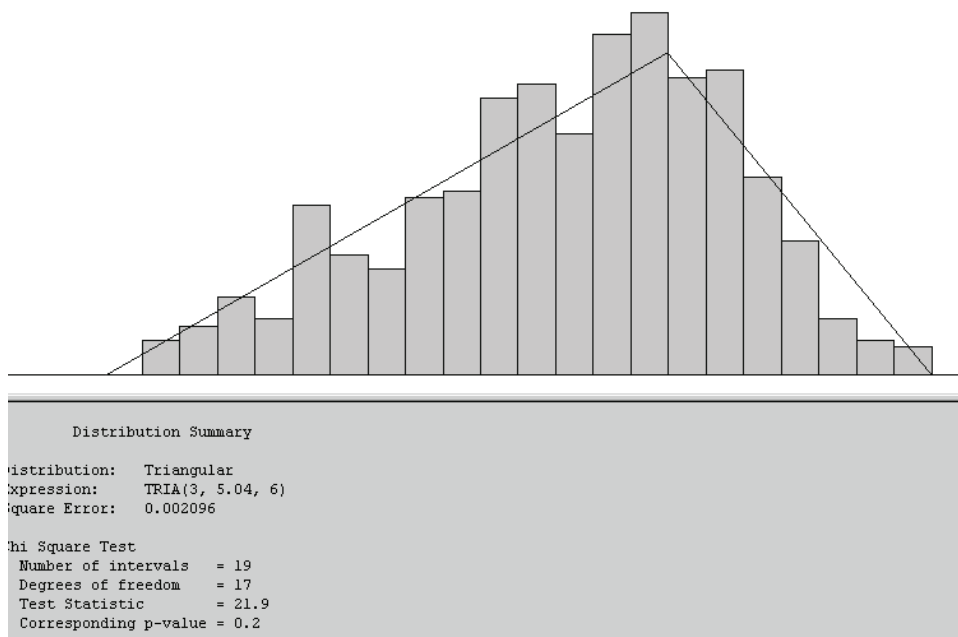
Rys. 7.4. Ciągły charakter zmiennej *Czas trwania obsługi*
Źródło: opracowanie własne

Możemy domyślać się, że próba ta jest odzwierciedleniem jakiegoś ciągłego rozkładu teoretycznego i chcielibyśmy określić postać takiego rozkładu. Wzrokowa ocena histogramu zbudowanego z zebranych danych pozwala jedynie typować pewien typ rozkładu (ciągła pozioma linia na rys. 7.4).

Przykład 4. *Usprawniamy funkcjonowanie hali obsługi w banku.* W tym celu zebraliśmy dane o czasie pobytu przy okienku kasowym kilkudziesięciu klientów (w minutach):

3.82 4.38 4.48 4.55 4.93 4.26 3.31 5.60 4.18 3.61 4.02 4.15 5.03 4.55 4.65
 5.67 4.90 3.74 5.49 4.31 3.28 3.76 3.73 3.71 3.46 4.91 3.77 5.07 4.48 3.46
 4.04 4.48 4.86 3.88 3.72 4.58 5.32 3.74 4.72 5.04 3.89 5.49 4.98 4.91 5.15
 4.50 4.18 4.15 5.54 4.87 5.54 4.23 4.18 5.50 4.24 4.75 4.03 5.17 5.75 5.39
 4.98 4.46 5.09 4.62 4.72 5.21 4.40 3.98 4.53 5.35 5.21 4.72 4.87 3.75 4.97
 5.21 4.91 4.27 4.56 5.60 5.32 4.63 3.54 4.89 4.81 5.19 5.09

Typowany przez nas typ rozkładu pokazuje ciągła linia na rys. 7.5.



Rys. 7.5. Ciągły charakter zmiennej *Czas obsługi klientów w banku*
 Źródło: opracowanie własne

7.3.3. OKREŚLENIE DYSTRYBUANTY DLA ROZKŁADÓW DYSKRETYCH

W przypadku rozkładów dyskretnych, empirycznych musimy wyznaczyć wartości prawdopodobieństwa skumulowanego, czyli dystrybuanty (por. tab. 7.1 i 7.2).

Tabela 7.1. Dystrybuanta dla przykładu 1 *Pieczarki*

Wartość zmiennej losowej (pogoda)	Numeryczna wartość zmiennej losowej	Liczba wystąpień	Prawdopodobieństwo $P()$	$P()$ skumulowane
Dni bardzo ciepłe	1	32	$32/365 = 0,09$	0,09
Dni ciepłe	2	78	$78/365 = 0,21$	$0,09 + 0,21 = \mathbf{0,30}$
Dni chłodne	3	158	$158/365 = 0,43$	$0,3 + 0,43 = \mathbf{0,73}$
Dni zimne	4	97	$97/365 = 0,27$	$0,73 + 0,27 = \mathbf{1,00}$
	Suma	365	1	

Tabela 7.2. Dystrybuanta dla przykładu 2 Rowery

Wartość zmiennej losowej (sprzedaż)	Numeryczna wartość zmiennej losowej	Liczba wystąpień	Prawdopodobieństwo $P()$	$P()$ skumulowane
0	0	10	$10/200 = 0,05$	0,05
1	1	20	$20/200 = 0,10$	0,15
2	2	40	$40/200 = 0,20$	0,35
3	3	60	$60/200 = 0,30$	0,65
4	4	40	$40/200 = 0,20$	0,85
5	5	30	$30/200 = 0,15$	1,00
Suma		200	1	

7.3.4. OKREŚLENIE POSTACI FUNKCJI GĘSTOŚCI I DYSTRYBUANTY DLA ROZKŁADÓW CIĄGŁYCH

Przykład 3. *Gniazdo produkcyjne*

Przyjmujemy (por. podrozdz. 2.8), że czas wykonania czynności na badanym stanowisku będzie poprawnie opisany przez **rozkład jednostajny**.

Rozkład jednostajny to zwykle pierwszy rozkład, który przyjmuje się w badaniach symulacyjnych wszędzie tam, gdzie podejrzewa się losowość zdarzeń w pewnym przedziale wartości od a do b . Jedyne informacje, które należy mieć, aby wygenerować wartości według tego rozkładu to liczby a i b . Wartość średnia leży dokładnie pomiędzy wartościami a i b , natomiast każda wartość pomiędzy a i b jest jednakowo prawdopodobna. W przykładzie: $a = 3$, $b = 6$ (por. podrozdział 2.5.1).

Przykład 4. *Bank*

Analizując zebrane dane (por. podrozdz. 2.8) przyjmujemy, że czas obsługi przy okienku kasowym opisany jest **rozkładem trójkątnym niesymetrycznym**.

Rozkład trójkątny charakteryzowany jest przez trzy wielkości: minimalną, najbardziej prawdopodobną i maksymalną. Decydenci pytani o losowe zachowanie danego parametru rzadko powołają się na konkretny rozkład losowy, natomiast są w stanie przytoczyć właśnie te trzy wartości. Dlatego z rozkładem tym często spotkamy się w modelach symulacyjnych. W przykładzie: $a = 3$, $c = 5,04$, $b = 6$ (por. podrozdział 2.5.4).

7.4. WYBÓR ZMIENNYCH WYJŚCIOWYCH

Powróćmy do przykładu wprowadzającego (podrozdz. 7.1). Celem badań symulacyjnych będzie analiza *Zysku ze sprzedaży*. Będziemy chcieli znaleźć odpowiedź m.in. na następujące pytania:

Jakie jest prawdopodobieństwo, że zanotujemy zysk?

Jakie jest prawdopodobieństwo, że zanotujemy stratę?

Jakie jest prawdopodobieństwo, że uzyskany zysk będzie większy niż 5000 zł?

Jakie jest prawdopodobieństwo, że uzyskany zysk będzie większy niż 15 000 zł?

7.5. PRZEPROWADZENIE JEDNEGO PEŁNEGO EKSPERYMENTU SYMULACYJNEGO

7.5.1. GENEROWANIE LICZB LOSOWYCH

Uzyskanie wartości zmiennej losowej, według uprzednio zdefiniowanej dystrybuanty, wymaga pozyskania liczb losowych (o liczbach losowych, ich własnościach statystycznych, historii pozyskiwania – patrz rozdz. 3). Liczby losowe, które wykorzystujemy w modelowaniu należą do przedziału (0, 1). Dlaczego? Ponieważ bardzo łatwo przekształcić je na dowolną liczbę rzeczywistą, przemnażając przez dowolny mnożnik dziesiętny. W arkuszu kalkulacyjnym EXCEL do pozyskiwania liczb losowych stosujemy bezargumentową funkcję $LOS()$. Po każdym wywołaniu funkcja ta zwraca wartość z przedziału (0, 1), a dokładniej liczby większe lub równe 0 i mniejsze od 1. Co kryje się pod skrótem $LOS()$? Jest tam „zaszyty” algorytm generatora liczb pseudolosowych (więcej o generatorach również w rozdz. 3). Nad coraz lepszymi, coraz bardziej wydajnymi, o coraz lepszych własnościach statystycznych generatorami pracują kolejne pokolenia naukowców, a my korzystamy z tych osiągnięć. Teoretycznie moglibyśmy oczywiście przygotować sobie wielki kapelusz z karteczkami i każdorazowo losować jedną karteczkę, ale nietrudno wyobrazić sobie uciążliwość takiej metody.

7.5.2. GENEROWANIE WARTOŚCI ZMIENNEJ LOSOWEJ

Rozkłady empiryczne – dyskretne

Zbudujmy model symulacyjny metodą Monte Carlo, który pozwoli symulować pogodę producentowi pieczarek. Tabela z dystrybucją zamieszczona została ponownie poniżej (por. tab. 7.3). Zauważmy, że dodana została jedna kolumna *Zakres liczb losowych*, która pokrywa wszystkie wartości z przedziału (0, 1), ale jednocześnie przydziela je poszczególnym wartościom zmiennej losowej tak, aby odpowiadało to dokładnie *szansie* wystąpienia danego typu dnia. Jeżeli *dni bardzo ciepłe* stanowią 9% wszystkich dni, to przydzielamy tej wartości zmiennej losowej 9% zakresu liczb lo-

sowych (czyli liczby od 0,00 do 0,09). Skoro dni *cieple* stanowią 21% wszystkich dni, to przydzielamy im 21% zakresu liczb losowych (od 0,10 do 0,30). I tak dalej.

Tabela 7.3. Przydział zakresu liczb losowych dla przykładu *Pieczarki*

Wartość zmiennej losowej	Numeryczna wartość zmiennej losowej	Liczba wystąpień	Prawdopodobieństwo $P()$	$P()$ skumulowane	Zakres liczb losowych
Dni bardzo ciepłe	1	32	0,09	0,09	(0,00 do 0,09]
Dni ciepłe	2	78	0,21	0,30	(0,09 do 0,30]
Dni chłodne	3	158	0,43	0,73	(0,30 do 0,73]
Dni zimne	4	97	0,27	1,00	(0,73 do 1,00)

W modelu symulacyjnym będziemy generowali liczby losowe dla kolejnych *symulowanych* dni, następnie będziemy sprawdzali, do którego z czterech przedziałów z tab. 7.3 wygenerowana liczba należy i w ten sposób uzyskamy informację o symulowanej pogodzie. Na przykład, jeżeli w pierwszym dniu symulacji generator liczb losowych (czyli funkcja $LOS()$) zwróci nam wartość 0,2456, to będzie to oznaczało, że tego dnia będzie ciepła pogoda, a zmienna losowa przyjmie wartość 2. Jeżeli w drugim dniu $LOS() = 0,8765$, to wskaże nam to na dzień zimny (zmienna losowa przyjmie wartość 4).

Uwaga 1: Kolejno symulowane przez nas dni nie wskazują, jaka pogoda w przyszłym roku będzie 1 stycznia, 2 stycznia, 3 stycznia,... itd. Prowadząc symulację, ustalimy jedynie ilu dni zimnych, chłodnych, ciepłych itd. możemy spodziewać się w przyszłym roku.

Uwaga 2: Wielkość empirycznej próby, na podstawie której zbudowaliśmy rozkład dyskretny opisujący zeszłoroczną pogodę (tu: 365 dni), w żaden sposób nie wymusza na nas konieczności symulacji kolejnych dokładnie 365 dni. Prowadzona przez nas symulacja może objąć 10 dni (ale wtedy statystycznie byłaby niewiele warta), 100 dni, 1000 dni czy dowolną inną ich liczbę. Im więcej dni uwzględnimy w symulacyjnym eksperymencie, tym bardziej wiarygodne statystycznie wyniki uzyskamy.

Rozkłady teoretyczne – ciągle

W rozdziale 3.6.1 omówione zostały podstawy *metody transformacji odwrotnej*, za pomocą której możemy generować w modelu symulacyjnym wartości zmiennej losowej o określonej postaci funkcyjnej. Przypomnijmy, że pozyskiwanie wartości zmiennej losowej typu ciągłego polega na odczytywaniu wartości odwrotnej funkcji dystrybuanty dla losowych wartości argumentu. Najpierw określana jest liczba losowa z przedziału (0, 1), czyli wartość $F(x)$, a następnie za pomocą funkcji odwrotnej wyznaczana jest wartość x . Wykaz wyprowadzonych za pomocą metody transformacji odwrotnej formuł dla podstawowych ciągłych rozkładów losowych zaprezentowano w tabeli 3.2 w podrozdziale 3.6.2. W podrozdziale tym wymieniono również niektóre funkcje dostępne w zestawie formuł arkusza kalkulacyjnego Excel, które można wykorzystać do generowania wartości ciągłych zmiennych losowych.

7.5.3. WYKONANIE DUŻEJ LICZBY POWTÓRZEŃ

Abyśmy mogli wiarygodnie opisać zachowanie modelu, musimy wykonać dużą liczbę powtórzeń, np. 100, 1000 lub więcej.

	I	J	K	L	M	N	O
1	Metoda Monte Carlo - uprawa pieczarek						
2							
3	Liczba wystąpienia	Prawdopodobieństwo P()	Lewy zakres	P() skumulowane	Zakres liczb losowych	Wartość zmiennej losowej	Numeryczna wartość zm. losowej
4				0,00			0,00
5	32	0,09	0,00	0,09	0,00 do 0,09	Bardzo ciepło	1
6	78	0,21	0,09	0,30	0,09 do 0,30	Ciepło	2
7	158	0,43	0,30	0,73	0,30 do 0,73	Chłodno	3
8	97	0,27	0,73	1,00	0,73 do 1,00	Zimno	4
9							
10	365	1					

Rys. 7.6. Fragment arkusza z przygotowanymi dyskretnymi wartościami zmiennej losowej.

Tablica z danymi

R	S	T
Metoda Monte Carlo - uprawa pieczarek		
Kolejny dzień	Liczba losowa	Symulowana pogoda
		=LOS()
1	0,28	Ciepło
2	0,15	Ciepło
3	0,83	Zimno
4	0,94	Zimno
5	0,12	Ciepło
6	0,81	Zimno
7	0,28	Ciepło
8	0,76	Zimno
9	0,83	Zimno
10	0,33	Chłodno
11	0,95	Zimno
12	0,62	Chłodno
13	0,41	Chłodno
14	0,24	Ciepło
15	0,18	Ciepło
16	0,74	Zimno
17	0,42	Chłodno
18	0,31	Chłodno
19	0,84	Zimno
20	0,28	Ciepło
21	0,09	Ciepło

=WYSZUKAJ.PIONOWO(S5;\$K\$5:\$O\$8;4;PRAWDA)

Rys. 7.7. Wykonanie dużej liczby powtórzeń. Przykład *Pieczarki*

Funkcja *Wyszukaj.Pionowo()* (z rys. 7.7, kolumna T) szuka wskazanej wartości, czyli liczby losowej (rys. 7.7, kolumna S) w skrajnej lewej kolumnie tablicy z danymi – por. rys. 7.6 (tablica z danymi podana jest jako drugi argument funkcji *Wyszukaj.Pionowo()*). Funkcja wyszukuje wzdłuż lewej skrajnej kolumny tej tablicy, aż

znajdzie liczbę większą od liczby losowej. Wtedy wraca do wiersza poprzedniego i zwraca wartość z tego wiersza, ale znajdującą się w kolumnie wskazanej jako trzeci argument funkcji. W przykładzie zwracane są wartości z czwartej kolumny tablicy (czyli wartości tekstowe typu „zimno”, „chłodno” itd.), licząc od pierwszej kolumny tablicy podanej w drugim argumentem funkcji. Ostatni argument funkcji precyzuje, czy poszukiwana jest dokładnie wskazana wartość.

Po zakończeniu eksperymentu, czyli po wykonaniu zaplanowanej liczby powtórzeń, standardowo wyznacza się *wartości średnie*, *odchylenie standardowe*, *długość przedziału ufności* i inne mierniki statystyczne.

7.6. WYKONANIE PEŁNEGO BADANIA SYMULACYJNEGO

Jeżeli model zawiera zmienne losowe (a tak jest w przypadku symulacji Monte Carlo), jedno powtórzenie jest niewystarczające do przeprowadzenia prawidłowych oszacowań (to tak, jakby rzucić raz monetą i stwierdzić, że wszystkie rzuty monetą zawsze kończą się wyrzuceniem orła czy reszki). Jest rzeczą oczywistą, że powinniśmy wykonać tych powtórzeń wiele, a następnie (traktując uzyskane wyniki jako niezależne) wyznaczyć wartości średnie oraz prawdopodobieństwo. Powiedzmy, że wykonaliśmy 100 powtórzeń rzutu monetą i orzeł pojawił się w 60% przypadków. Następnie wykonaliśmy jeszcze raz 100 powtórzeń i tym razem orzeł pojawił się w 65% przypadków. Który wynik należy przyjąć? Aby uzyskać statystycznie wiarygodną próbę możemy:

(a) zwiększyć liczbę powtórzeń do, powiedzmy, 1000,

(b) wykonać wiele razy po 100 powtórzeń, każdorazowo obliczając średnie i prawdopodobieństwa, a następnie potraktować uzyskane wyniki jako próbę losową i powtórnie wyznaczyć wartość średnią (czyli obliczyć średnią ze średnich),

(c) możemy połączyć (a) i (b), czyli zwiększyć liczbę powtórzeń w jednym eksperymencie oraz zwiększyć liczbę eksperymentów. Takie postępowanie zapewni nam uzyskanie najbardziej precyzyjnych estymacji wyników.

7.7. ANALIZA WYNIKÓW

7.7.1. OCENA PRECYZJI OSZACOWANIA

Do oceny precyzji oszacowań symulacyjnych najlepiej jest wykorzystać *przedział ufności* oraz *przedział predykcji*.

Przedział ufności

Przedział ufności to przedział liczbowy, o którym przypuszczamy, że mieści się w nim nieznaną wartość parametru populacji. Wzory na wyznaczenie przedziału ufności zaprezentowano w podrozdziale 6.6. W odniesieniu do prezentowanego modelu symulacji pogody, mogliśmy powiedzieć, że gdybyśmy powtarzali eksperyment symulacyjny nieskończenie wiele razy (wykonując każdorazowo np. po 100 powtórzeń) i obliczali za każdym razem przedział ufności dla założonej wcześniej wartości alfa (np. $\alpha = 0,1$), to 90% obliczonych przedziałów ufności zawierałoby prawdziwą (lecz nieznaną) *wartość przeciętną liczby ciepłych dni*. Obliczając przedział ufności tylko raz, możemy być pewni na 90% (z $p = 0,9$), że wyznaczony przez nas przedział pokryje prawdziwą (ale nieznaną) wartość średnią.

Przedział ufności to przedział „losowy”. Kolejny eksperyment złożony ze 100 powtórzeń doprowadzi nas do nieco innego przedziału. Im więcej powtórzeń, tym przedział ufności staje się coraz węższy, tak aby po nieskończeniu wielu eksperymentach skurczyć się do punktu – szukanej wartości średniej (estymacja punktowa). Stąd proste wnioski, aby poprawić precyzję oszacowań, należy zwiększać liczbę powtórzeń.

Przedział ufności nie jest natomiast przedziałem, w którym mieści się 90% wyników powtórzeń symulacyjnych. Taki przedział również możemy skonstruować i nosi on nazwę przedziału predykcji.

Przedział predykcji

Przy każdym powtórzonym eksperymencie (np. wyznaczeniu wielkości sprzedaży) mamy 90% szans, że popyt zawarty będzie w wyznaczonym przedziale. Jeżeli powtarzalibyśmy eksperyment wiele razy, to około 90% powtórzeń wskaże nam sprzedaż z tego właśnie przedziału. Przedział predykcji nie będzie się kurczył do punktu w miarę zwiększania się liczby powtórzeń, ponieważ sprzedaż będzie różniła się w każdym badanym okresie i w zbudowanym przedziale predykcji musimy uwzględnić wystąpienie wariancji.

Podsumowując: przedział ufności dotyczy oszacowania wartości średniej badanej zmiennej wyjściowej (inaczej – jest to obserwacja zachowania typowego), natomiast przedział predykcji odnosi się do obserwowanych wartości zmiennych losowych (inaczej – jest to obserwacja pojedynczych zachowań zmiennej).

7.7.2. SZACOWANIE WARTOŚCI ŚREDNIEJ I OBLICZANIE PRAWDOPODOBIENSTWA

Powróćmy do przykładu wprowadzającego z początku rozdziału. Zbudujemy model symulacyjny metodą Monte Carlo i poddamy analizie uzyskane wyniki. Założmy, że *wielkość sprzedaży* zmienia się zgodnie z rozkładem jednostajnym o parametrach $a = 1000$, $b = 2000$; *cena* zmienia się zgodnie z rozkładem trójkątnym symetrycznym o parametrach $a = 21$, $c = 23$, $b = 25$; *koszt jednostkowy* zmienia się również zgodnie z rozkładem trójkątnym symetrycznym o parametrach $a = 15$, $c = 17$, $b = 19$. Dane liczbowe do modelu przedstawiono na rys. 7.8.

	A	B	C	D	E
1	Metoda MC - szacowanie zysku ze sprzedaży				
2					
3	Scenariusz	Parametr a	Parametr c	Parametr b	
4					
5	Wielkość sprzedaży	1000,00	1500,00	2000,00	
6	Cena jednostkowa	21,00	23,00	25,00	
7	Koszt jednostkowy	15,00	17,00	19,00	
8					
9	Koszty stałe	5000,00			
10					

Rys. 7.8. Dane do modelu szacowania zysku ze sprzedaży

Podstawowy model symulacyjny obejmujący 100 powtórzeń przedstawiono na rys. 7.9 (na rysunku widoczne są wyniki pierwszych siedmiu powtórzeń). W modelu generowanych jest pięć liczb losowych (kolumny I, K, L, N, O). Pierwsza liczba losowa wykorzystywana jest w formule rozkładu jednostajnego do generowania wielkości sprzedaży, liczby losowe nr 2 i nr 3 występują w formule generującej cenę z rozkładu trójkątnego symetrycznego, a liczby losowe nr 4 i nr 5 są konieczne do generowania kosztu jednostkowego również za pomocą rozkładu trójkątnego symetrycznego.

	G	H	I	J	K	L	M	N	O	P	Q
1											
2						p=	0,5		p=	0,5	
3		Powtórzenie	Liczba Losowa 1	Sprzedaż	Liczba losowa 2	Liczba losowa 3	Cena	Liczba losowa 4	Liczba losowa 5	Koszt	Zysk
4		1	0,25373093	1254	0,10281364	0,94405735	23	0,79855033	0,77407319	17	2149,87814
5		2	0,51808954	1518	0,36865633	0,78511781	23	0,85335131	0,97454462	17	3723,72886
6		3	0,21405433	1214	0,43890667	0,71376796	23	0,12641981	0,78026752	17	2190,89325
7		4	0,21860241	1219	0,10561271	0,43378097	22	0,374271	0,92678791	17	1570,51474
8		5	0,19543215	1195	0,55048367	0,5114101	24	0,76996001	0,98860717	17	2840,02141
9		6	0,32232455	1322	0,76074568	0,49466782	24	0,34397749	0,4142805	16	4660,97589
10		7	0,86009467	1860	0,20109481	0,16341076	22	0,06843901	0,35081359	16	5460,97074

Rys. 7.9. Model symulacyjny szacowania zysku ze sprzedaży

W kolumnach J, M i P wykorzystano formuły z tabeli 3.2. I tak:

$$\text{Kolumna J: } J4 = \text{\$C\$5} + (\text{\$E\$5} - \text{\$C\$5}) \cdot I4$$

$$\text{Kolumna M: } M4 = \text{\$C\$6} + ((\text{\$E\$6} - \text{\$C\$6})/2) \cdot (K4 + L4)$$

$$\text{Kolumna P: } P4 = \text{\$C\$7} + ((\text{\$E\$7} - \text{\$C\$7})/2) \cdot (N4 + O4)$$

W kolumnie Q naliczany jest zysk, według formuły 7.1:

$$Q4 = J4 \cdot (M4 - P4) - \text{\$C\$9}$$

Wyniki z przeprowadzenia 100 eksperymentów, każdy po 100 powtórzeń (por. tab. 7.4) pozwalają stwierdzić, że wartość oczekiwana zysku z prawdopodobieństwem 0,95 będzie zawierała się w przedziale (3991,26; 4081,14).

W obliczeniach wykorzystano funkcję Excela $UFNOŚĆ(Alfa, Odchylenie standardowe, Wielkość próby)$ przyjmując poziom ufności: $(1 - \alpha)100\% = 95\%$, $\alpha = 0,05$. Funkcja $UFNOŚĆ()$ wyznacza długość połowy przedziału ufności, którą należy odmierzyć po każdej stronie średniej z próby (por. dwie ostatnie komórki tab.7.4). Uzyskana wartość pokrywa się z przewidywaniami scenariusza neutralnego z rys. 7.1, co było łatwe do przewidzenia ze względu na charakter wybranych rozkładów (rozkłady symetryczne). Symulacja pozwala jednak znaleźć odpowiedź na wiele innych pytań:

Jakie jest prawdopodobieństwo, że zanotujemy zysk?

Jakie jest prawdopodobieństwo, że zanotujemy stratę?

Jakie jest prawdopodobieństwo, że uzyskany zysk będzie większy niż 5000 zł?

Jakie jest prawdopodobieństwo, że uzyskany zysk będzie większy niż 15 000 zł?

Tabela 7.4. Wyniki badań symulacyjnych, $\alpha = 0,05$

Średni zysk	4 036,20 zł
Liczba obserwacji	100
Odchylenie standardowe	229,26
Długość połowy przedziału ufności	44,94
Dolna granica przedziału ufności	3991,26
Górna granica przedziału ufności	4081,14

Aby odpowiedzieć na te pytania, *nie szacujemy średniego zysku*, ale dla każdego eksperymentu symulacyjnego zliczamy liczbę przypadków, w których wielkość zysku osiągnęła wartość większą niż 0 zł (i odpowiednio: była < 0 zł, > 5000 zł, $> 15 000$ zł) – por. tab. 7.5. Wykonujemy 100 powtórzeń, obliczając wartość średnią i długości przedziałów ufności. Obliczone wartości przedstawiono w tabeli 7.5.

Wyniki wskazują, że nasze przedsięwzięcie powinno zakończyć się dla nas pomyślnie: mamy od 96% do 97% szans, że **nie poniesiemy** straty, natomiast od 30% do 34% szans, że odnotujemy spory zysk (powyżej 5000 zł). Co prawda scenariusz optymistyczny nie ma żadnych szans powodzenia – jak wykazuje symulacja, prawdopodobieństwo wysokiego zysku (powyżej 15 000 zł) jest zerowe, ale też „czarny scenariusz” raczej nam nie grozi – prawdopodobieństwo straty waha się od 0,03 do 0,04 czyli jest praktycznie zerowe.

Tabela 7.5. Wyniki badań symulacyjnych – szacowanie prawdopodobieństw dla zysku, $\alpha = 0,05$

		$P(X > 5000)$	$P(X > 0)$	$P(X < 0)$	$P(X > 15 000)$
Liczba obserwacji	100	100	100	100	100
Średnia z próby	4036,2	0,32	0,96	0,04	0,00
Odchylenie standardowe	229,26	0,06	0,03	0,03	0,00
Długość połowy przedziału ufności	44,94	0,02	0,01	0,01	
Dolna granica przedziału ufności	3991,26	0,30	0,96	0,03	
Górna granica przedziału ufności	4081,16	0,34	0,97	0,04	

Więcej informacji na temat częstości wystąpienia określonych wartości zysku zawiera tzw. *tabela częstości* (por. tab. 7.6). Formuła *CZĘSTOŚĆ()* w Excelu służy do wyznaczania rozkładu częstości elementów zbioru danych. Wynik podawany jest w postaci tabeli pionowej. Funkcja jest wprowadzona jako tzw. *wzór tablicowy*. Najpierw przygotowujemy opis tabeli częstości: w kolumnie podajemy prawe granice przedziałów definiujące zakresy, dla których będziemy liczyli przypadki wystąpienia poszczególnych wartości zysku. Następnie zaznaczamy obszar, do którego wpisane będą wyznaczone częstości (w tab. 7.6 tym obszarem jest kolumna Częstość). Wprowadzanie funkcji *CZĘSTOŚĆ()* kończymy naciśnięciem CTRL+SHIFT+ENTER.

Wyniki pokazują, że dość wysokie prawdopodobieństwo (duża liczba, bo 80 przypadków) charakteryzuje wielkości zysku od 1000 zł do 6000 zł. Rozkład częstości wskazuje ponadto, że powinniśmy się spodziewać zysku, aczkolwiek nie powinniśmy liczyć na ogromny sukces. Przewaga przypadków (56 wystąpień) jest po lewej stronie wartości średniej – czyli od 0 zł do 4000 zł, podczas gdy prawa strona odznacza się mniejszą liczbą zanotowanych wartości (44 wystąpienia).

Tabela 7.6. Tablica częstości

Wartości zysku w zł		Częstość
-50000	0	1
0	1000	2
1000	2000	11
2000	3000	19
3000	4000	23
4000	5000	15
5000	6000	12
6000	7000	5
8000	9000	7
9000	10000	2
>10000		3

7.7.3. BADANIE TYPU *WHAT-IF*

Jedną z istotnych zalet symulacji jest możliwość przeprowadzenia tzw. badań *what-if*, czyli *co-się-stanie-jeśli?* Możemy dowolnie zmieniać wybrany parametr i obserwować wpływ tych zmian na zachowanie modelu. Załóżmy, że zastanawiamy się, jaki skutek odniosłaby obniżka kosztów jednostkowych? Za pomocą symulacji możemy przetestować rozmaite scenariusze, jak w tab. 7.7: w scenariuszu 1 i 2 zmniejszamy parametry rozkładu trójkątnego odpowiednio o 1 oraz o 2 jednostki,

natomiast w scenariuszu 3 i 4 zwiększamy wartości parametrów odpowiednio o 1 oraz o 2 jednostki.

Tabela 7.7. Scenariusze badań typu *What-if*

Scenariusz nr:	Opis scenariusza	Wartość parametrów
1	Obniżenie kosztów jednostkowych	Rozkład trójkątny (14,16,18) Rozkład przesunięty w lewo o 1
2	Obniżenie kosztów jednostkowych	Rozkład trójkątny (13,15,17) Rozkład przesunięty w lewo o 2
3	Zwiększenie kosztów jednostkowych	Rozkład trójkątny (16,18,20) Rozkład przesunięty w prawo o 1
4	Zwiększenie kosztów jednostkowych	Rozkład trójkątny (17,19,21) Rozkład przesunięty w prawo o 2

Wyniki symulacji pokazują, że zysk jest wrażliwy na wartość kosztu jednostkowego. W pierwszym scenariuszu (por. tab. 7.8) zmniejszenie parametrów rozkładu kosztów o 1 jednostkę powoduje wzrost zysku o około 35%, z 4036,20 zł do 5425,90 zł. Dla scenariusza drugiego ten wzrost wynosi 50% w stosunku do scenariusza bazowego, z 4036,20 zł do 7060,47 zł. Z kolei zwiększenie parametrów o 2 jednostki powoduje spadek zysku o 75%, z 4036,20 zł do 2551,00 zł.

Podobne badania możemy przeprowadzić również dla pozostałych parametrów modelu, tj. ceny oraz wielkości sprzedaży (por. tab. 7.9 i tab. 7.10).

Okazuje się, że podobną wrażliwość model wykazuje na zmiany ceny (tab. 7.9). Natomiast porównywalny wzrost zysku, który chcielibyśmy uzyskać poprzez wzrost wielkości sprzedaży, wymagałby zwiększenia parametrów rozkładu popytu o 200 jednostek (tab. 7.10).

Tabela 7.8. Badania *What-if*. Wyniki przy zmianie parametrów rozkładu kosztu jednostkowego, $\alpha = 0,05$

Zmiana parametrów rozkładu kosztu jednostkowego				
Scenariusz →	Nr 1	Nr 2	Nr 3	Nr 4
Parametry rozkładu				
<i>a</i>	14	13	16	17
<i>b</i>	18	17	20	21
<i>c</i>	16	15	18	19
	Zysk	Zysk	Zysk	Zysk
Liczba obserwacji	100	100	100	100
Średnia z próby	5425,90	7060,47	2551,00	1016,45
Odchylenie standardowe	374,55	389,11	335,62	333,86
Długość połowy przedziału ufności	103,82	107,85	93,03	92,54
Dolna granica przedziału ufności	5322,08	6952,62	2457,98	923,91
Górna granica przedziału ufności	5529,72	7168,32	2644,03	1108,99

Tabela 7.9. Badania *What-if*. Wyniki przy zmianie parametrów rozkładu ceny jednostkowej, $\alpha = 0,05$

Zmiana parametrów rozkładu ceny jednostkowej				
Scenariusz →	Nr 1	Nr 2	Nr 3	Nr 4
Parametry rozkładu				
<i>a</i>	22	23	20	19
<i>b</i>	26	27	24	23
<i>c</i>	24	25	22	21
	Zysk	Zysk	Zysk	Zysk
Liczba obserwacji	100	100	100	100
Średnia z próby	5488,57	6957,09	2475,83	991,02
Odchylenie standardowe	396,04	369,35	307,05	304,53
Długość połowy przedziału ufności	109,78	102,38	85,11	84,41
Dolna granica przedziału ufności	5378,79	6854,72	2390,72	906,61
Górna granica przedziału ufności	5598,35	7059,47	2560,94	1075,43

Tabela 7.10. Badania *What-if*. Wyniki przy zmianie parametrów rozkładu wielkości sprzedaży, $\alpha = 0,05$

Zmiana wielkości sprzedaży				
Scenariusz →	Nr 1	Nr 2	Nr 3	Nr 4
Parametry rozkładu				
<i>a</i>	900	800	1100	1200
<i>b</i>	1900	1800	2100	2200
	Zysk	Zysk	Zysk	Zysk
Liczba obserwacji	100	100	100	100
Średnia z próby	3424,94	2776,47	4600,95	5237,79
Odchylenie standardowe	385,08	346,01	341,12	375,27
Długość połowy przedziału ufności	106,74	95,91	94,55	104,02
Dolna granica przedziału ufności	3318,20	2680,56	4506,40	5133,77
Górna granica przedziału ufności	3531,67	2872,37	4695,51	5341,81

7.8. PRZYKŁAD MODELU SYMULACYJNEGO MONTE CARLO

Problem

Nieduży sklep próbuje ustalić, ile powinien zamawiać sztuk miesięcznika *Pies*. Analiza sprzedaży zeszłorocznej wskazuje, że popyt na czasopismo odpowiada pewnej zmiennej losowej dyskretnej (por. tab. 7.11). Sklep płaci wydawnictwu 1 zł za

każdy egzemplarz *Psa* i sprzedaje go za 1,55 zł. Każdy niesprzedany egzemplarz może być zwrócony za 0,5 zł. Ile egzemplarzy *Psa* sklep powinien zamawiać? Dodatkowo, w najbliższym czasie w mieście ma się odbyć duża wystawa psów. Sklepiarz spodziewa się, że popyt będzie większy niż ten szacowany na podstawie danych historycznych. Sklepiarz nie jest przesadnym optymistą, prognozuje jednak wzrost popytu na 10–20%. Czy wielkość zamówienia powinna się zmienić?

Tabela 7.11. Zeszłoroczna sprzedaż miesięcznika *Pies*

Popyt [szt]	Liczba miesięcy, w których sprzedano daną liczbę sztuk czasopisma
15	1
20	4
25	4
30	2
35	1

W badaniu symulacyjnym będziemy symulować wybrane wielkości zamówienia z przedziału [15, 35] sztuk wiele razy (np. przeprowadzimy 250 powtórzeń). Następnie powtórzymy po 100 razy każdy eksperyment z 250 powtórzeniami i zbadamy, przy której wielkości zamówienia sklepiarz osiąga najwyższy średni zysk.

Zapis danych do modelu

Budowę modelu rozpoczynamy od wprowadzenia do arkusza kalkulacyjnego wszystkich danych z zadania (por. rys. 7.10). Następnie, w komórkach F3 do I7 budujemy przedziały losowe, według których będą generowane w modelu wartości popytu odpowiadające zeszłorocznej sprzedaży:

Kolumna F to prawdopodobieństwo, że zmienna losowa osiągnie wskazaną wartość popytu, np. $F3 = D3/SE12$. W formule, przy komórce E12 wprowadzamy adresowanie bezwzględne (znaki \$).

Kolumna G to lewy zakres liczb losowych. Rozpoczynamy od wartości zero (G3), a w kolejnych komórkach przepisujemy wartości z komórek położonych wiersz wyżej w kolumnie H, np. $G4 = H3$.

Kolumna H to prawy zakres liczb losowych (czyli prawdopodobieństwo skumulowane). Do komórki H3 przepisujemy wartość z F3, a w kolejnych komórkach do poprzedniej wartości prawdopodobieństwa skumulowanego dodajemy aktualną wartość prawdopodobieństwa, czyli np. $H4 = H3 + F4$. Ostatnia wartość w kolumnie H musi oczywiście wynosić 1.

Do kolumny I kopiujemy wartości zmiennej losowej (czyli popytu) z kolumny B.

W komórce D9 wprowadzamy próbną wartość wielkości zamówienia (np. 25 sztuk), a w komórce D10 obliczamy koszt związany z zamówieniem, $D10 = D9 \cdot E14$.

Komórki E14:E16 zawierają pozostałe dane.

	A	B	C	D	E	F	G	H	I
1									
2		Popyt		Liczba miesięcy, w których sprzedano taką liczbę sztuk czasopisma		p()	Lewy zakres	p0 skum	Popyt
3		15		1		0,083	0,000	0,083	15
4		20		4		0,333	0,083	0,417	20
5		25		4		0,333	0,417	0,750	25
6		30		2		0,167	0,750	0,917	30
7		35		1		0,083	0,917	1,000	35
8						1,0			
9		Zamówienie		25					
10		Koszt Zakupu		25 zł					
11									
12		Razem miesiący		12					
13									
14		Cena zakupu 1 egzemplarza		1,00 zł					
15		Cena sprzedaży 1 egzemplarza		1,55 zł					
16		Cena zwrotu 1 egzemplarza		0,50 zł					

Rys. 7.10. Dane do zadania „Sprzedaż miesięcznika *Pies*”

Budowa modelu podstawowego

Po wprowadzeniu danych przystępujemy do budowy modelu logicznego. Model podstawowy będzie zdefiniowany w jednym wierszu (jeden wiersz odpowiada jednemu powtórzeniu), którego formuły będą następnie skopiowane do kolejnych wierszy. Główny model symulacyjny przedstawiony jest na rys. 7.11. Pokazano pierwszych 10 powtórzeń z przeprowadzonych 250.

Komórka B19: numer kolejnego powtórzenia (będzie ich 250).

Komórka C19: liczba losowa generowana za pomocą bezparametrowej formuły $C19 = LOS()$.

Komórka D19: prognozowany popyt. Ta wartość symulowana jest za pomocą formuły $D19 = WYSZUKAJ.PIONOWO(C19; \$G\$3: \$I\$7; 3)$.

Komórka E19: liczba sprzedanych egzemplarzy czasopisma jest zawsze mniejszą z dwóch wartości, tj. wielkości zamówienia (D9) i wielkości popytu (D19). $E19 = MIN(\$D\$9, D19)$.

Komórka F19: przychód ze sprzedaży wyliczany jest jako iloczyn wielkości sprzedaży i ceny sprzedaży. $F19 = \$E\$15 \cdot E19$.

Komórka G19: wielkość niezrealizowanego popytu ma wartość większą od zera, gdy zamówiliśmy mniej egzemplarzy czasopisma niż wynosił popyt (por. powtórzenie nr 1, wiersz 19) i w związku z tym nie zaspokoiliśmy całego popytu. W przeciwnym przypadku w komórce wpisujemy zero. Będzie to zatem różnica pomiędzy popytem i sprzedażą. $G19 = D19 - E19$.

Komórka H19: liczba zwracanych sztuk ma wartość większą od zera, gdy zamówiliśmy więcej egzemplarzy czasopisma niż wynosił popyt, w przeciwnym razie w komórce wpisujemy zero. Będzie to zatem różnica pomiędzy zamówieniem i sprzedażą. $H19 = \$D\$9 - E19$.

dy eksperyment złożony będzie z 250 powtórzeń). Następnie w komórkach M18:U18 wprowadzamy możliwe wielkości zamówienia. Ze względu na to, że zmienna losowa popytu przyjmowała wartość maksymalną przy popycie 20 i 25 sztuk, w przedziale od 20 do 25 egzemplarzy decydujemy się na dokładniejsze badanie. Ponieważ w tabeli chcemy wyznaczyć zysk przy różnej wielkości zamówienia, odwołujemy się do formuły obliczającej średni zysk (komórka J270 na rys. 7.12) w komórce stanowiącej lewy górny róg naszej tabeli (L18 = J270).

	K	L	M	N	O	P	Q	R	S	T	U
17											
18		10,92 zł	15	20	21	22	23	24	25	30	35
19		1									
20		2									
21		3									
22		4									
23		5									

Rys. 7.13. Fragment niewypełnionej tabeli danych z dwiema zmiennymi. Na rysunku pokazano 5 spośród 100 wykonanych eksperymentów. Cała tabela obejmuje wiersze od 18 do 118

	K	L	M	N	O	P	Q	R	S	T	U
17											
18		10,92 zł	15	20	21	22	23	24	25	30	35
19		1	8,25	10,643	10,6092	10,6972	10,8314	11,4654	11,146	9,759	7,469
20		2	8,25	10,559	10,5714	10,8358	10,7852	10,7808	10,999	9,99	7,364
21		3	8,25	10,475	10,7982	10,6888	10,9448	11,2722	11,293	9,906	8,036
22		4	8,25	10,58	10,7058	10,7896	10,9616	10,7514	10,684	9,759	7,889
116		98	8,25	10,601	10,6638	10,7476	10,9952	11,0118	11,272	9,948	8,624
117		99	8,25	10,622	10,668	10,8022	10,8692	11,0328	11,419	10,41	8,288
118		100	8,25	10,601	10,8906	10,7224	10,7096	11,289	11,44	10,221	8,519
119											
120			15	20	21	22	23	24	25	30	35
121		Średnia	8,25	10,552175	10,67908	10,80068	10,9018	11,00277	11,09481	9,896025	7,850413
122		Obserwacje	100	100	100	100	100	100	100	100	100
123		Odchylenie	0	0,0773414	0,116127	0,126561	0,151913	0,206896	0,226789	0,299437	0,326366
124		Przedział Ufności		0,0151586	0,022761	0,024805	0,029774	0,040551	0,04445	0,058688	0,063967
125		Dół		10,537016	10,65632	10,77587	10,87203	10,96222	11,05036	9,837337	7,786446
126		Góra		10,629516	10,7952	10,92724	11,05372	11,20967	11,3216	10,19546	8,176779
127											
128											
129		Ryzyko[%]		-65,89724	-48,7951	-44,1945	-33,0156	-8,77181	0	32,03297	43,90721
130		Zmiana Zysku [%]		-4,890912	-3,74711	-2,6511	-1,73964	-0,8296	0	-10,8049	-29,2425

Rys. 7.14. Tabela z wynikami dla różnej wysokości zamówienia.

W tabeli z eksperymentami ukryto wiersze od 23 do 115

Zaznaczamy zakres tabeli (L18:U118) i wybieramy z menu *Dane > Tabela*. Następnie wskazujemy komórkę D9 jako wierszową komórkę wejściową (jest to komórka, w której zdefiniowaliśmy wielkość zamówienia (por. rys. 7.10) i dowolną pustą komórkę jako kolumnową komórkę wejściową (np. L17). Po kliknięciu na OK, Excel wypełnia każdą pustą komórkę tabeli, wykonując każdorazowo 250 powtórzeń i obli-

czając wartość średnią (por. rys. 7.14). I tak na przykład, aby wypełnić kolumnę M w tabeli, Excel podstawia pod komórkę D9 z rys. 7.10 wartość 15 sztuk i rozpoczyna symulację, która powtarzana jest 250 razy dla każdej komórki kolumny M. Analogicznie, aby wypełnić kolumnę N, Excel podstawia pod komórkę D9 wartość 20 i ponownie rozpoczyna symulację powtarzając ją 250 razy dla każdej komórki kolumny N. Proces jest kontynuowany aż do wypełnienia wszystkich kolumn w tabeli. Technicznie rzecz biorąc, proces wypełniania komórek w arkuszu odbywa się wzdłuż wierszy, ale istota wypełniania tabeli jest dokładnie taka, jak opisano.

Każde naciśnięcie klawisza F9 powoduje uruchomienia symulacji i ponowne obliczenie wartości wszystkich komórek.

Wyznaczone średnie wartości ze 100 eksperymentów (wiersz 121 na rys. 7.14) przy różnej liczbie zamawianych egzemplarzy sugerują wybór zamówienia w wysokości 25 egzemplarzy czasopisma (wyróżniona komórka S121 na rys. 7.14). Przedział ufności obliczony dla poziomu ufności $(1 - \alpha)100\% = 95\%$ informuje nas o tym, że mamy 95% pewności, że średni zysk przy zamówieniu 25 sztuk będzie zawierał się w przedziale od 11,05 do 11,32 zł.

Wpływ ryzyka na decyzję

Zamówienie 23 egzemplarzy czasopisma (kolumna Q) zamiast 25 spowoduje spadek spodziewanego zysku o niecałe 2%, ale za to ryzyko (mierzone odchyleniem standardowym) uzyskania innej wartości zysku niż wartość oczekiwana spada o prawie 20% w porównaniu z analogicznym ryzykiem przy zakupie 25 egzemplarzy. Zwiększenie zamówienia do 30 sztuk spowoduje wzrost ryzyka o ponad 50% przy jednoczesnym spadku spodziewanego zysku o prawie 10%. Tak więc, jeżeli sklepikarz nie lubi ryzyka, powinien zdecydować się na zamówienie mniejsze niż 25 sztuk. Zauważmy również, że zamówienie 15 sztuk nie wiąże się z żadnym ryzykiem (odchylenie standardowe równa się zeru), ponieważ w takim przypadku zawsze wszystkie zamówione egzemplarze zostaną sprzedane. Formuły w wierszach 129 i 130 na rys. 7.14 są następujące:

$$Q129 = (Q123 - \$\$123)/\$\$123\%$$

$$Q130 = (Q121 - \$\$121)/\$\$121\%$$

Modyfikacja zadania – wystawa psów

Zbudowany model możemy wykorzystać do badań typu *co-się-stanie-jeśli?* Chcemy sprawdzić, jaki wpływ na wielkość zamówienia może mieć wzrost popytu w granicach między 10% a 20%. Nie wiemy, czy będzie to dokładnie 10%, czy 12% czy 18%. Wprowadzimy zatem do modelu drugi rozkład losowy, tym razem będzie to rozkład ciągły (jednostajny), za pomocą którego będziemy korygowali wielkość popytu o losowo wygenerowaną wartość z przedziału od 10% do 20%. Zmodyfikowany arkusz przedstawiono na rys. 7.15.

W zmodyfikowanej wersji modelu wprowadziliśmy parametry rozkładu jednostajnego (D19:E19), które wykorzystamy do wygenerowania procentu wzrostu popytu. W modelu podstawowym dodajemy kolumnę „Liczba losowa 2”, w której generujemy

drugi ciąg liczb losowych za pomocą formuły $LOS()$ oraz kolumnę „Wzrost popytu”, w której za pomocą formuły dla rozkładu jednostajnego (por. tab. 3.2 w podrozdziale 3.6.2):

$$F22 = D\$19 + (\$E\$19 - \$D\$19) \cdot E22$$

symulujemy o ile procent (w tym powtórzeniu) wzrośnie popyt. Ostateczną wartość prognozowanego popytu obliczamy w kolumnie „Popyt prognozowany”:

$$G22 = D22 + (F22/100) \cdot D22.$$

Oczywiście, w kolumnie „Sprzedaż” i „Niezrealizowany popyt” musimy poprawić formuły tak, aby odwoływać się do nowej komórki „Popyt prognozowany”. Pozostała część modelu podstawowego nie ulega zmianie.

W tabeli, w której szukamy optymalnego poziomu zamówienia, musimy skorygować badane wartości zmiennej decyzyjnej tak, aby odpowiadały powiększonemu popytowi (por. rys. 7.16).

Wyniki symulacji sugerują zamówienie w wysokości 30 sztuk, chociaż niewielkie różnice w porównaniu z zamówieniem 25 sztuk wskazują na potrzebę dokładniejszego przebadania zakresu zamówienia od 25 do 30 sztuk.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2		Popyt	Liczba miesięcy, w których sprzedano taką liczbę sztuk czasopisma		p0	Lewy zakres	p0 skum	Popyt					
3		15	1		0,083	0,000	0,083	15					
4		20	4		0,333	0,083	0,417	20					
5		25	4		0,333	0,417	0,750	25					
6		30	2		0,167	0,750	0,917	30					
7		35	1		0,083	0,917	1,000	35					
8					1,0								
9		Zamówienie	25										
10		Koszt Zakupu	25 zł										
11													
12		Razem miesięcy	12										
13													
14		Cena zakupu 1 egzemplarza	1,00 zł										
15		Cena sprzedaży 1 egzemplarza	1,55 zł										
16		Cena zwrotu 1 egzemplarza	0,50 zł										
17													
18		Prognoza wzrostu [%]	a	b									
19			10	20									
20													
21		Numer powtórzenia	Liczba losowa 1	Popyt zeszlatoroczny [szt]	Liczba losowa 2	Wzrost popytu [%]	Popyt prognozowany [szt]	Sprzedaż [szt]	Przychód [zł]	Niezrealizowany popyt [szt]	Zwrot [szt]	Wartość Zwrotu [zł]	Zysk/ Strata
22		1	0,686585	25	0,1651797	11,65	27,91	25,00	38,75 zł	2,91	0,00	0,00 zł	13,75 zł
23		2	0,627878	25	0,5967024	15,97	28,99	25,00	38,75 zł	3,99	0,00	0,00 zł	13,75 zł
24		3	0,425423	25	0,2870176	12,87	28,22	25,00	38,75 zł	3,22	0,00	0,00 zł	13,75 zł
25		4	0,011123	15	0,454995	14,55	17,18	17,18	26,63 zł	0,00	7,82	3,91 zł	5,54 zł
26		5	0,71675	25	0,410546	14,11	28,53	25,00	38,75 zł	3,53	0,00	0,00 zł	13,75 zł
27		6	0,183707	20	0,3974212	13,97	22,79	22,79	35,33 zł	0,00	2,21	1,10 zł	11,43 zł
28		7	0,285187	20	0,0326294	10,33	22,07	22,07	34,20 zł	0,00	2,93	1,47 zł	10,67 zł
29		8	0,664622	25	0,4672238	14,67	28,67	25,00	38,75 zł	3,67	0,00	0,00 zł	13,75 zł
30		9	0,816111	30	0,2329961	12,33	33,70	25,00	38,75 zł	8,70	0,00	0,00 zł	13,75 zł
31		10	0,466329	25	0,0295772	10,30	27,57	25,00	38,75 zł	2,57	0,00	0,00 zł	13,75 zł

Rys. 7.15. Zmodyfikowany model: popyt wzrasta od 10% do 20%

N	O	P	Q	R	S	T	U	V	W	X
20										
21	12,43 zł	15	20	25	30	35	40	45	50	55
22	1	8,25	10,631669	12,46737	13,00049	11,55667	9,393078	6,867811	3,848404	1,38052
23	2	8,25	10,816551	12,42712	12,26182	10,75564	9,096422	6,411217	4,47763	1,557142
24	3	8,25	10,834594	12,2875	12,41681	11,45625	9,528538	6,387618	4,347821	1,439881
25	4	8,25	10,655344	12,3794	12,33399	10,98199	9,541684	7,163891	4,457381	1,906524
119	98	8,25	10,832195	12,55689	12,46073	10,94405	9,732732	6,56695	3,471588	0,882411
120	99	8,25	10,779091	12,47145	12,51106	11,56807	9,064649	6,749971	4,322246	2,253607
121	100	8,25	10,800969	12,45262	12,78593	10,94823	9,064883	6,529184	3,499285	1,216829
122										
123		15	20	25	30	35	40	45	50	55
124	Srednia	8,25	10,757371	12,38373	12,51535	11,21102	9,249396	6,761859	4,119347	1,633666
125	Obserwacje	100	100	100	100	100	100	100	100	100
126	Odchylenie	0	0,0515549	0,124447	0,242841	0,373123	0,358503	0,395867	0,395934	0,405883
127	Przedział Ufności		0,0101046	0,024391	0,047596	0,073131	0,070265	0,077588	0,077602	0,079552
128	Dół		10,747266	12,35934	12,46776	11,13789	9,179131	6,68427	4,041745	1,554114
129	Góra		10,808925	12,50818	12,75819	11,58415	9,607899	7,157726	4,515281	2,039549
130										
131										
132	Ryzyko[%]		-78,7701	-48,7537	0	53,64925	47,62857	63,0149	63,04264	67,13938
133	Zmiana Zysku [%]		-14,04661	-1,0517	0	-10,4218	-26,0956	-45,9715	-67,0857	-86,9467

Rys. 7.16. Tabela z dwiema zmiennymi do poszukiwania optymalnego zamówienia

7.9. DYNAMICZNE MODELE SYMULACYJNE MONTE CARLO

7.9.1. MODELE KOLEJKOWE W KONWENCJI SYMULACJI MONTE CARLO

Jak podano na początku rozdziału 7, terminu *metoda Monte Carlo* używa się najczęściej w odniesieniu do symulacji, w której nie uwzględnia się wpływu czasu (np. symulacja zysku po zakończeniu planowanego przedsięwzięcia). Możliwe jest jednak wykorzystanie tego podejścia do prowadzenia symulacji dynamicznej, w trakcie której obserwuje się zmiany zachodzące w systemie w kolejnych chwilach i wykorzystuje się informacje zebrane podczas wcześniej symulowanych okresów do wyznaczenia stanu systemu w okresie kolejnym. Podstawowa różnica w realizacji dynamicznej symulacji Monte Carlo w porównaniu z modelami statycznymi polega na tym, że w modelach statycznych jedno powtórzenie obejmuje symulację stanu systemu w danej chwili (np. symulacja stanu magazynu na koniec dnia), a w symulacjach dynamicznych jedno powtórzenie obejmuje obserwację działania systemu w trakcie zadanego (zwykle dłuższego) okresu – np. w ciągu miesiąca, pół roku, roku. Jeden wiersz arkusza reprezentuje historię systemu w danym momencie i dopiero zbiór wierszy (np. 180 w przypadku symulacji półrocznej) stanowi jedno powtórzenie.

W podrozdziale 4.5 opisano trzy grupy kolejkowych modeli symulacyjnych: *przeglądu działań* (ang. *activity scanning*), *zdarzeniowe* (ang. *event-driven*) i *procesowe* (ang. *process driven*). Symulacja typu *przeglądu działań* odwzorowywana jest w arkuszu kalkulacyjnym jako symulacja dyskretna o *stałym kroku czasowym*. W zaplanowanych momentach (np. w odstępie jednego dnia, jednego miesiąca itd.) badany jest stan systemu, ale przy uwzględnieniu stanu z momentu wcześniejszego. Stan zaobserwowany w chwili bezpośrednio poprzedzającej chwilę badaną oraz stan aktualny (w danej chwili) pozwalają na wyznaczenie stanu końcowego. Tego rodzaju symulacje stosowane są najczęściej w modelach produkcyjnych, sterowania zapasami, harmonogramowania i planowania projektów. W łatwy sposób można te modele zaimplementować w arkuszu kalkulacyjnym z wykorzystaniem metody Monte Carlo.

Modele kolejkowe o *zmiennym kroku czasowym* (czyli modele *zdarzeniowe* i *procesowe*) mogą być również budowane w arkuszach, chociaż wymaga to bardziej zaawansowanej wiedzy o funkcjach arkusza kalkulacyjnego. W modelach *zdarzeniowych* jeden wiersz w arkuszu odpowiada jednemu zdarzeniu, natomiast w symulacjach *procesowych* w jednym wierszu arkusza modeluje się ciąg zdarzeń logicznych związanych z jednym klientem (zgłoszeniem).

Poważnym mankamentem modeli o *zmiennym kroku czasowym* budowanych w arkuszu jest konieczność stworzenia specjalizowanych procedur zapisanych na przykład w VBA (*Visual Basic Applications*), umożliwiających śledzenie zapisu historii pobytu obiektów dynamicznych w systemie. Niektórzy autorzy (por. [Hora 2003] czy [Ingolfsson i Grossman 2002]) proponują jednak rozwiązania pozwalające na zbudowanie modelu dyskretnego o *zmiennym kroku czasowym* wyłącznie za pomocą standardowych funkcji Excela. Hora [2003] proponuje wykorzystanie funkcji tablicowych wprowadzanych za pomocą zestawu klawiszy CTRL+SHIFT+ENTER i prezentuje prosty model kolejkowy o nieograniczonej kolejce obsługiwanej według zasady FIFO. Grossman [1999] przedstawia model kolejkowy z kilkoma stanowiskami obsługi, uwzględniający bardziej złożone zagadnienia modelowania systemów dyskretnych, takie jak: rezygnacja z oczekiwania w kolejce ze względu na brak miejsca (ang. *balloking*), rezygnacja z oczekiwania w kolejce po uprzednim spędzeniu tam jakiegoś czasu (ang. *reneging*) czy zamiana kolejki (ang. *jockeying*).

7.9.2. MODEL KOLEJKOWY MONTE CARLO – METODA PRZEGLĄDU DZIAŁAŃ

Liczba statków podplywających do nabrzeża portowego opisana jest dyskretnym rozkładem losowym (por. tab. 7.12). W porcie pracują 3 dźwigi. Statki nieobsłużone danego dnia formują kolejkę i są obsługiwane w kolejnym dniu. Zakładamy, że rozładunek statków rozpoczyna się dopiero po wpłynięciu wszystkich statków do portu oraz że jeden dźwig może rozładować jednego dnia dokładnie jeden statek. Symulacja prowadzona jest metodą przeglądu działań dla zadeklarowanego okresu (1 miesiąc), a celem symulacji jest zbadanie dynamiki tworzenia się kolejki oczekujących statków (rys. 7.18).

Tabela 7.12. Liczba statków podplywających dziennie do nabrzeża

Liczba statków [szt]	Prawdopodobieństwo
0	0,13
1	0,17
2	0,15
3	0,25
4	0,2
5	0,1

Na rysunku 7.17 przedstawiono model kolejkowy zbudowany w arkuszu. Jeden wiersz opisuje wszystkie *działania*, które mają miejsce w systemie w badanym odcinku czasu, przyjętym jako stały krok symulacji (tu – w ciągu 1 dnia). Budując modele kolejkowe w arkuszu, musimy pamiętać o tym, że formuły umieszczone w pierwszym wierszu różnią się czasami od formuł wpisanych w wierszach kolejnych, dlatego przeciąganie formuł zaczynamy zawsze od wiersza drugiego. W prezentowanych dalej modelach kolejkowych Monte Carlo w razie wystąpienia różnic między pierwszym i drugim wierszem modelu opisywane są obie formuły. W pozostałych przypadkach opisywane są formuły wiersza drugiego.

Kolumna H: liczba statków podplywających do nabrzeża w ciągu jednego dnia. Wartość ta symulowana jest za pomocą poznanej już wcześniej formuły.

$H13 = WYSZUKAJ.PIONOWO(G13; \$C\$3: \$E\$8; 3)$

	A	B	C	D	E	F	G	H	I	J	K
1											
2		P0	lewy zakres	P0 skumulowane	Liczba statków przybywających do portu						
3		0,13	0	0,13	0						
4		0,17	0,13	0,3	1						
5		0,15	0,3	0,45	2						
6		0,25	0,45	0,7	3						
7		0,2	0,7	0,9	4						
8		0,1	0,9	1	5						
9		1									
10											
11		Liczba dostępnych dźwigów	3			Kolejny dzień	Liczba losowa	liczba statków przybywających do portu	liczba statków oczekująca na rozładunek	obsłużone statki	nieobsłużone statki
12						1	0,0453	0	0	0	0
13						2	0,6778	3	3	3	0
14						3	0,3403	2	2	2	0
15						4	0,4594	3	3	3	0
16						5	0,8517	4	4	3	1
17						6	0,7758	4	5	3	2
18						7	0,6614	3	5	3	2
19						8	0,7555	4	6	3	3
20						9	0,7702	4	7	3	4
21						10	0,0926	0	4	3	1
22						11	0,4529	3	4	3	1
23						12	0,4837	3	4	3	1
24						13	0,1417	1	2	2	0
25						14	0,8473	4	4	3	1
26						15	0,3300	2	3	3	0
27						16	0,5390	3	3	3	0
28						17	0,2833	1	1	1	0
29						18	0,9223	5	5	3	2
30						19	0,3991	2	4	3	1
31						20	0,4088	2	3	3	0

Rys. 7.17. Model kolejkowy według podejścia przeglądu działań. Fragment.

Źródło: opracowanie własne

Kolumna I: kolejka statków oczekujących na rozładunek. W pierwszym dniu będą to wszystkie statki, które wpłynęły do portu. W drugim dniu będzie to suma statków nieobsłużonych dnia poprzedniego oraz statków nowo przybyłych.

$$I12 = H12$$

$$I13 = H13 + K12$$

Kolumna J: liczba statków obsłużonych w danym dniu jest mniejszą wartością z liczby statków oczekujących w kolejce i liczby dostępnych dźwigów.

$$J13 = \text{MIN}(I13; \$C\$11)$$



Rys. 7.18. Dynamika kształtowania się kolejki statków w modelu *przeglądu działań*
Źródło: opracowanie własne

Kolumna K: liczba statków, które pozostają do rozładunku w dniu następnym jest różnicą pomiędzy długością kolejki i liczbą statków obsłużonych danego dnia.

$$K13 = I13 - J13$$

Symulację prowadzimy dla 30 dni (na rys. 7.17 pokazano tylko 20 dni symulacji). Jeżeli interesują nas wartości średnie (np. średnia długość kolejki), zadeklarowany okres symulacji powinien być następnie wielokrotnie powtórzony w kolejnych przebiegach symulacyjnych. Możemy w tym celu wykorzystać poznaną wcześniej funkcję Excela *Tabela*.

7.9.3. MODEL KOLEJKOWY MONTE CARLO – METODA KOLEJNYCH ZDARZEŃ

Kolejkowy model *zdarzeniowy* koncentruje się na opisie zmian zachodzących w systemie w momencie, kiedy pojawia się kolejne *zdarzenie*, wykorzystując w tym celu jeden wiersz arkusza kalkulacyjnego.

Do prostego systemu obsługi z jednym stanowiskiem klienci przybywają pojedynczo. Czas między nadejściem kolejnych klientów opisany jest rozkładem wykładniczym o średniej 6 minut. Czas obsługi klientów na stanowisku opisany jest również

rozkładem wykładniczym o średniej 7 minut (por. rys. 7.19). W modelu zdefiniowano dwa zdarzenia: *Przybycie* nowego zgłoszenia do systemu i *Zakończenie* obsługi zgłoszenia na stanowisku. Stanowisko obsługi może pozostawać w trybie *Zajęte* lub *Wolne*.

	A	B	C	D	E	F	G
2							
3	Parametry wejściowe						
4							
5	Średni czas pomiędzy zgłoszeniami [min]						6
6	Średni czas obsługi [min]						7
7	Liczba stanowisk obsługi						1
8							
9	Typy zdarzeń:			Status stanowiska			
10	Przybycie			Wolne			
11	Zakończenie			Zajęte			
12							
13	Wyniki						

Rys. 7.19. Dane do modelu kolejkowego wg podejścia zdarzeniowego

Rysunek 7.20 przedstawia pierwszych 15 zdarzeń wygenerowanych w modelu. Każdy wiersz arkusza odpowiada dynamicznie kreowanemu zdarzeniu, a czas symulacji przesuwany jest wg zmiennego kroku (por. kolumna *Zegar* na rys. 7.20). Zgodnie z procedurą przebiegu symulacji *zdarzeniowej* (por. rozdz. 4), w momencie pojawienia się nowego zdarzenia w systemie (czyli w każdym wierszu w arkuszu) planowane są kolejne zdarzenia – kolumna *Następne Przybycie* (Kolumna P) i *Następne Zakończenie* (Kolumna Q). Komórki P i Q w każdym wierszu odzwierciedlają aktualną wartość *Kalendarza Zdarzeń* (por. rozdz. 4). W przypadku kiedy zdarzenie nie może być zaplanowane (np. wtedy, gdy zakończyła się obsługa zgłoszenia, ale kolejka jest pusta, por. wiersz dla zdarzenia nr 2 na rys. 7.20), w komórce Q arkusza wpisywana jest liczba o dużej wartości (tu: 99 999,00).

	H	I	J	K	L	M	N	O	P	Q
3										
4		Zdarzenie	Typ	Zegar	Liczba zgłoszeń w systemie	Liczba zgłoszeń w kolejce	Status stanowiska	Czas bezczynności	Następne Przybycie	Następne Zakończenie
5		1	Przybycie	8,43	1	0	Zajęte	0,00	8,62	8,45
6		2	Zakończenie	8,45	0	0	Wolne	0,02	8,62	99999,00
7		3	Przybycie	8,62	1	0	Zajęte	0,00	10,70	11,49
8		4	Przybycie	10,70	2	1	Zajęte	0,00	14,83	11,49
9		5	Zakończenie	11,49	1	0	Zajęte	0,00	14,83	14,77
10		6	Zakończenie	14,77	0	0	Wolne	3,28	14,83	99999,00
11		7	Przybycie	14,83	1	0	Zajęte	0,00	21,87	40,03
12		8	Przybycie	21,87	2	1	Zajęte	0,00	25,18	40,03
13		9	Przybycie	25,18	3	2	Zajęte	0,00	28,37	40,03
14		10	Przybycie	28,37	4	3	Zajęte	0,00	33,03	40,03
15		11	Przybycie	33,03	5	4	Zajęte	0,00	39,43	40,03
16		12	Przybycie	39,43	6	5	Zajęte	0,00	49,61	40,03
17		13	Zakończenie	40,03	5	4	Zajęte	0,00	49,61	42,85
18		14	Zakończenie	42,85	4	3	Zajęte	0,00	49,61	47,69
19		15	Zakończenie	47,69	3	2	Zajęte	0,00	49,61	48,35

Rys. 7.20. Model kolejkowy wg podejścia zdarzeniowego
Źródło: opracowanie własne na podstawie [Evans i Olson 2002]

Kolumna J: zdarzenie pobrane do realizacji z *Kalendarza Zdarzeń*.

J5 = B10. Pierwszym zdarzeniem jest zawsze *Przybycie* pierwszego klienta.

$J6 = JEŻELI(Q5 < P5; \$B\$11; \$B\$10)$. Sprawdzamy, które zdarzenie pojawi się w systemie wcześniej: *Przybycie* czy *Zakończenie*.

Kolumna K: wskazanie zegara symulacji, czyli czas realizacji zdarzenia z kolumny J.
 $K5 = -G5 \cdot LN(LOS())$. Przybycie pierwszego klienta generujemy z rozkładu wykładniczego o zadanym parametrze. Ścisłej, generujemy czas, jaki upływa od chwili 0,0.
 $K6 = MIN(P5; Q5)$. Zegar symulacji ustawi się na czas pojawienia się wcześniejszego ze zdarzeń.

Kolumna L: liczba zgłoszeń przebywających w systemie.
 $L5 = 1$. W momencie rozpoczęcia symulacji w systemie znajduje się tylko pierwszy (jeden) klient.
 $L6 = JEŻELI(J6 = \$B\$11; L5 - 1; L5 + 1)$. Jeżeli mamy do czynienia ze zdarzeniem *Zakończenie*, liczba klientów zmniejszy się o jeden. Jeżeli realizowanym zdarzeniem jest *Przybycie*, liczba klientów zwiększy się o jeden.

Kolumna M: liczba zgłoszeń oczekujących w kolejce na obsługę.
 $M5 = 0$. W momencie startu symulacji żaden klient nie czeka w kolejce.
 $M6 = MAX(L6 - 1; 0)$. Liczba zgłoszeń w kolejce to liczba zgłoszeń w systemie pomniejszona o jeden (rozważamy jedno stanowisko obsługi). Funkcja $MAX()$ nie dopuszcza do ujemnej długości kolejki.

Kolumna N: status stanowiska obsługi
 $N5 = E11$. W momencie startu symulacji i przybycia pierwszego klienta stanowisko jest zawsze *Zajęte*.
 $N6 = JEŻELI(L6 = 0; \$E\$10; \$E\$11)$. Jeżeli w systemie nie ma zgłoszeń stanowisko jest *Wolne*, natomiast w przeciwnym przypadku jest *Zajęte*.

Kolumna O: badamy, czy czas, który upłynął od ostatniego zdarzenia był czasem beczynności, czy czasem, w którym stanowisko było zajęte obsługą zgłoszeń. W kolumnie podajemy długość czasu beczynności.

$O5 = 0$.
 $O = JEŻELI(N5 = \$E\$10; K6 - K5; 0)$. Jeżeli stanowisko oznaczone było poprzednio jako *Wolne*, czas od ostatniego zdarzenia był czasem beczynności. W przeciwnym razie czas beczynności wynosił zero.

Kolumna P: aktualizujemy czas *Przybycia* kolejnego zgłoszenia.
 $P5 = K5 + (-G5 \cdot LN(LOS()))$. Ponieważ pierwszym zdarzeniem jest zawsze *Przybycie*, zatem kolejne zgłoszenie przybędzie zgodnie rozkładem wykładniczym.
 $P6 = JEŻELI(J6 = \$B\$10; K6 + (-G5 \cdot LN(LOS())); P5)$. Jeżeli aktualnym zdarzeniem jest *Przybycie*, to kolejne zgłoszenie przybędzie zgodnie rozkładem wykładniczym. Jeżeli aktualnym zdarzeniem jest *Zakończenie*, przybycie kolejnego zgłoszenia już zostało zaplanowane i należy je przepisać z wiersza wcześniejszego.

Kolumna Q: aktualizujemy czas *Zakończenia* obsługi zgłoszenia.

$Q5 = K5 + (-G6 \cdot LN(LOS()))$. Dla pierwszego zgłoszenia czas zakończenia obsługi generowany jest z rozkładu wykładniczego o średniej 7 minut.

Tabela 7.13. Formuła z kolumny Q modelu z rys.7.19

$Q6=$		
$JE\acute{Z}ELI(ORAZ(J6=\$B\$11;L6>0);$		Jeżeli aktualnym zdarzeniem jest <i>Zakończenie</i> , a w systemie przebywają inne zgłoszenia ($L6 > 0$), to:
	$K6+(-\$G\$6 \cdot LN(LOS()));$	czas zakończenia obsługi generowany jest z rozkładu wykładniczego o średniej 7 minut.
		W przeciwnym razie:
$JE\acute{Z}ELI(ORAZ(J6=\$B\$11;L6=0);$		Jeżeli aktualnym zdarzeniem jest <i>Zakończenie</i> , a w systemie NIE przebywają inne zgłoszenia ($L6 = 0$), to:
	99999;	czas zakończenia obsługi nie powinien być generowany (w systemie nie ma zgłoszeń, których obsługa mogłaby się zakończyć). Wpisujemy zatem do komórki bardzo dużą wartość, wykraczającą poza spodziewane wskazania zegara symulacji.
		W przeciwnym razie:
$JE\acute{Z}ELI(ORAZ(J6=\$B\$10;N5=\$E\$10);$		Jeżeli aktualnym zdarzeniem jest <i>Przybycie</i> , a stanowisko obsługi było <i>Wolne</i> , to:
	$K6+(-\$G\$6 \cdot LN(LOS()));$	możemy pobrać zgłoszenie z kolejki i wyznaczyć czas zakończenia obsługi tego zgłoszenia z rozkładu wykładniczego.
	$Q5))$	W przeciwnym razie przepisujemy czas zakończenia obsługi z wiersza poprzedniego (do systemu przybyło nowe zgłoszenie, ale stanowisko obsługi jest ciągle zajęte obsługą poprzedniego zgłoszenia)

Formuła, która w całości zapisana jest w trzech wierszach poniżej, została szczegółowo omówiona w tabeli 7.13.

$Q6 = JE\acute{Z}ELI(ORAZ(J6 = \$B\$11;L6 > 0);K6+(-\$G\$6 \cdot LN(LOS()));$

$JE\acute{Z}ELI(ORAZ(J6 = \$B\$11;L6 = 0);99999;$

$JE\acute{Z}ELI(ORAZ(J6 = \$B\$10;N5 = \$E\$10);K6 + (-\$G\$6 \cdot LN(LOS()));Q5)))$

Proces symulacji prowadzony jest dla zadanego czasu. Jeżeli interesują nas średnie wartości mierników wyjściowych, takie jak średnia długość kolejki, średni czas oczekiwania w kolejce, średnia liczba zgłoszeń przebywających w systemie itd., musimy symulację powtórzyć wielokrotnie, każdorazowo obliczając wartości interesujących nas mierników.

7.9.4. MODEL KOLEJKOWY MONTE CARLO – METODA INTERAKCJI PROCESÓW

W modelu *interakcji procesów* w jednym wierszu arkusza zapisuje się informacje o wszystkich zdarzeniach związanych z jednym *zgłoszeniem* (klientem). Symulacja przebiega według zmiennego kroku (podobnie jak symulacja *zdarzeniowa*), ale wszystkie informacje na temat pobytu jednego *zgłoszenia* w systemie zebrane są w jednym miejscu, co czyni model *procesowy* bardziej intuicyjnym i zrozumiałym, z zachowaniem przy tym wydajności i elastyczności symulacji według zmiennego kroku.

Model kolejkowy Monte Carlo zbudowany w arkuszu według podejścia *interakcji procesów* omówiono jedynie w sposób ogólny, ze względu na złożoność formuł matematycznych. W przedstawionym modelu (por. rys. 7.21) z trzema stanowiskami obsługi, dla każdego *zgłoszenia* generowany jest z rozkładu losowego czas, jaki upłynie do momentu pojawienia się *zgłoszenia* w systemie (kolumna *Czas do Przybycia*). Na tej podstawie wyznaczany jest *Czas Przybycia* do systemu, który pełni rolę *Zegara Symulacji*. Informacje zapisane we wcześniejszych komórkach kolumny *Koniec Obsługi* pozwalają określić liczbę *zgłoszeń* przebywających w systemie w badanym momencie (są to *zgłoszenia*, których *Koniec Obsługi* jest późniejszy niż aktualne wskazanie *Zegara Symulacji*). *Początek Obsługi* jest tożsamy z *Czasem Przybycia* lub wyznaczany jest poprzez przejrzanie wszystkich wcześniejszych zapisów w kolumnie *Koniec Obsługi* i ustalenie na tej podstawie momentu, w którym może się rozpocząć obsługa *zgłoszenia*. *Czas Obsługi* jest generowany z odpowiedniego rozkładu losowego.

Zgłoszenie	Czas do Przybycia	Czas Przybycia	Liczba zgłoszeń w systemie	Początek obsługi	Czas obsługi	Koniec obsługi	Czas oczekiwania	Czas w systemie	Kolejka w chwili przybycia	Liczba zajętych stanowisk w chwili przybycia
1	1,72	1,72	0	1,72	5	6,72	0,00	5,00	0	0
2	0,51	2,24	1	2,24	5	7,24	0,00	5,00	0	1
3	0,10	2,34	2	2,34	2	4,34	0,00	2,00	0	2
4	0,95	3,29	3	4,34	2	6,34	1,05	3,05	0	3
5	1,23	4,51	3	6,34	2	8,34	1,83	3,83	0	3
6	3,92	8,43	0	8,43	2	10,43	0,00	2,00	0	0
7	1,88	10,31	1	10,31	5	15,31	0,00	5,00	0	1
8	0,23	10,53	1	10,53	5	15,53	0,00	5,00	0	1
9	1,04	11,58	2	11,58	5	16,58	0,00	5,00	0	2
10	1,03	12,61	3	15,31	2	17,31	2,70	4,70	0	3
11	3,74	16,36	2	16,36	2	18,36	0,00	2,00	0	2

Rys. 7.21. Model kolejkowy wg podejścia *interakcji procesów*

Źródło: opracowanie własne na podstawie [Hora 2003]

W modelach *przeglądu działań* i *zdarzeniowym* w każdym kolejnym kroku symulacji model odwołuje się tylko do ostatniego chronologicznie zapisu. W modelu *procesowym* każdy nowy wiersz arkusza tworzony jest poprzez analizę stanów zapisanych we wszystkich poprzednich wierszach.

8. ARENA – ŚRODOWISKO SYMULACJI DYSKRETNEJ

8.1. PRACA Z PROGRAMEM – INFORMACJE OGÓLNE

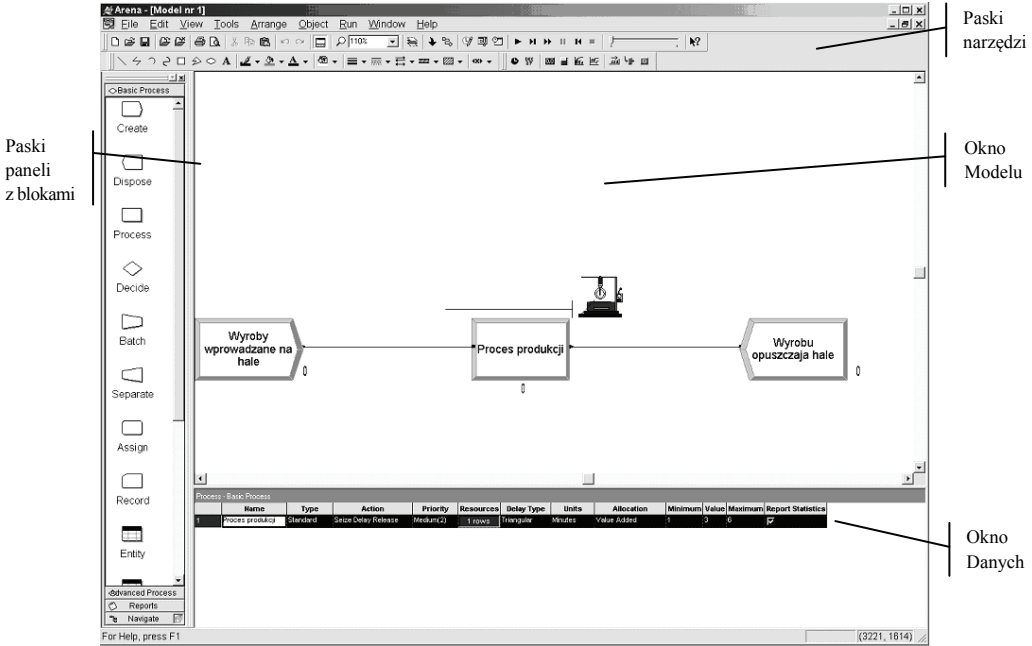
Arena jest środowiskiem do prowadzenia symulacji dyskretnej na bazie podejścia interakcji procesów. Modele w Arenie budowane są za pomocą schematów graficznych, których poszczególne elementy wypełniane są wartościami i relacjami logicznymi. Konstruując schemat, musimy pamiętać o tym, że przepływ elementów odbywa się w czasie. Zgłoszenia przemieszczają się od jednego bloku schematu do drugiego, co może oznaczać (ale nie musi), że w momencie przejścia zgłoszenia do kolejnego bloku zegar symulacji wskazuje już inną godzinę. Drugą istotną cechą budowanych modeli graficznych jest to, że definiowane są one z punktu widzenia zgłoszenia, a nie systemu. Każda zatem wartość wprowadzona poprzez bloki schematu, każda relacja czy definicja dotyczy tylko i wyłącznie zgłoszeń, które do tego bloku trafiły. Definicje te są niewidoczne dla zgłoszeń, których droga przejścia przez model biegnie poprzez inne bloki.

Arena jest aplikacją pracującą w środowisku Microsoft® Windows® i wszystkie typowe dla tego środowiska polecenia, komendy, reguły mają tu również zastosowanie.

Praca z Areną może przebiegać na różnych poziomach. Z jednej strony użytkownik może korzystać z gotowych struktur i predefiniowanych procesów, koncentrując się wyłącznie na odtworzeniu działania systemu, a całą trudność związaną z oprogramowaniem modelu pozostawiając Arenie. Z drugiej strony, użytkownik poszukujący elastyczności, jaką dają języki programowania, może samodzielnie tworzyć kod programu. Pomiędzy tymi płaszczyznami istnieje cała gama możliwości pośrednich, Arena bowiem pozwala dowolnie zestawiać różne dostępne poziomy kreowania modeli.

W tej książce zostaną omówione struktury podstawowe, za pomocą których można stworzyć typowe modele kolejkowe, wystarczające do odtworzenia większości prostych systemów obsługi.

Ekran Areny podzielony jest na kilka obszarów (por. rys. 8.1). Największą powierzchnię zajmuje *Okno Modelu*, w którym układane są *bloki** schematu pobrane ze znajdującego się po lewej stronie *Panelu*. W ten sposób tworzony jest schemat graficzny modelu systemu. Mniejszy, dolny fragment ekranu przeznaczony jest na *Okno Danych*, w którym wprowadza się dane do *bloków danych*, również wybierane z *Paneli*.



Rys. 8.1. Główne okno Areny. Schemat przykładu nr 1

Bloki, z których buduje się schemat graficzny i bloki służące wyłącznie do przechowywania wartości liczbowych różnią się wyraźnie kształtem i kolorystyką. Te ostatnie przypominają małe arkusze kalkulacyjne (por. rys. 8.2). Odmienny jest również sposób wprowadzania obu typów bloków do modelu. *Bloki schematu* są przeciągane myszą, natomiast *bloki danych* należy wskazać, co spowoduje pojawienie się odpowiednich komórek w *Oknie Danych*.



Rys. 8.2. Zróżnicowany wygląd *bloku schematu* (lewy) i *bloku danych* (prawy)

* Będziemy zamiennie używać słów *blok* i *moduł* na określenie obiektów graficznych, z których budowane będą modele Areny.

Z każdym *blokiem schematu* powiązane jest okno dialogowe, które otwiera się po dwukrotnym kliknięciu na blok (okna dialogowe podstawowych bloków będą szczegółowo omówione w dalszych podrozdziałach). Pojedyncze kliknięcie na *blok schematu* spowoduje natomiast pojawienie się w dolnej części ekranu (w *Oknie Danych*) wiersza z komórkami, do których można bezpośrednio wprowadzać wartości z pominięciem okna dialogowego.

Blok danych nie ma swojego odpowiednika w postaci okna dialogowego i wartości wprowadzane są bezpośrednio do komórek.

Umieszczone z lewej strony ekranu *Paski Paneli* zawierają różne zestawy *bloków schematu* i *bloków danych*. Podstawowym panelem jest *Basic Process*, z którego bloków można zbudować proste modele. Panel *Advanced Process* pozwala na budowę bardziej złożonych struktur. Dostępność innych paneli zależy od posiadanego typu licencji.

Wybierane z poziomu *menu* opcje *View > Split Screen* oraz *View > Project Bar* pozwalają dowolnie sterować podziałem ekranu na okna.

8.2. PRZYKŁADOWY MODEL NR 1. PROCES PRODUKCJI Z JEDNYM STANOWISKIEM

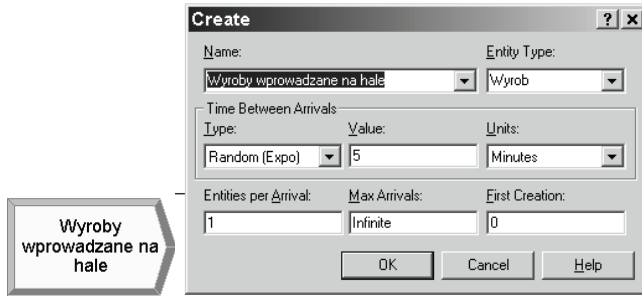
8.2.1. OPIS PRZYKŁADU

W dalszej części podrozdziału zostaną omówione podstawowe zasady konstruowania modeli w Arenie z wykorzystaniem bloków panelu *Basic Process* na przykładzie modelu przedstawionego na rys. 8.1. Model ten odpowiada prostemu systemowi obsługi opisanemu w podrozdziale 4.1 (por. rys. 4.1). Dla przypomnienia, jest to system z jednym stanowiskiem obsługi i jedną kolejką obsługiwaną wg reguły *FIFO*. Zgłoszenia (tu – wyroby) napływają do systemu (na halę produkcyjną), są poddawane obróbce na stanowisku obróbki (o nazwie *Maszyna*), po czym opuszczają halę produkcyjną. Będziemy chcieli określić: średni czas pobytu w systemie, średni czas oczekiwania w kolejce, średnią długość kolejki, wykorzystanie stanowiska obsługi i jeszcze kilka innych mierników.

8.2.2. BLOK SCHEMATU *CREATE*

Blok *Create* jest pierwszym węzłem budowanego modelu. Jego rolą jest wprowadzenie zgłoszeń do modelu według zadanego schematu. W świecie rzeczywistym zgłoszenia pojawiają się w systemie, przechodząc z otoczenia systemu: przychodzą klienci banku, klienci supermarketu, podjeżdżają ciężarówki z towarem, na rozdziel-

nię poczty trafiają paczki z listami itd. W modelu musimy zapewnić dopływ zgłoszeń według zasad zaobserwowanych w systemie rzeczywistym. Jeżeli zatem w systemie zgłoszenia pojawiają się zawsze w grupach o określonej liczebności (np. dana liczba palet na ciężarówce), w modelu musimy to wiernie odtworzyć. Jeżeli klienci przychodzą pojedynczo lub dwójkami (matki z dziećmi do lekarza), ten schemat powinien pojawić się w modelu. Rolą modułu *Create* jest właśnie wprowadzanie zgłoszeń do modelu według określonych przez modelującego zasad.



Rys. 8.3. Ikona bloku *Create* i okno dialogowe bloku *Create*

Create - Basic Process								
	Name	Entity Type	Type	Value	Units	Entities per Arrival	Max Arrivals	First Creation
1	Wyroby wprowadzane na hale	Wyrob	Random (Expo)	5	Minutes	1	Infinite	0


Rys. 8.4. Wiersz z komórkami liczbowymi sprzężony z oknem dialogowym *Create*

Moduł *Create* umieszczamy z lewej strony ekranu i klikamy dwukrotnie, aby otworzyć okno dialogowe (por. rys. 8.3). Automatycznie, w dolnej części ekranu otwiera się wiersz z tymi samymi komórkami (por. rys. 8.4), które widoczne są w oknie dialogowym. Nie ma znaczenia, za pomocą którego narzędzia będą wprowadzane wartości.

Każdy blok wprowadzany do modelu musi mieć swoją unikalną nazwę. Nazwa ta nie może powtarzać się w żadnym innym miejscu modelu (tak samo zatem nie może nazywać się inny blok, stanowisko obsługi, zmienna czy atrybut). W każdym bloku znajdziemy pole *Name*, za pomocą którego nadajemy nazwę blokowi. Nazwy powinny być proste, czytelne i odpowiadające wiernie czynności czy obiektowi, które są z ich pomocą opisywane. Nazwy nie mogą zawierać polskich znaków ani znaków specjalnych typu %\$#@ itd.

Nasz blok *Create* został nazwany Wyroby wprowadzane na hale. W kolejnym polu *Entity Type* wpisujemy słowo Wyrob. Przypomnijmy: *Entity* to *Zgłoszenie*, czyli obiekt dynamiczny przemieszczający się przez system. W przykładzie zgłoszeniami są wyroby wprowadzone na halę produkcyjną i poddane obróbce na stanowisku obsługi. W modelu może występować wiele *typów* zgłoszeń: możemy rozpatrywać wyroby typu A, B i C; rozważać pacjentów przychodzących na pierwszą i na kolejną wizytę; uwzględniać listy zwykłe, polecane i paczki. Celowe jest zatem zdefiniowanie

każdej grupy zgłoszeń oddzielnie. W naszym prostym przykładzie będzie występowała tylko jedna grupa o nazwie `Wyrob`.

W środkowej części okna dialogowego znajduje się wyraźnie odznaczony fragment o nazwie *Time Between Arrivals* (por. rys. 8.3), za pomocą którego definiujemy *odstęp czasu pomiędzy kolejno pojawiającymi się zgłoszeniami* (por. podrozdz. 3.7 – Generowanie strumieni zgłoszeń). W przykładzie wybieramy określenie `Random (Expo)` (za pomocą strzałki wyboru ) tak, aby czas pomiędzy kolejnymi zgłoszeniami był generowany według rozkładu wykładniczego (`Expo`, czyli *Exponential* to rozkład wykładniczy). W polu *Value* wpisujemy 5, a w polu *Units – minutes*, dzięki czemu Arena będzie generowała nowe zgłoszenia według rozkładu wykładniczego o średniej 5 minut. Przypomnijmy – średnia rozkładu nie oznacza, że zgłoszenia będą pojawiały się dokładnie co 5 minut! Odstęp czasu może wynosić 1 minutę, 15 minut czy nawet 60 minut. Średnia z tych losowych wartości będzie równa 5 minut, pod warunkiem, że dostatecznie długo będziemy zbierali dane.


W najniższym wierszu okna dialogowego mamy trzy pola. *Entities per Arrival* informuje nas, ile zgłoszeń jednocześnie pojawia się w systemie. Wrócimy jeszcze do tego pola, a teraz wpiszymy tam wartość 1, co będzie oznaczało, że wyroby są wprowadzane na halę pojedynczo.

Pole *Max Arrivals* nakłada ograniczenie na liczbę zgłoszeń wygenerowanych modulem *Create*. W większości badanych systemów to ograniczenie nie występuje (co potwierdzamy słowem *Infinite*, czyli *Nieskończoność*). Do galerii handlowej może przyjść dowolna liczba klientów, podobnie do banku czy do restauracji. Na seans filmowy wpuścimy już jednak tylko pewną maksymalną liczbę osób, podobnie na wizytę lekarską zapiszemy nie więcej niż na przykład 30 pacjentów. W takich przypadkach wartość wpisana w polu *Max Arrivals* spowoduje zatrzymanie procesu kreowania zgłoszeń po wygenerowaniu maksymalnej zadeklarowanej wartości.

Ostatnie pole *First Creation* pozwala rozpocząć proces symulacji w innym niż zerowym momencie. Możliwość opóźniania modelowanych procesów ma znaczenie w bardziej złożonych modelach.

8.2.3. BLOK DANYCH *ENTITY*

W momencie kiedy w bloku *Create* wypełniane jest pole *Entity Type*, automatycznie kreowany jest kolejny wiersz z danymi w bloku *Entity*, odpowiadający nowozdefiniowanemu typowi zgłoszeń (por. rys. 8.5). Zawartość modułu można obejrzeć i poddać edycji, wybierając ikonę *Entity* w panelu *Basic Process*. W polu *Initial Picture* modułu *Entity* można rozwinąć listę dostępnych ikon i wybrać jedną, która w trakcie symulacji będzie animowała przemieszczające się zgłoszenia. Domyślnie wybierana jest ikonka kartki papieru (*Picture.Report*), ale często bardziej odpowiednie będą inne obrazki. W przykładzie zdecydowano się na obrazek czerwonej piłeczki (*Picture.Red Ball*).



Entity

Entity - Basic Process									
	Entity Type	Initial Picture	Holding Cost / Hour	Initial VA Cost	Initial NVA Cost	Initial Waiting Cost	Initial Tran Cost	Initial Other Cost	Report Statistics
1	Wyrob	Picture.Red.Bal	0.0	0.0	0.0	0.0	0.0	0.0	<input checked="" type="checkbox"/>

Double-click

- Picture.Red.Bal
- Picture.Red.Page
- Picture.Report
- Picture.Telephone
- Picture.Truck
- Picture.Van
- Picture.Widgets

Rys. 8.5. Ikona i wiersz bloku danych *Entity*

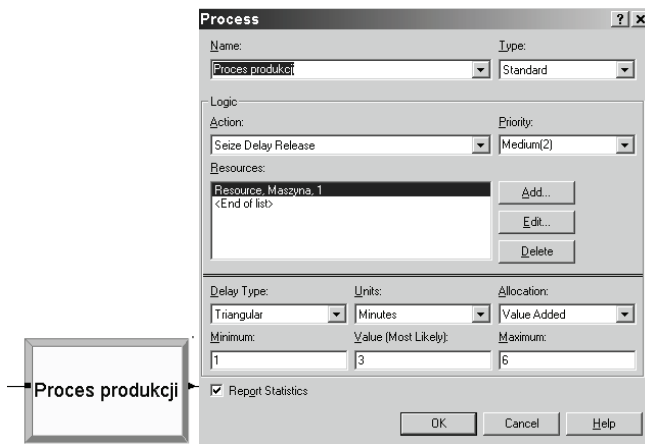
8.2.4. BLOK SCHEMATU *PROCESS*

Moduł *Process* umożliwia zdefiniowanie dowolnego jednostkowego procesu łącznie ze stanowiskami obsługi, które biorą w nim udział, kolejkami, które mogą się tworzyć w ramach tego procesu i opóźnieniem związanym z realizacją procesu. Okno dialogowe modułu *Process* przedstawia rys. 8.6, a wiersz modułu danych – rys. 8.7.

Po wprowadzeniu nazwy bloku w polu *Name* (Proces produkcji), wybieramy *Standard* w polu *Type*, co oznacza, że relacje logiczne związane z tym procesem będą zdefiniowane w tym oknie dialogowym, a nie za pomocą podmodelu.

Największą część okna dialogowego zajmuje fragment opatrzony nazwą *Logic*, który bardzo dokładnie opisuje, co dzieje się ze zgłoszeniem, które trafiło do tego bloku.

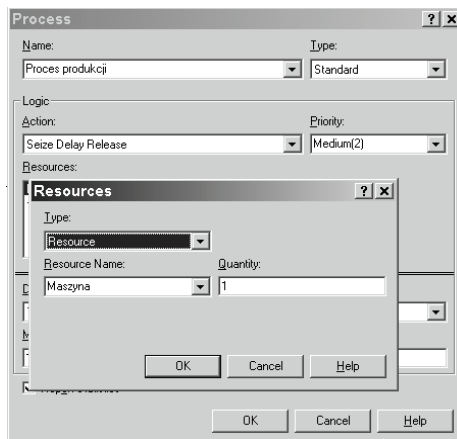
W polu *Action* wybieramy *Seize Delay Release*, co oznacza, że każde zgłoszenie, które znajdzie się w tym bloku, najpierw Zajmie (*Seize*) stanowisko obsługi (choć może wcześniej będzie musiało czekać w kolejce), następnie będzie na nim Przebywać (*Delay*), aż w końcu je Zwolni (*Release*) i umożliwi kolejnym zgłoszeniom przejście stanowiska.

Rys. 8.6. Ikona bloku *Process* i okno dialogowe bloku *Process*

Process - Basic Process												
Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Minimum	Value	Maximum	Report Statistics	
1	Proces produkcji	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Minutes	Value Added	1	3	6	<input checked="" type="checkbox"/>

Rys. 8.7. Wiersz z komórkami liczbowymi sprzężony z oknem dialogowym *Process*

Niekoniecznie wszystkie trzy czynności muszą być zrealizowane w ramach jednego procesu jednostkowego. Zgłoszenie może zająć stanowisko w ramach jednego procesu, ale zwolnić je w ramach następnego procesu. Na przykład w modelu przychodni zdrowia definiujemy jeden proces, w którym *pacjent* (zgłoszenie) trafia do gabinetu zabiegowego i jest przygotowywany do zabiegu przez *pielęgniarkę* (pielęgniarka będzie zdefiniowana jako stanowisko obsługi), a następnie **razem z pielęgniarką** przeprowadzany jest do innego pomieszczenia, w którym wykonywany jest właściwy zabieg (oczywiście pacjent będzie musiał „zająć” jeszcze inne stanowiska obsługi, np. sprzęt do przeprowadzenia zabiegu). W ramach jednego procesu jednostkowego stanowisko obsługi jest zajmowane, w ramach innego – zwalniane. W modelu definiujemy tego typu sytuacje poprzez odpowiedni dobór słów kluczowych *Seize Delay Release* i wprowadzenie większej liczby bloków *Process*.

Rys. 8.8. Okno dialogowe bloku *Process-Resources*

Stanowisko obsługi (*Resource*), które będzie zajmowane/zwalniane w ramach danego procesu jednostkowego musi oczywiście zostać wskazane. W tym celu klikamy na przycisk *Add* (rys. 8.8) i za pomocą nowego okna dialogowego wprowadzamy nazwę stanowiska obsługi (*Maszyna*) i liczbę jednostek tego stanowiska (pole *Quantity* = 1), która będzie przez zgłoszenie *zajmowana* i/lub *zwalniana*. Najczęściej (w prostszych modelach) zgłoszenie zajmuje i zwalnia jedno stanowisko danego typu w ramach jednego procesu, ale oczywiście nie musi to być regułą. We wspomnianym modelu przychodni zdrowia, w wykonaniu właściwego zabiegu muszą na przykład uczestniczyć dwie pielęgniarki. Wtedy w polu *Quantity* wpisujemy liczbę 2. Możemy mieć również do czynienia z sytuacją, kiedy realizacja procesu jednostkowego wymaga jednoczesnego zajęcia dwóch różnych stanowisk obsługi. W modelu przychodni, wykonanie zabiegu wymaga dostępności pielęgniarki i aparatury specjalistycznej.

W tym przypadku wprowadzamy informacje o stanowiskach obsługi niezbędnych do realizacji danego procesu poprzez sukcesywne wybieranie przycisku *Add* i wprowadzanie nazw kolejnych stanowisk.

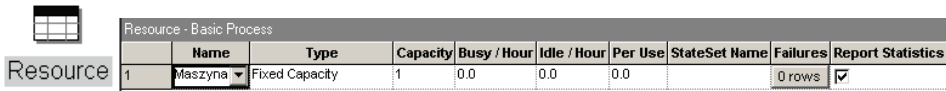
Pole *Quantity* **nie** określa liczby jednostek stanowiska obsługi dostępnych w całym systemie. Tę wielkość definiujemy za pomocą bloku danych *Resource* (por. podrozdz. 8.2.4).

W oknie dialogowym *Process* (rys. 8.6) pozostaje nam jeszcze do wypełnienia dolna część okna, za pomocą której definiujemy czas spędzany przez zgłoszenie na stanowisku (stanowiskach) obsługi. W polu *Delay Type* wybieramy z listy jeden z trzech rozkładów losowych (*Normal* = Normalny, *Triangular* = Trójkątny, *Uniform* = Jednostajny), wartość stałą (*Constant*) lub wyrażenie (*Expression*). Wybór rozkładu losowego powoduje automatyczne dostosowanie liczby parametrów do typu rozkładu. Z kolei wybór *Expression* umożliwi zdefiniowanie czasu trwania obsługi za pomocą dowolnie skonstruowanego wyrażenia matematycznego (szerzej o *Expressions* – por. podrozdz. 8.4.4).

Pole *Units* ustala jednostki czasu, w których obliczany będzie czas pobytu na stanowisku (*Delay*), a pole *Allocation* określa, z jaką grupą kosztów to opóźnienie będzie powiązane.

8.2.5. BLOK DANYCH *RESOURCE*

Podobnie jak definicja typu zgłoszenia pojawiała się automatycznie w bloku danych *Entity* po wypełnieniu pola *Entity Type* w bloku *Create*, tak definicja kolejnych występujących w modelu stanowisk obsługi (*Resources*) jest wprowadzana do bloku danych *Resource* po wypełnieniu okna dialogowego w bloku *Process*. Zawartość bloku danych *Resource* jest widoczna po kliknięciu na odpowiednią ikonę na panelu *Basic Process* (por. rys. 8.9).



Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	Maszyna	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Rys. 8.9. Ikona i zawartość bloku danych *Resource*

Blok danych *Resource* przechowuje charakterystykę wszystkich stanowisk obsługi zdefiniowanych w budowanym modelu. Szczególnie pole *Type* określa, czy pojemność stanowiska jest stała (*Fixed Capacity*) czy zmienia się w trakcie symulacji (*Based on Schedule*). Ze stałą pojemnością mamy do czynienia wtedy, gdy np. liczba otwartych stanowisk kasowych jest stała w ciągu całego okresu pracy supermarketu. Zmienna pojemność jest wtedy, gdy stanowiska pracują według ustalonego harmonogramu, przykładowo: rano trzy, potem w południe pięć, a wieczorem ponownie trzy. W naszym modelu ustalamy, że stanowisko obróbki jest dokładnie jedno w ciągu całej zmiany roboczej. Pozostawiamy zatem *Fixed Capacity* w kolumnie *Type*, a w polu *Capacity* wpisujemy wartość 1.

Alternatywnie, wybór zmiennej pojemności stanowiska obsługi (*Based on Schedule*) spowodowałby wymianę komórki *Capacity* na *Schedule Name*, w której należałoby podać nazwę harmonogramu opisującego dokładnie rytm pracy stanowiska (rys. 8.10) (por. podrozdz.8.3).

Resource - Basic Process										
	Name	Type	Schedule Name	Schedule Rule	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	Maszyna	Based on Schedule	Harmonogram nr 1	Wait	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Rys. 8.10. Alternatywna zawartość bloku *Resource*

8.2.6. BLOK DANYCH *QUEUE*

Jeżeli w modelu przewidziano możliwość zajmowania/zwalniania stanowiska, automatycznie tworzony jest obiekt *Queue* (kolejka), w którym zgłoszenia będą mogły oczekiwać, gdy stanowisko obsługi będzie chwilowo niedostępne. Kolejce nadawana jest nazwa bloku *Process*, w którym przewidziano zajmowanie/zwalnianie (*Seize/Release*) stanowiska i dodawane jest rozszerzenie *Queue*. W naszym modelu, blokowi *Process* nadaliśmy nazwę *Proces produkcji* i dlatego kolejka powiązana z tym blokiem jest identyfikowana jako *Proces produkcji.Queue* (por. rys. 8.11). Zapamiętajmy, że kolejka jest powiązana z procesem jednostkowym, a nie ze stanowiskiem obsługi. Oczywiście, jeżeli w ramach procesu pracuje tylko jedno stanowisko, rozróżnienie to nie ma znaczenia. Powiedzieliśmy jednak wcześniej, że proces jednostkowy może być jednocześnie realizowany na więcej niż jednym stanowisku. W takim przypadku zgłoszenia muszą oczekiwać, aż zwolnią się wszystkie potrzebne stanowiska. To wyjaśnia, dlaczego kolejka powiązana jest z procesem jednostkowym, a nie ze stanowiskiem.

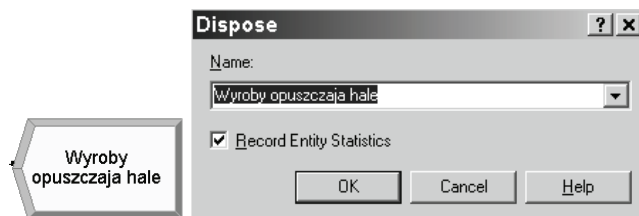
Queue - Basic Process				
	Name	Type	Shared	Report Statistics
1	Proces produkcji.Queue	First In First Out	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Rys. 8.11. Ikona i zawartość bloku danych *Queue*

Za pomocą bloku *Queue* możemy wybrać inną dyscyplinę wyboru zgłoszeń z kolejki. W polu *Type* mamy do dyspozycji oprócz zasad *FIFO* i *LIFO*, również wybór według wartości wskazanego atrybutu (*Lowest Attribute Value* – według najmniejszej wartości atrybutu, *Highest Attribute Value* – według największej wartości atrybutu).


8.2.7. BLOK SCHEMATU *DISPOSE*

Blok *Dispose* (rys. 8.12) jest ostatnim węzłem modelu graficznego. Jego rolą jest pozbycie się (zniszczenie) zgłoszeń opuszczających system i obliczenie statystyk wynikowych.

Rys. 8.12. Ikona i zawartość bloku *Dispose*

8.2.8. ŁĄCZENIE BLOKÓW SCHEMATU GRAFICZNEGO

Bloki schematu graficznego muszą być ze sobą połączone. Linie połączeń, tzw. *Connections*, determinują kolejność pokonywania bloków przez zgłoszenia.

Connections są tworzone za pomocą ikony  (*Connect*) wybieranej z głównego menu. Po kliknięciu na ikonę wskaźnik myszy zmienia się w krzyżyk. Należy teraz kliknąć na punkt wyjścia z bloku źródłowego (*exit point* ►), przeciągnąć myszkę i kliknąć na punkt wejścia bloku docelowego (*entry point* ■). Aby ułatwić poprawne wybranie myszką obu punktów, po wybraniu punktu początkowego podświetla się zielony kwadrat, a po wybraniu punktu końcowego – czerwony.

W przypadku konieczności wykonania wielu połączeń, po wprowadzeniu każdej kolejnej linii można kliknąć prawym klawiszem myszy na pustym miejscu ekranu i wybrać opcję *Repeat Last Action*, kontynuując w ten sposób łączenie bloków. Gdy połączeń do wykonania jest bardzo wiele, można na stałe włączyć opcję łączenia poprzez dwukrotne kliknięcie na ikonę *Connect*.

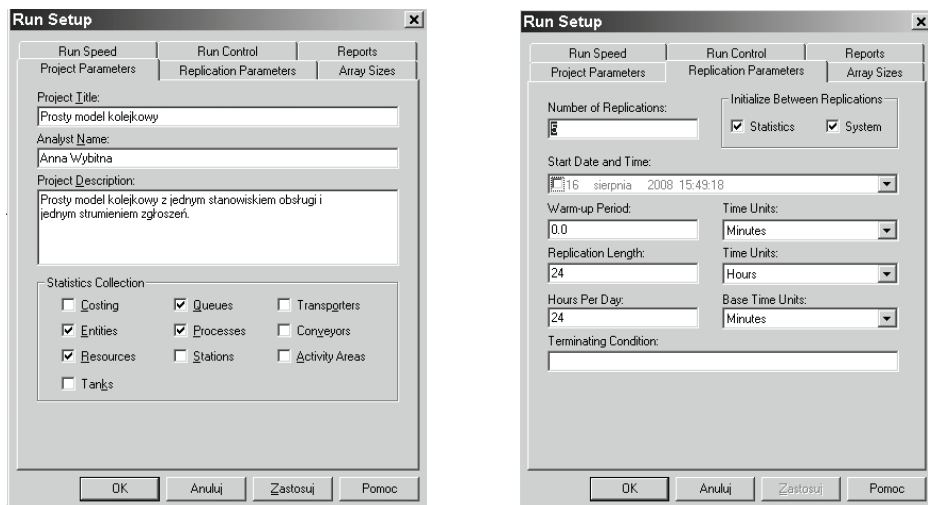
Jeżeli zaznaczona jest opcja *Object > Auto-Connect*, Arena automatycznie nanosi kolejne połączenia w kolejności umieszczania bloków na schemacie.

8.2.9. DEFINIOWANIE PARAMETRÓW EKSPERYMENTU SYMULACYJNEGO

Po zbudowaniu schematu graficznego i wypełnieniu wszystkich okien dialogowych, należy określić parametry eksperymentu symulacyjnego. Poprzez opcję *Run > Setup* i zakładkę *Project Parameters* możemy zapisać ogólne informacje o modelu i wybrać zestawy statystyk, które będą automatycznie gromadzone przez Arenę (rys. 8.13, lewy).

W zakładce *Replication Parameters* (rys. 8.13, prawy) ustalamy: liczbę powtórzeń (*Number of Replications*) jako 5, okres rozgrzewania modelu (*Warm-up Period*) równy 0.0, długość powtórzenia (*Replication Length*) równą 24 hours, długość doby w godzinach (*Hours Per Day*) jako 24 i bazowe jednostki symulacji (*Base Time Units*) jako minutes. Bazowe jednostki symulacji określają, w jakich jednostkach czasu będą obliczane i prezentowane w raportach końcowych statystyki wynikowe. Budując model, możemy podawać w różnych jednostkach wartości od-

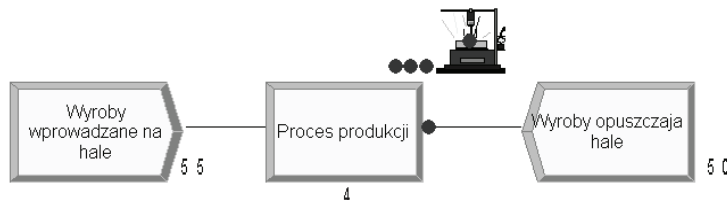
noszące się do czasu. Arena automatycznie przeliczy je właśnie na jednostki bazowe i w tych jednostkach zaprezentuje w raportach wyniki końcowe.



Rys. 8.13. Definiowanie parametrów eksperymentu symulacyjnego

8.2.10. URUCHAMIANIE SYMULACJI

Uruchomienie symulacji następuje poprzez wybranie ikony *Go* (▶) z menu głównego (lub opcji *Run > Go*). Następuje wtedy automatyczne sprawdzenie modelu przez Arenę i – po ewentualnej poprawie błędów przez użytkownika – uruchomienie modelu. Animacja symulowanego procesu prezentowana jest na ekranie (rys. 8.14), na którym widać przemieszczające się zgłoszenia (czarne piłeczki) i kolejkę tworzącą się nad blokiem *Process*. Liczby wyświetlane przy każdym bloku pokazują na bieżąco wartości różnych liczników. Dla modułu *Create* jest to liczba zgłoszeń, które opuściły model od początku powtórzenia, dla modułu *Process* – liczba zgłoszeń aktualnie przebywających w bloku (czyli długość kolejki plus liczba obsługiwanych zgłoszeń), dla modułu *Dispose* – liczba zgłoszeń, które opuściły system.



Rys. 8.14. Animacja modelu

Tempem symulacji można sterować za pomocą suwaka umieszczonego w górnym menu (rys. 8.15, lewy), natomiast na dolnym pasku ekranu można odczytać numer

aktualnie wykonywanego powtórzenia (rys. 8.15, prawy) oraz aktualne wskazanie zegara symulacji.



Rys. 8.15. Suwak sterowania prędkością symulacji (lewy) i pasek stanu (prawy)

Po zakończeniu symulacji, okno z modelem pozostaje cały czas w trybie *Run*, co oznacza, że nie można go poddać edycji. Ta ostatnia jest możliwa dopiero po zatrzymaniu modelu za pomocą przycisku *End* (■), wybieranego z głównego menu.

8.2.11. ODCZYTYWANIE RAPORTÓW SYMULACJI

Po zakończeniu symulacji wyświetlane jest pytanie, czy chcemy obejrzeć wyniki przeprowadzonych eksperymentów. Po twierdzącej odpowiedzi, otwierane jest okno raportów (por. rys. 8.16). Z lewej strony ekranu znajduje się wykaz utworzonych raportów, takich jak *Category Overview*, *Category by Replications*, *Entities*, *Queues*, *Resources* i inne. Wybór jednego z nich otwiera odpowiednie okno w głównej części ekranu.

Simple Processing						
Values Across All Replications						
Replications:		5	Time Units:		Minutes	
Entity						
Time						
	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
VA Time						
Wyrob	3.2984	0,07	3.2314	3.3795	1.0542	5.9319
NVA Time						
Wyrob	0,00	0,00	0,00	0,00	0,00	0,00
Wait Time						
Wyrob	3.8963	1,03	2.5343	4.6864	0,00	35.6855
Transfer Time						
Wyrob	0,00	0,00	0,00	0,00	0,00	0,00
Other Time						
Wyrob	0,00	0,00	0,00	0,00	0,00	0,00
Total Time						
Wyrob	7.1957	1,02	5.8421	7.9372	1.1805	40.5662
Other						

Rys. 8.16. Fragment raportu z wynikami symulacji

Podstawowym raportem jest *Category Overview*, który zawiera wartości większości statystyk końcowych. Wygodnym sposobem przeglądania raportu jest wybór odpowiedniej pozycji rodzajowej z zestawienia typu „drzewko wyboru” znajdującego się w środkowej części ekranu. Wskazanie wybranej pozycji automatycznie otwiera okno z właściwymi statystykami.

Wartości podawane w raporcie *Category Overview* są uśrednionymi wynikami ze wszystkich wykonanych powtórzeń, dlatego znajdziemy tam takie pozycje jak *Average* (średnia), *Half Width* (połowa długości przedziału ufności), *Minimum Average* (minimalna średnia), *Maximum Average* (maksymalna średnia), *Minimum Value* (minimalna wartość), *Maximum Value* (maksymalna wartość).

Na rysunku 8.16 przedstawione są informacje dotyczące czasu pobytu wyrobów w systemie, z których możemy odczytać, że średni (z 5 powtórzeń) czas pobytu to 7,1975 minut, najdłuższy zaobserwowany czas pobytu wynosił 40,5662 minuty, a najkrótszy 1,805 minuty. Ponadto, z pięciu wartości średnich obliczonych dla każdego powtórzenia najmniejszą wartością średnią było 5,8421 minuty, a największą 7,9372 minuty.

Inne zestawienia, które można znaleźć w raporcie to:

Entity→*Time*→*Wait Time*→*Wyrob*: czas oczekiwania wyrobów we wszystkich kolejkach, przy czym w tej statystyce uwzględniane są tylko wyroby, które opuściły system.

Entity→*Other*→*WIP*→*Wyrob*: liczba wyrobów przebywających jednocześnie w systemie (WIP oznacza Work In Process).

Process→*Other*→*Number In*→*Proces produkcji*: liczba wyrobów, która weszła do bloku *Process*.

Process→*Other*→*Number Out*→*Proces produkcji*: liczba wyrobów, która opuściła blok *Process*.

Queue→*Time*→*Waiting Time*→*Proces produkcji*.*Queue*: czas oczekiwania w kolejce przez blokiem *Process*.

Queue→*Other*→*Number Waiting*→*Proces produkcji*.*Queue*: liczba zgłoszeń oczekujących w kolejce przed blokiem *Process*.

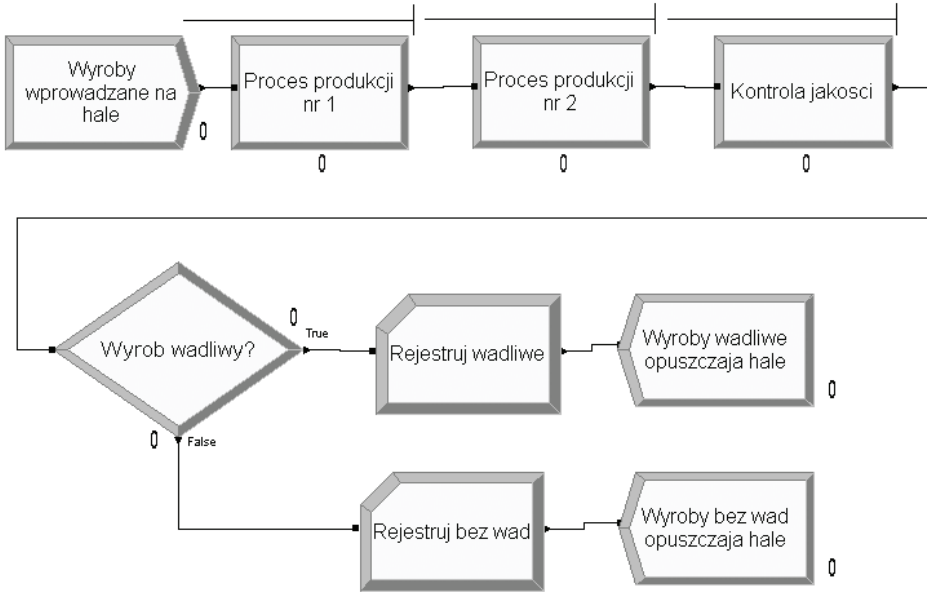
Resource→*Usage*→*Instantaneous Utilization*→*Maszyna*: Wykorzystanie stanowiska obsługi. Wartość 0,6847 oznacza, że stanowisko obsługi było wykorzystane przez 68,47% całego czasu symulacji.

8.3. PRZYKŁADOWY MODEL NR 2. PROCES PRODUKCJI Z TRZEMA STANOWISKAMI

8.3.1. OPIS PRZYKŁADU

Aby wyjaśnić działanie kolejnych trzech bloków panelu *Basic Process*, zmodyfikujemy przykład nr 1 z rozdziału 8.2.1.

Wyroby napływają na halę produkcyjną, tak jak poprzednio, w tempie opisanym rozkładem wykładniczym o średniej 5 minut. Po przybyciu kierowane są na pierwsze stanowisko (o nazwie *Maszyna 1*), gdzie poddawane są obróbce w czasie zgodnym

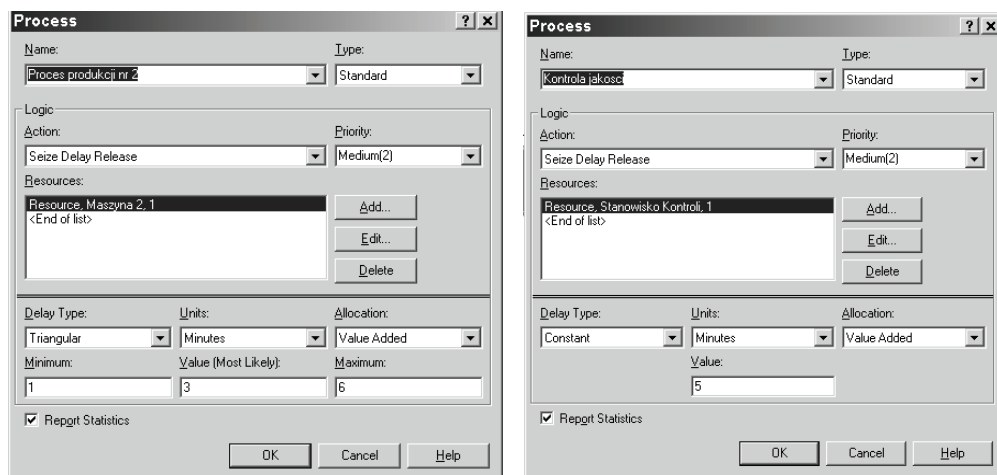


Rys. 8.17. Schemat przykładu nr 2

z rozkładem trójkątnym o parametrach (1, 3, 6) minut. Po opuszczeniu pierwszego stanowiska wszystkie wyroby kierowane są na drugie stanowisko (o nazwie *Maszyna 2*), gdzie poddawane są kolejnemu procesowi obróbki. Czas obsługi na stanowisku drugim opisany jest takim samym rozkładem jak dla stanowiska pierwszego. Po opuszczeniu drugiego stanowiska następuje kontrola jakości wyrobów, która trwa dokładnie 5 minut. Kontrola jakości wykonywana jest na specjalnie do tego celu wyznaczonym stanowisku (o nazwie *Stanowisko Kontroli*). 80% wyrobów pomyślnie przechodzi test. System opuszczają wszystkie wyroby bez względu na wynik testu, ale wyroby uszkodzone są znakowane. Co można powiedzieć o pracy systemu? Jak długie są kolejki, jaki jest czas oczekiwania, ile czasu wyroby przebywają w systemie?

Model dla przykładu nr 2 przedstawia rys. 8.17. Oprócz znanych nam bloków schematu *Create*, *Process* i *Dispose*, model zawiera dwa nowe bloki panelu *Basic Process*, tj. blok *Decide* i *Record* (trzeci nowy blok dodamy za chwilę).

Budowę modelu zaczynamy od znanego nam bloku *Create*, następnie umieszczamy pierwszy blok *Process*, nadając mu nazwę *Proces produkcji nr 1* i przydzielając mu jedno stanowisko obsługi o nazwie *Maszyna 1*. Następnie dodajemy drugi blok *Process* o nazwie *Proces produkcji nr 2*, który obsługiwany jest przez drugie (inne) stanowisko obsługi o nazwie *Maszyna 2* (por. rys. 8.18, lewy). Czas trwania obróbki jest na obu stanowiskach opisany rozkładem trójkątnym o tych samych parametrach. W kolejnym kroku dodajemy trzeci blok *Process* (rys. 8.18, prawy), który odpowiada kontroli jakości i realizowany jest na *Stanowisku Kontroli*. Czas trwania kontroli opisany jest za pomocą stałej wartości równej 5 minut.



Rys. 8.18. Okna dialogowe dla drugiego i trzeciego bloku *Process*

W ten sposób wprowadziliśmy do modelu trzy stanowiska obsługi, których definicje możemy obejrzeć w bloku danych *Resource* (rys. 8.19). Zauważmy, że automatycznie Arena utworzyła również trzy kolejki (por. rys. 8.20), ponieważ każdy z trzech bloków *Process* przewiduje zajmowanie/zwalnianie stanowisk obsługi.

Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	Maszyna 1	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
2	Maszyna 2	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
3	Stanowisko Kontrolni	Fixed Capacity	1	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Rys. 8.19. Zawartość modułu danych *Resource*

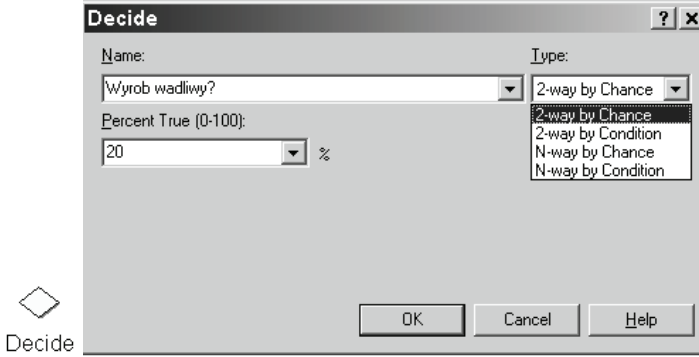
Queue - Basic Process				
	Name	Type	Shared	Report Statistics
1	Proces produkcji nr 1.Queue	First In First Out	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	Proces produkcji nr 2.Queue	First In First Out	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	Kontrola jakosci.Queue	First In First Out	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Rys. 8.20. Zawartość modułu danych *Queue*

8.3.2. BLOK SCHEMATU *DECIDE*

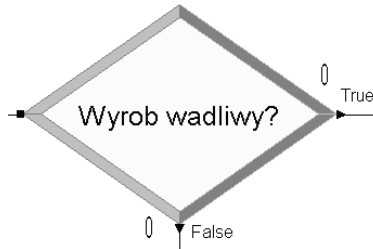
Blok *Decide* (rys. 8.21) pozwala podzielić strumień zgłoszeń na dowolną liczbę podstrumieni. Do wyboru mamy albo podział procentowy (*by Chance*), albo według zadanego warunku (*by Condition*). W przykładzie zastosujemy podział procentowy, ponieważ z opisu wynika, że około 80% wyrobów pomyślnie przechodzi test kontroli jakości, a 20% to wyroby wadliwe. Jeżeli strumień główny chcemy rozdzielić na dwa

podstrumienie (tak jak w naszym przykładzie), wybieramy *2-way by Chance* i w polu *Percent True* wpisujemy jedną z wartości: 20 lub 80. Nie ma znaczenia, którą wartość wpisujemy, musimy jednak pamiętać, że wybranej przez nas wartości będzie odpowiadał podstrumień opatrzony symbolem *True* (rys. 8.22) i dla tego strumienia musimy definiować bloki odpowiadające wpisanej wartości procentowej.



Rys. 8.21. Ikona i zawartość okna dialogowego bloku *Decide*

Gdybyśmy chcieli podzielić strumień główny na więcej podstrumieni (N), musieliśmy wybrać typ podziału jako *N-way by Chance* i przydzielić odpowiednie wartości procentowe $N - 1$ podstrumieniom (czyli przykładowo, dla czterech podstrumieni wartości procentowe przydzielamy trzem podstrumieniom). Do ostatniego podstrumienia kierowany będzie procent zgłoszeń uzupełniający całość do 100%.



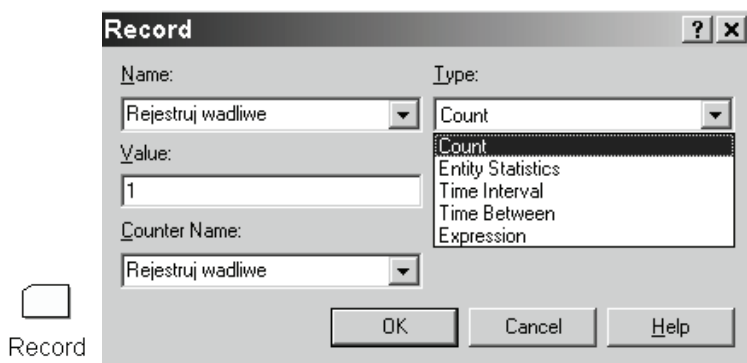
Rys. 8.22. Blok schematu *Decide*

8.3.3. BLOK SCHEMATU *RECORD*

Po zdefiniowaniu wszystkich operacji pozostaje nam już tylko wyprowadzenie wyrobów z systemu (por. rys. 8.17) za pomocą dwóch bloków *Dispose*, wcześniej jednak umieścimy na naszym schemacie dwa (nowe) bloki *Record* (rys. 8.23).

Arena automatycznie zbiera i umieszcza w raporcie końcowym bardzo dużo statystyk, jednak czasami będziemy chcieli zebrać jeszcze inne, dodatkowe dane. Jed-

nym ze sposobów na ich uzyskanie jest blok *Record*, który nie stanowi elementu modelowanego systemu, ale pełni ważną rolę w gromadzeniu danych o działaniu systemu.



Rys. 8.23. Ikona i okno dialogowe bloku schematu *Record*

Działanie bloku jest uzależnione od wyboru pozycji w polu *Type*. I tak:

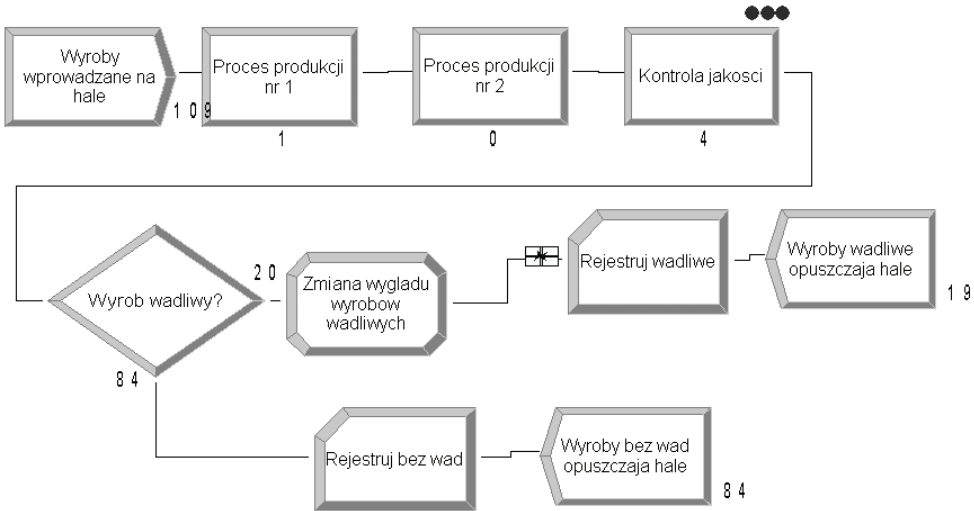
- *Count* zmniejszy lub zwiększy wartość licznika o nazwie jak w polu *Counter Name* o wskazaną wartość (jak w polu *Value*).
- *Entity Statistics* wygeneruje ogólne statystyki związane ze zgłoszeniem.
- *Time Interval* obliczy i zapamięta różnicę między wartością wskazanego atrybutu (zamiast pola *Value* pojawi się wtedy pole *Attribute Name*) a bieżącym czasem symulacji.
- *Time Between* wyznaczy i zapamięta czas między wejściem dwóch kolejnych zgłoszeń do modułu.
- *Expression* obliczy wartość podanego wyrażenia.

W naszym przykładzie bloki *Record* posłużą nam jako liczniki w celu określenia liczby wyrobów wadliwych i bez wad. Wartości obu liczników pozwolą nam zweryfikować działanie bloku *Decide*. Przypomnijmy, że około 80% wyrobów to wyroby pozbawione wad, a więc w naszym modelu takich właśnie wartości liczników powinniśmy oczekiwać.

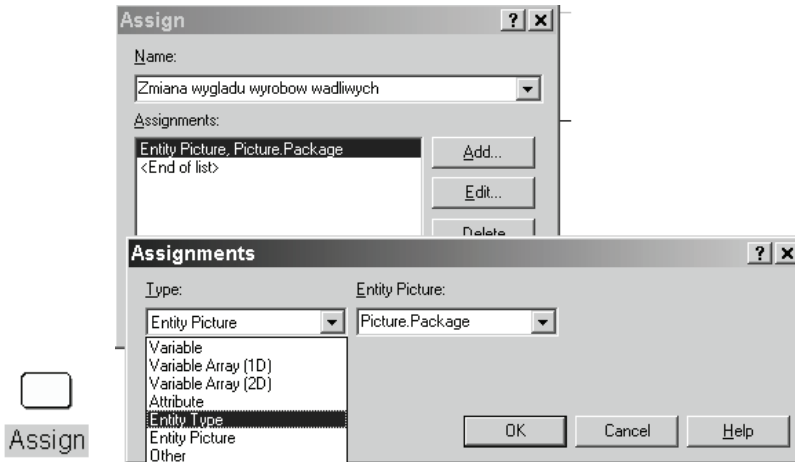
8.3.4. BLOK SCHEMATU *ASSIGN*

Trzecim nowym blokiem, który wprowadzimy do modelu, będzie blok *Assign*. Jest to blok, który odgrywa bardzo ważną rolę w procesie modelowania, ponieważ dzięki niemu możemy między innymi definiować atrybuty i nadawać wartości zmiennym w modelu. W przykładzie wykorzystamy blok *Assign* do tego, aby zmienić ikonkę wyrobów wadliwych (por. rys. 8.24), tak aby w trakcie symulacji można było rozróżnić obie grupy wyrobów. Być może w naszym prostym modelu nie jest to konieczne (wy-

roby wadliwe i te bez wad przemieszczają się wzdłuż odrębnych procesów), ale w bardziej złożonych modelach taki zabieg może nam pomóc w badaniu poprawności przebiegu symulacji.



Rys. 8.24. Zmodyfikowany model nr 2 – dodany blok *Assign* zmieniający ikonkę wyrobów wadliwych



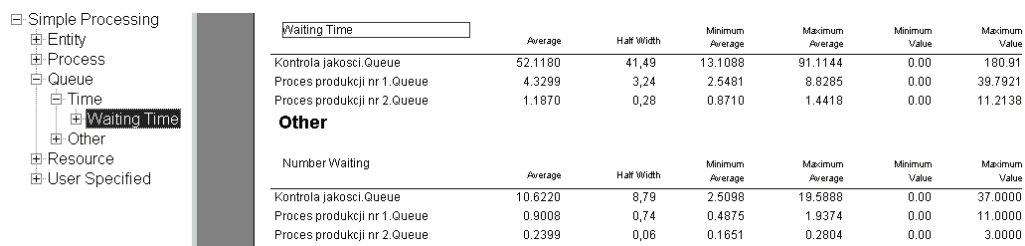
Rys. 8.25. Ikona i okno dialogowe bloku schematu *Assign*

Wprowadzenie bloku *Assign* (rys. 8.25) powoduje, że każde zgłoszenie, które przejdzie przez ten blok, doświadczy wszystkich zmian w nim zdefiniowanych. Zmiany (*Assignments*) wprowadzane są za pomocą przycisku *Add* i mogą dotyczyć (pole *Type*): nadawania wartości atrybutom (*Attribute*), nadawania wartości zmiennym (*Variable*), zmiany typu zgłoszenia (*Entity Type*), zmiany ikonki zgłoszenia (*Entity Picture*) i innych (*Other*).

Ponieważ chcemy zmienić wygląd wyrobów wadliwych, umieścimy blok *Assign* w odgałęzieniu, którym przemieszczają się wyroby z wadami, wybierzemy typ zmiany jako *Entity Picture* i wybierzemy dowolną z dostępnych ikonki (inną oczywiście od ikonki wybranej pierwotnie).

8.3.5. WYNIKI SYMULACJI DLA MODELU NR 2

Po uruchomieniu i przeprowadzeniu symulacji (parametry jak dla przykładu nr 1), odczytujemy wyniki. Okazuje się, że wąskim gardłem jest kontrola jakości, przed którą ustawia się długa kolejka (por. rys. 8.26), licząca średnio 10,622 wyrobów. Bardziej niepokojące jest jednak to, że maksymalna odnotowana długość kolejki wynosi 37 czekających wyrobów. Z długą kolejką związany jest również długi czas oczekiwania – wynosi on średnio 52,118 minut, a najdłuższy zaobserwowany czas to 180,91 minuty (czyli 3! godziny). Gdybyśmy sprawdzili dane dotyczące wykorzystania stanowiska kontroli, okazałoby się, że pracuje właściwie bezustannie, średnie bowiem wykorzystanie sięga 96%,



The screenshot shows the 'Waiting Time' report for the 'Kontrola jakosci.Queue' entity. The report is divided into two sections: 'Waiting Time' and 'Other'. The 'Waiting Time' section provides statistics for the queue, including Average (52.1180), Half Width (41.49), Minimum Average (13.1088), Maximum Average (91.1144), Minimum Value (0.00), and Maximum Value (180.91). The 'Other' section provides statistics for the number of items waiting, including Average (10.6220), Half Width (8.79), Minimum Average (2.5098), Maximum Average (19.5888), Minimum Value (0.00), and Maximum Value (37.0000).

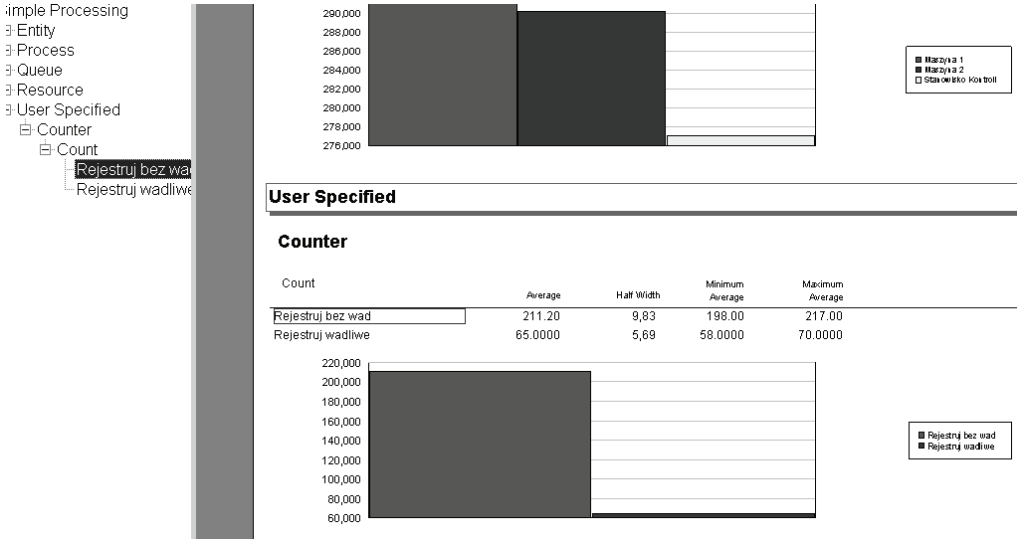
Waiting Time		Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Kontrola jakosci.Queue		52.1180	41.49	13.1088	91.1144	0.00	180.91
Proces produkcji nr 1.Queue		4.3299	3.24	2.5481	8.8295	0.00	39.7921
Proces produkcji nr 2.Queue		1.1870	0.28	0.8710	1.4418	0.00	11.2138
Other		Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Number Waiting							
Kontrola jakosci.Queue		10.6220	8.79	2.5098	19.5888	0.00	37.0000
Proces produkcji nr 1.Queue		0.9008	0.74	0.4875	1.9374	0.00	11.0000
Proces produkcji nr 2.Queue		0.2399	0.06	0.1651	0.2804	0.00	3.0000

Rys. 8.26. Przykładowe wyniki symulacji modelu nr 2

Wróćmy jeszcze na chwilę do dwóch zdefiniowanych w modelu procesów produkcyjnych. Pamięamy, że czas trwania obróbki w ramach jednego i drugiego procesu opisany był tym samym rozkładem (trójkątnym), o tych samych parametrach. Wydawać by się mogło, że w związku z tym przechodzenie wyrobów z jednego procesu na drugi powinno odbywać się płynnie i przy drugim stanowisku nie powinna tworzyć się kolejka. Tymczasem obserwujemy tworzenie się kolejki zarówno przed pierwszym, jak i przed drugim stanowiskiem. Oczywiście w modelu nie ma błędu – tworząca się kolejka jest wynikiem *losowych* wartości generowanych przez rozkład losowy, które nie muszą być (i raczej bardzo rzadko będą) identyczne.

Sprawdźmy jeszcze poprawność działania bloku *Decide*, odczytując wskazania dwóch liczników zdefiniowanych w blokach *Record*. Wartości liczników odczytamy pod nową kategorią *User Specified*, która automatycznie została dodana przez Arenę w raporcie *Category Overview* (rys. 8.27). Nasze liczniki zarejestrowały średnio 65 wyrobów wadliwych i 211,2 wyrobów bez wad. Wynika z tego, że wyroby wadliwe stanowiły około 23,53% wszystkich wyrobów. Czy rozbieżność między zadeklarowaną (20%) a uzyskaną wartością jest duża, czy mała? Pamiętajmy, że rozdział zgłoszeń

za pomocą bloku *Decide* odbywa się w sposób losowy. Im więcej powtórzeń wykonałobyśmy, tym bardziej zbliżylibyśmy się do zdefiniowanych 20% i uzyskalibyśmy węższy przedział ufności. Przykładowo, 100 powtórzeń dałoby nam wynik 19,6% wyrobów wadliwych, a połowa przedziału ufności (*Half width*) byłaby równa jedynie 1,5 (w przykładzie z 5 powtórzeniami *Half Width* wynosi 5,69).



Rys. 8.27. Wartości wynikowe liczników z bloków *Record*

8.3.6. HARMONOGRAM PRACY NA STANOWISKU *KONTROLI JAKOŚCI*

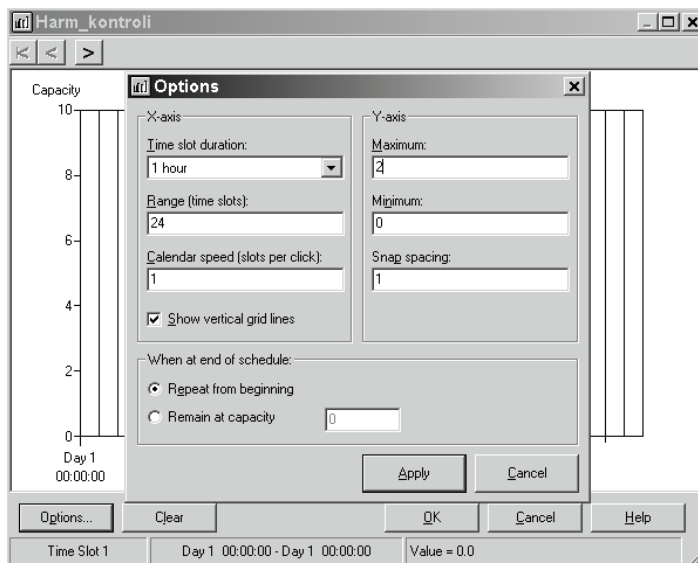
Analizując wyniki symulacji zauważyliśmy, że problemy z utrzymaniem płynności ruchu pojawiają się na stanowisku *Kontroli Jakości*. Załóżmy zatem, że liczba kontrolerów zmienia się w ciągu doby: na pierwszej zmianie pracuje jeden pracownik, na drugiej dwóch, na trzeciej ponownie jeden. Zmieniającą się w trakcie symulacji liczbę pracowników na stanowiskach obsługi możemy wprowadzić do modelu za pomocą harmonogramu pracy poprzez blok danych *Schedule*. Najpierw jednak musimy zmodyfikować blok danych *Resource*, zmieniając w wierszu *Stanowisko Kontrolni* pole *Type* na *Based on Schedule* i wprowadzając nazwę harmonogramu *Harm_Kontroli* w polu *Schedule Name* (por. rys. 8.28).

Resource - Basic Process											
	Name	Type	Capacity	Schedule Name	Schedule Rule	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	Maszyna 1	Fixed Capacity	1		Wait	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
2	Maszyna 2	Fixed Capacity	1		Wait	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>
3	Stanowisko Kontrolni	Based on Schedule	Harm_	Harm_Kontroli	Wait	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

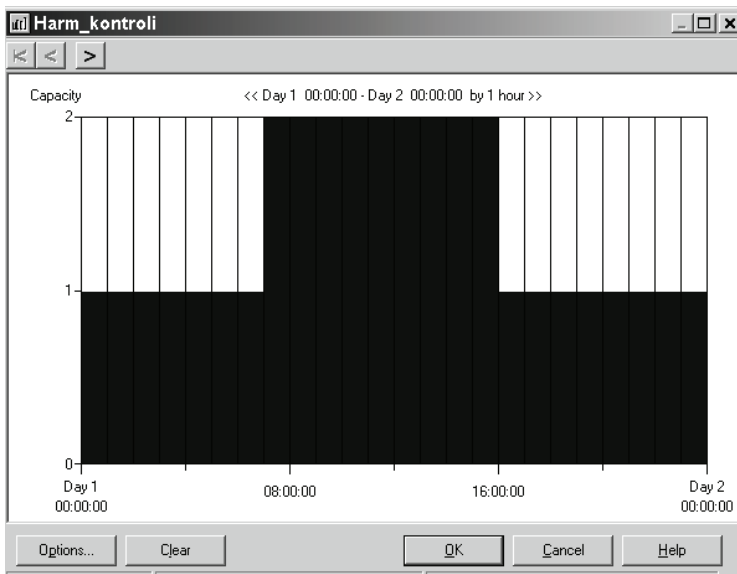
Rys. 8.28. Zmodyfikowana zawartość modułu danych *Resource*

Schedule - Basic Process						
	Name	Format Type	Type	Time Units	Scale Factor	Durations
1	Harm_kontroli	Duration	Capacity	Hours	1.0	0 rows

Rys. 8.29. Zawartość modułu danych *Schedule*



Rys. 8.30. Definiowanie opcji harmonogramu w bloku danych *Schedule*

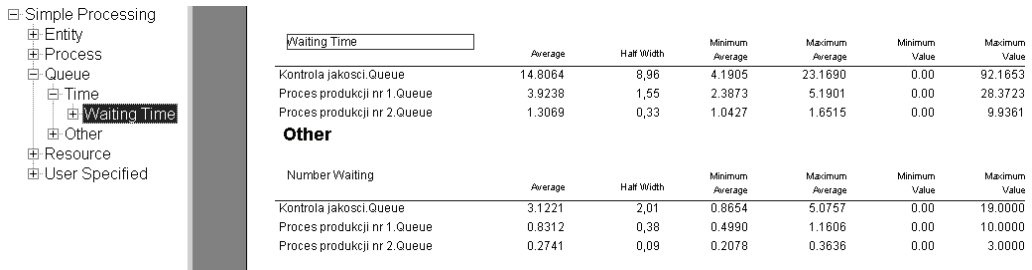


Rys. 8.31. Definiowanie harmonogramu w bloku danych *Schedule*

Następnie możemy już zdefiniować harmonogram. W bloku danych *Schedule* (por. rys. 8.29) klikamy na pole *Durations*. Zanim przystąpimy do zdefiniowania szczegółów, wybieramy przycisk *Options* (Opcje) znajdujący się w lewej dolnej stronie okna dialogowego i wprowadzamy następujące wartości parametrów (por. rys. 8.30): *Time slot durations* (kalibracja osi czasu) = 1 hour, *Range* (horyzont czasowy) = 24, *Maximum* (maksymalna wartość osi Y) = 2, *Minimum* (minimalna wartość osi Y) = 0.

Po wybraniu *Apply* (rys. 8.30) wprowadzamy szczegóły harmonogramu (rys. 8.31): zaznaczamy wartość 1 dla pierwszych 8 godzin, następnie 2 przez kolejne 8 godzin i znowu 1 na ostatniej ośmiogodzinnej zmianie. Akceptujemy harmonogram przyciskiem *OK*.

Ponownie uruchamiamy symulację i w wynikach obserwujemy znaczącą poprawę obsługi na *Stanowisku Kontroli* (por. rys. 8.32).



Waiting Time						
	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Kontrola jakości.Queue	14.8064	8,96	4.1905	23.1690	0.00	92.1653
Proces produkcji nr 1.Queue	3.9238	1,55	2.3873	5.1901	0.00	28.3723
Proces produkcji nr 2.Queue	1.3089	0,33	1.0427	1.6515	0.00	9.9361

Other						
Number Waiting						
	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Kontrola jakości.Queue	3.1221	2,01	0.8654	5.0757	0.00	19.0000
Proces produkcji nr 1.Queue	0.8312	0,38	0.4990	1.1606	0.00	10.0000
Proces produkcji nr 2.Queue	0.2741	0,09	0.2078	0.3636	0.00	3.0000

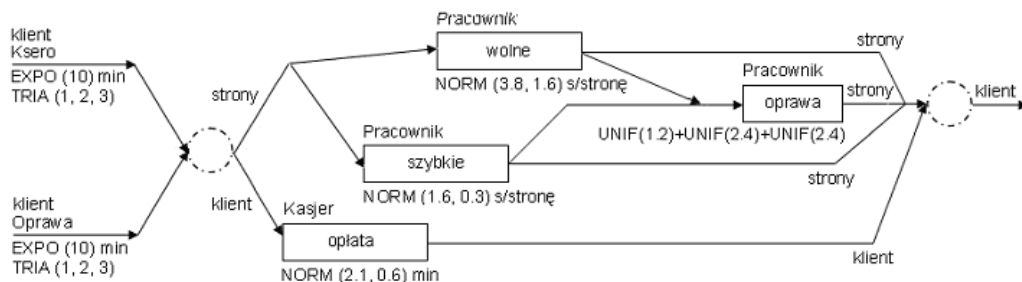
Rys. 8.32. Wyniki symulacji po wprowadzeniu zmian w harmonogramie pracy na stanowisku *Kontrola Jakości*

8.4. PRZYKŁADOWY MODEL NR 3. PUNKT KSEROGRAFICZNY

8.4.1. OPIS PRZYKŁADU

Za pomocą ostatniego, trzeciego przykładu zademonstrujemy działanie kolejnych dwóch bloków pobranych z panelu *Basic Process*, tj. bloków *Batch* i *Separate*, a także wykorzystamy wszystkie poznane wcześniej bloki do zbudowania bardziej złożonego modelu.

W punkcie kserograficznym (por. rys. 8.33) stoją dwie kopiarki (*Szybka* i *Wolna*) oraz jedna *Bindownica*. Urządzenia te obsługiwane są przez dwóch pracowników. Ponadto, w punkcie pracuje jeszcze trzeci pracownik, który zajmuje się wyłącznie pobieraniem opłat od klientów.



Rys. 8.33. Schemat działania punktu kserograficznego

Do punktu przychodzą klienci dwóch typów: *Klient Ksero* przychodzi zgodnie z rozkładem wykładniczym o średniej 10 minut i prosi o skopiowanie od 1 do 50 stron. *Klient Oprawa* przychodzi zgodnie z rozkładem wykładniczym o średniej 10 minut, prosi o skopiowanie od 1 do 50 stron, a następnie o oprawę. Klienci (obu typów) przychodzą grupowo po 1, 2 lub 3 osoby, ale najczęściej dwójkami. Pracownik, który jest w danej chwili wolny (jeden z dwóch obsługujących kserokopiarkę) przyjmuje zlecenie, a następnie przystępuje do pracy na wolnej lub szybkiej kopiarkie, kierując się następującymi zasadami:

(a) Jeżeli liczba stron do kopiowania jest mniejsza lub równa 30, to wybierana jest kopiarka wolna.

(b) Gdy liczba stron > 30 , wybierana jest kopiarka szybka, **z jednym wyjątkiem**: jeżeli kopiarka wolna aktualnie nie pracuje, a w kolejce do szybkiej kopiarki czeka dwa lub więcej zleceń, to pomimo dużej liczby stron do kopiowania wybierana jest kopiarka wolna.

Czas kopiowania na kopiarkie szybkiej można opisać rozkładem normalnym o średniej 1,6 sekundy i odchyleniu 0,3 sekundy na stronę; czas kopiowania na kopiarkie wolnej to rozkład normalny o parametrach (3,8; 1,6) sekundy na stronę.

Jeżeli zlecenie obejmuje również oprawę, skopiowane strony kierowane są na stanowisko oprawy. Proces oprawy składa się z trzech procesów jednostkowych:

Nr 1 – przygotowanie stron (jeden pracownik) wg rozkładu jednostajnego (1, 2) minuty,

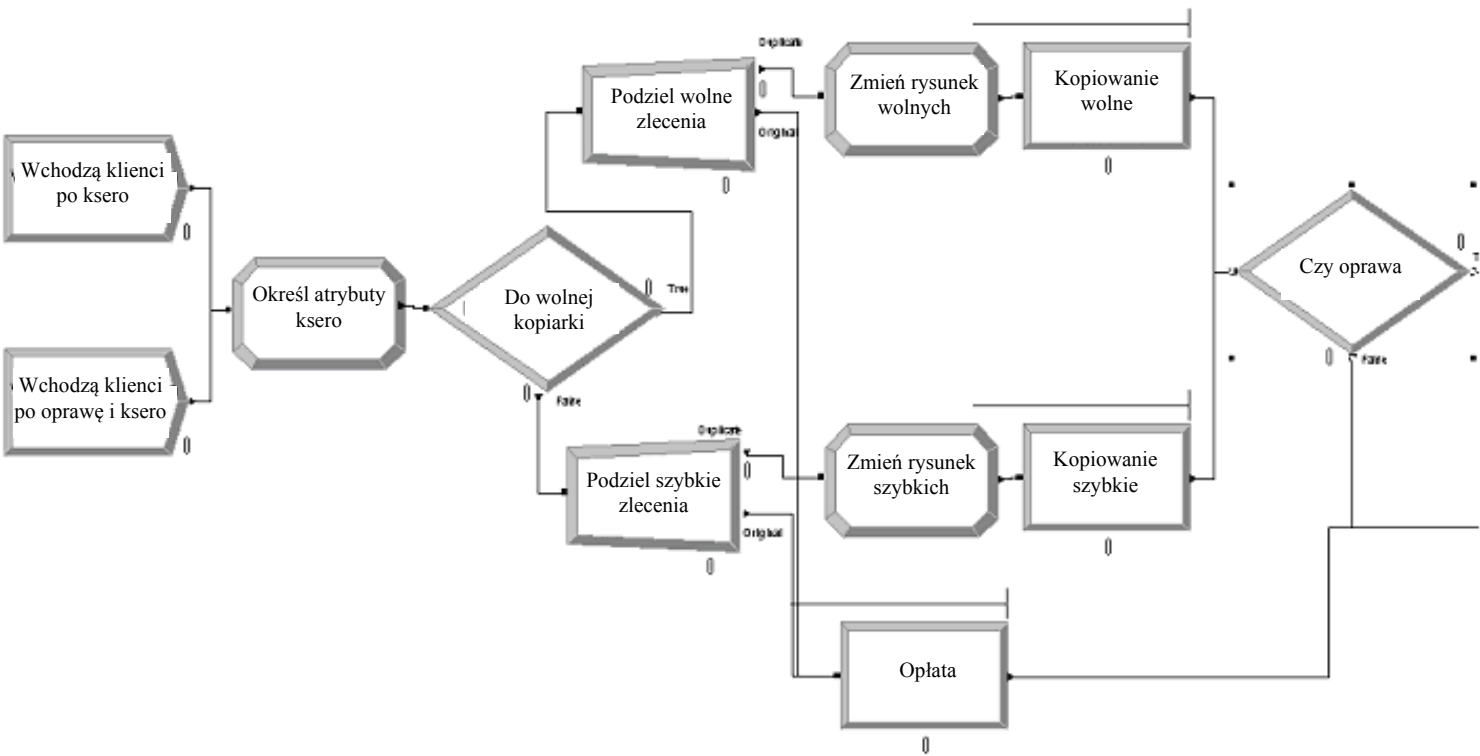
Nr 2 – oprawa właściwa (jeden pracownik+bindownica) wg rozkładu jednostajnego (2, 4) minuty,

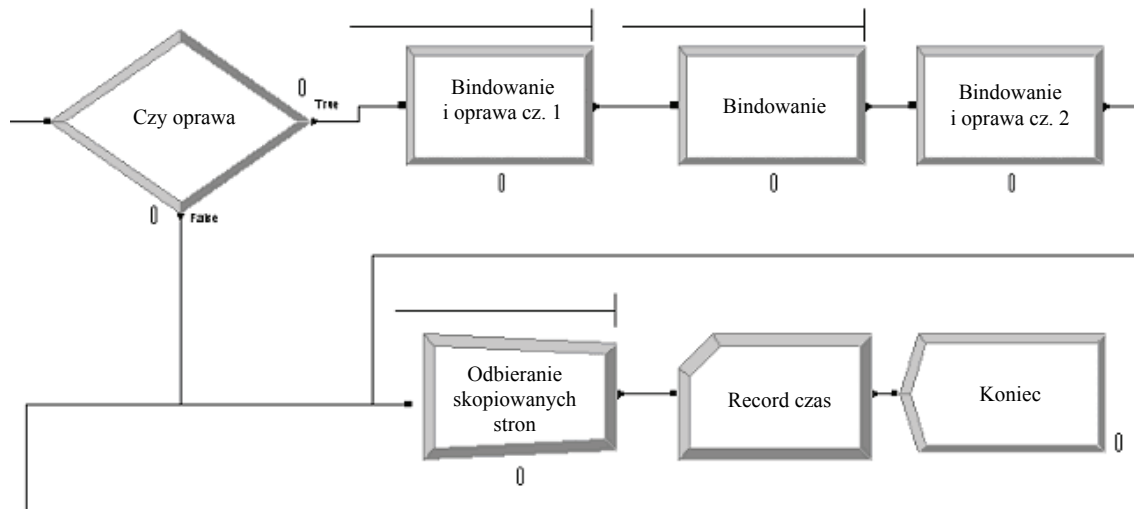
Nr 3 – zakończenie oprawy (jeden pracownik) wg rozkładu jednostajnego (2, 4) minuty.

Dwóch pracowników zajmujących się kopiowaniem i oprawą zawsze w pierwszej kolejności wybiera oprawę, a dopiero w drugiej kolejności – kserowanie. Trzeci pracownik (kasjer) pobiera opłatę w czasie opisanym rozkładem normalnym o parametrach (2,1; 0,6) minuty.

Przychodzący klient oddaje dokument do kserowania (i ewentualnie oprawy), a sam w tym czasie idzie zapłacić do kasy. Dopiero po opłaceniu może odebrać zlecenie.

Zdarzały się niedawno skargi klientów na zbyt długi czas oczekiwania w kolejce. Należy przeanalizować: czas pobytu klientów w systemie, czas oczekiwania we





Rys. 8.34. Model punktu kserograficznego

wszystkich kolejkach i oddzielnie w każdej kolejce, wykorzystanie maszyn i pracowników. Punkt jest czynny przez 10 godzin dziennie.

8.4.2. OGÓLNE ZAŁOŻENIA

Zauważmy, że każdy klient „uruchamia” dwa równoległe procesy: proces kopiowania i (ewentualnie) oprawiania oraz proces płacenia. Procesy te realizowane są jednocześnie, natomiast klient opuszcza punkt ksero, gdy oba procesy się zakończą. Wykorzystamy w modelu dwa nowe bloki: *BATCH* i *SEPARATE*. Dzięki tym blokom: (1) zamienimy jeden proces na dwa równoległe oraz (2) będziemy mieli pewność, że klient opuści punkt ksero dopiero wtedy, gdy oba procesy zostaną ukończone.

Blok *Assign* wykorzystamy do zdefiniowania dwóch atrybutów i do zmiany ikonki zgłoszenia, tak abyśmy obserwowali w trakcie symulacji poruszających się ludzi i (osobno) przemieszczające się kartki papieru.

Proces oprawy przedstawimy za pomocą trzech kolejnych bloków *Process*, w których będziemy zajmowali i zwalniali stanowiska obsługi.

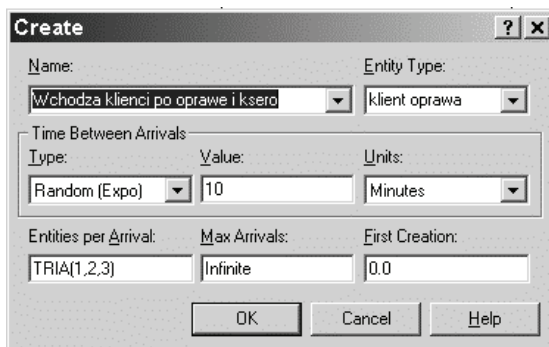
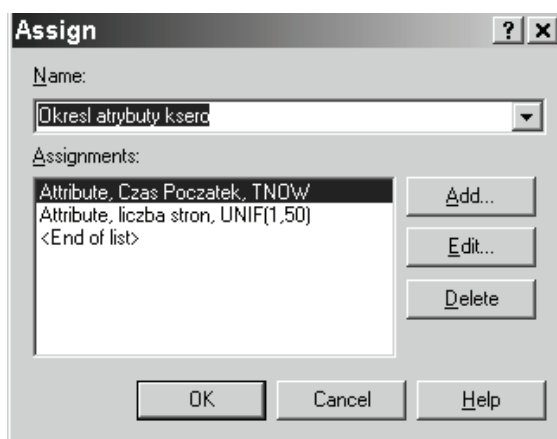
Blok *Record* zastosujemy tak jak w modelu nr 2 do zliczania, ale również wykorzystamy jego drugą funkcję – możliwość mierzenia czasu, jaki upłynął od wskazanej chwili do wybranego momentu pomiaru czasu.

8.4.3. BUDOWA MODELU, CZĘŚĆ PIERWSZA. BLOK *SEPARATE*

Gotowy model przedstawiony jest na rys. 8.34. Klienci przychodzą w dwóch różniących grupach, dlatego wprowadzamy do modelu dwa bloki *Create*, w których definiujemy dwie grupy klientów (klient ksero, klient oprawa), wprowadzamy dane dotyczące tempa przybywania klientów oraz definiujemy liczbę przychodzących osób (por. rys. 8.35). W opisie podano, że klienci przychodzą pojedynczo, dwójkami lub trójkami, ale najczęściej wchodzi dwie osoby. Ten opis odpowiada rozkładowi trójkątnemu o parametrach (1, 2, 3), który wywołany jest w Arenie jako funkcja *TRIA(1, 2, 3)*.

Po wprowadzeniu bloków *Create* i zdefiniowaniu dwóch typów zgłoszeń możemy za pomocą bloku danych *Entity* wybrać (różne) ikonki odpowiadające obu typom. Może to być na przykład obrazek kobiety (*Picture.Woman*) i mężczyzny (*Picture.Man*).

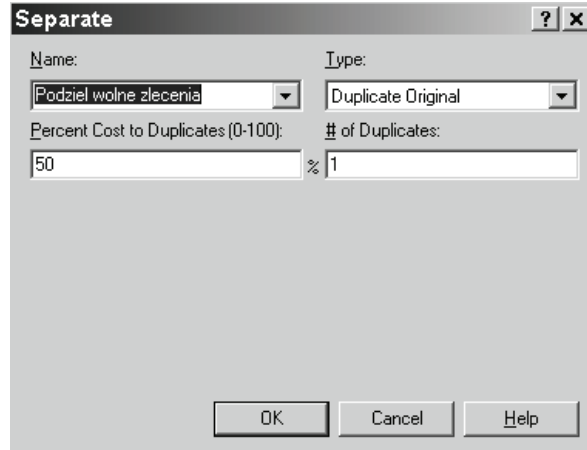
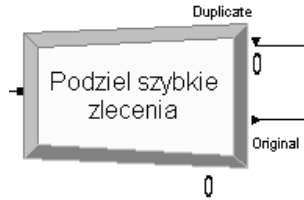
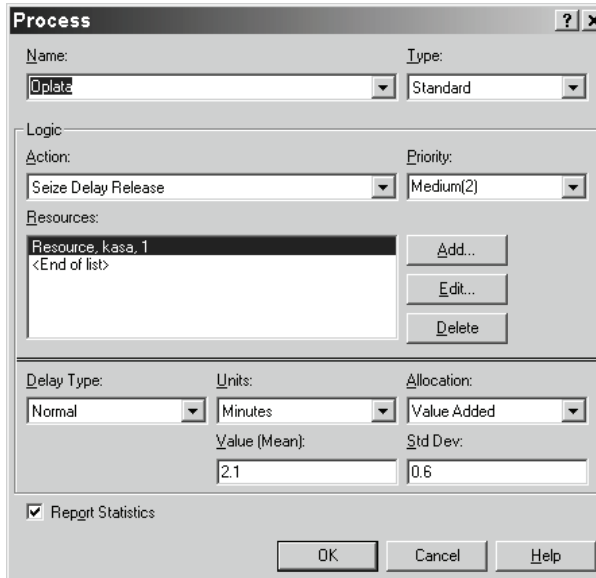
U wszystkich klientów, którzy przybyli do punktu kserograficznego, musimy zdefiniować dwa atrybuty o nazwach: *liczba stron* i *czas poczatek* (rys. 8.36). Pierwszy atrybut będzie nas informował, jaką liczbę stron klient zlecił do kserowania. Wartość ta opisana jest rozkładem jednostajnym, UNIF (1, 50). W drugim atrybucie zapamiętamy godzinę pojawienia się klienta w systemie i wykorzystamy później tę informację w bloku *Record*. Aby odczytać aktualną (w modelu oczywiście) godzinę, wystarczy sięgnąć po wewnętrzną zmienną Areny o nazwie *TNOW*. Zmienna ta jest bezparametrowa, a mamy do niej dostęp w każdej chwili i w każdym miejscu modelu.

Rys. 8.35. Zawartość jednego z dwóch bloków *Create*Rys. 8.36. Zawartość bloku *Assign*

Teraz wprowadzimy do modelu poznany wcześniej blok *Decide*, ale zdefiniujemy go dopiero po ukończeniu budowy pierwszej części modelu. W bloku *Decide* zamieścimy warunek logiczny, którego poprawna definicja wymaga wprowadzenia pozostałych bloków do modelu. Do bloku powrócimy zatem później, chwilowo nie wypełniając go (patrz podrozdz. 8.4.4).

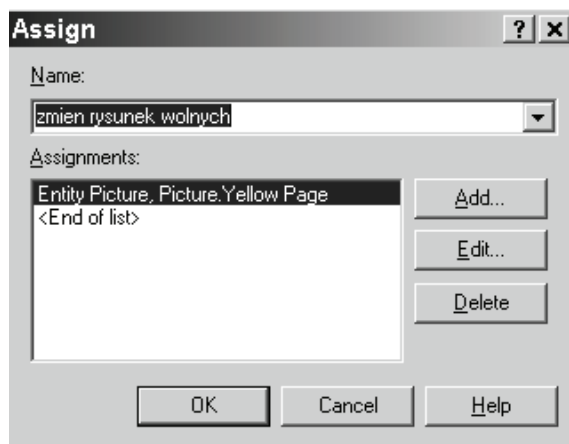
W kolejnym kroku, w modelu umieścimy nowy blok *Separate* (rys. 8.37). Jedną z funkcji, którą może pełnić ten blok, jest klonowanie zgłoszeń, które polega na tworzeniu dokładnych kopii (klonów) zgłoszeń oryginalnych (klon uzyskuje cechy identyczne z tymi, które miało zgłoszenie oryginalne). Zgłoszenie, które wpłynęło do bloku *Separate*, opuszcza go nie jako pojedyncze zgłoszenie, ale pod postacią kilku identycznych zgłoszeń.

Wprowadzamy do modelu dwa bloki *Separate*, wybierając w polu *Type* (rys. 8.37) opcję *Duplicate Original*, a w polu *# of Duplicates* (liczba klonów) wpisujemy 1. W ten sposób każde zgłoszenie opuści blok *Separate* razem z jednym klonem. Zauważmy, że na schemacie graficznym wyraźnie oznaczone jest wyjście z bloku dla oryginalnych zgłoszeń i wyjście dla klonów (rys. 8.38).

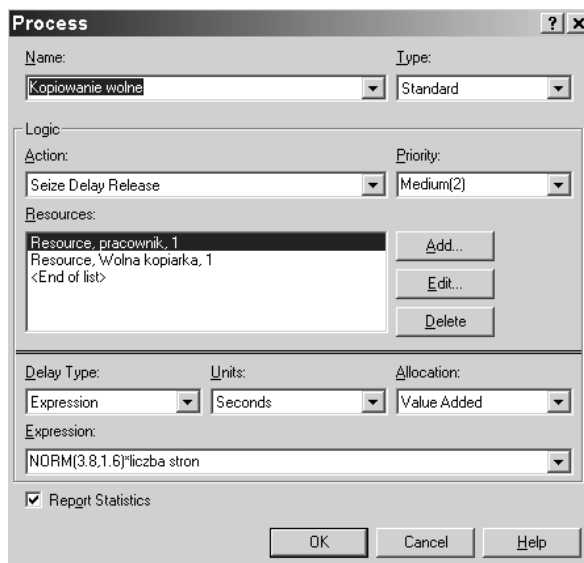
Rys. 8.37. Ikona i zawartość bloku *Separate*Rys. 8.38. Blok *Separate* z jednym wejściem i dwoma wyjściamiRys. 8.39. Blok *Process* – *Oplata*

Od tego momentu zgłoszenia przemieszczają się czterema ścieżkami. Dwie z nich (oryginały) przeznaczone są dla klientów, którzy kierowani są do bloku *Process* o nazwie *Oplata*, zdefiniowanego jak na rys. 8.39. Klienci płacą za zlecenie i kierują się do bloku *Batch*, w którym oczekują na odbiór swoich zleceń.

Dwie pozostałe ścieżki wyprowadzane z bloków *Separate* (wyjścia oznaczone jako *Duplicate*) to kartki papieru, które powinny być kserowane na kserokopiarkach. Tym zgłoszeniom zmieniamy wygląd za pomocą bloku *Assign* (rys. 8.40) i kierujemy do bloku *Process* o nazwie *Kopiowanie wolne* lub *Kopiowanie szybkie* (rys. 8.41).



Rys. 8.40. Zawartość bloku *Assign* zmieniającego wygląd zgłoszeń



Rys. 8.41. Blok *Process* – *Kopiowanie wolne*

Zauważmy, że w naszym punkcie kserograficznym *Pracownik* nie jest na stałe przypisany do stanowiska obsługi. Dwóch *Pracowników* obsługuje obie kserokopiarki (zamiennie), a także bindownicę, wykonując na życzenia klienta oprawę. Z tego względu, aby proces kserowania mógł się wykonać, zlecenie musi mieć zapewniony dostęp do dwóch stanowisk jednocześnie: do *Pracownika* i *Kopiarki*. Podobnie bindowanie jest możliwe tylko wtedy, gdy jednocześnie oba stanowiska, tj. *Pracownik* i *Bindownica*, są wolne.

Zwróćmy jeszcze uwagę na czas trwania procesu kserowania. W opisie zadania podano, że czas kopiowania jednej strony opisany jest rozkładem normalnym. Czas kserowania całego zlecenia musi natomiast uwzględniać liczbę kserowanych stron. Z tego względu nie wystarczy wybrać w polu *Delay Type* rozkład normalny. Wybieramy *Expression* i w dolnym polu wpisujemy z klawiatury $NORM(3.8, 1.6) * \text{liczba_stron}^8$ uważając, aby nazwa atrybutu była dokładnie taka, jaką wprowadziliśmy wcześniej w pierwszym bloku *Assign*.

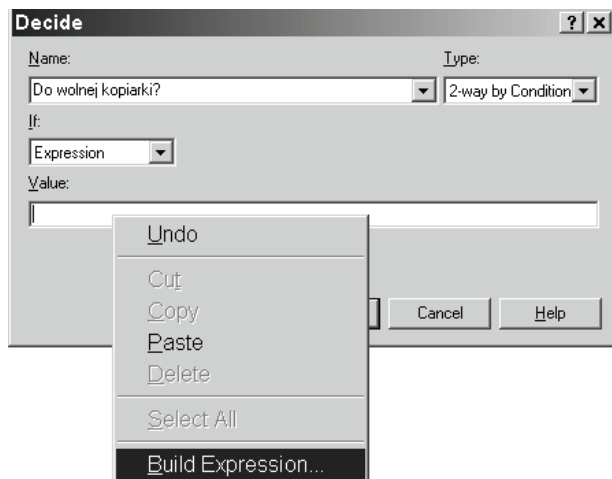
Na tym kończymy budowę pierwszej części modelu. W punkcie 8.4.5 zajmiemy się oprawą i odbiorem zleceń, wcześniej jednak (podrozdz. 8.4.4) powrócimy do niedokończonego bloku *Decide* i przy okazji wyjaśnimy działanie bardzo przydatnego narzędzia o nazwie *Expression Builder*.

8.4.4. EXPRESSION BUILDER

Teraz możemy już dokończyć definiowanie bloku *Decide*. W przykładzie nr 2, w bloku *Decide* zastosowaliśmy podział procentowy, tutaj jednak musimy wprowadzić do modelu warunek logiczny (rys. 8.42), bowiem zasada wyboru pomiędzy kopiarką wolną i szybką wymaga analizy warunków panujących aktualnie w modelu. W polu *Type* wybieramy zatem *2-way by Condition*, a w polu *If* wskazujemy na *Expression*.

W warunkach zadania podano, że wybór kopiarki zależy od liczby stron do kopiowania (więcej niż 30 stron kierowanych jest na kopiarkę szybką). Nie jest to jednak warunek rozstrzygający. Przypomnijmy: jeżeli kopiarka wolna aktualnie nie pracuje, a w kolejce do szybkiej kopiarki czeka dwa lub więcej zleceń, to pomimo dużej liczby stron do kopiowania wybierana jest kopiarka wolna. Aby zatem ułożyć warunek logiczny, musimy mieć dostęp do trzech informacji: liczby stron do kopiowania (w tym pomoże nam atrybut *Liczba stron*), stanu kopiarki wolnej (pracuje/nie pracuje), długości kolejki do kopiarki szybkiej. Aby uzyskać dwie ostatnie informacje, wywołałyśmy funkcje Areny, dostępne z poziomu modelu. Korzystanie z tych funkcji przypomina wywoływanie funkcji w Excelu: jest proste pod warunkiem, że wiemy, z jakiej funkcji i w jaki sposób skorzystać. W Excelu możemy przeglądać wszystkie dostępne funkcje, a w Arenie możemy skorzystać z narzędzia o nazwie *Expression Builder*.

⁸ W pakiecie Arena liczby rzeczywiste wpisujemy z kropką.



Rys. 8.42. Zawartość bloku *Decide*. Wywoływanie narzędzia *Expression Builder*

Zanim wpisujemy do modelu warunek logiczny za pomocą *Expression Builder*, zastanówmy się jak ten warunek powinien wyglądać. Przyjmijmy, że w przypadku spełnienia warunku zlecenie kierowane jest na kopiarkę wolną. Pierwsza część warunku logicznego powinna zatem sprawdzać liczbę stron:

Zlecenie zostanie skierowane na kopiarkę wolną, jeżeli: (Liczba stron <=30) .OR.

Operator logiczny *.OR.* (zapisany z kropkami) odpowiada funkcji *LUB()* w Excelu i zwraca wartość 0 (zero, czyli *FALSZ*), gdy wszystkie argumenty mają wartość *FALSZ*. Inne operatory logiczne wyszczególnione są w tabeli 8.1.

Tabela 8.1. Operatory logiczne

Opis	Zapis	Zapis alternatywny	Opis	Zapis	Zapis alternatywny
I	<i>.AND.</i>		Lub	<i>.OR.</i>	
Większy niż	<i>.GT.</i>	>	Większy lub równy	<i>.GE.</i>	>=
Mniejszy niż	<i>.LT.</i>	<	Mniejszy lub równy	<i>.LE.</i>	<=
Równy	<i>.EQ.</i>	==	Nierówny	<i>.NE.</i>	<>

W dalszej części (czyli za operatorem *.OR.*) wprowadzimy dwa pozostałe warunki:

((Liczba stron > 30) .AND. (NR(Wolna kopiarka) == 0) .AND. (NQ(Kopiowanie szybkie.Queue) >= 2)

Zlecenie zostanie skierowane na wolną kopiarkę również wtedy, gdy *Liczba stron* będzie duża (większa niż 30) i jednocześnie *Wolna kopiarka* będzie stała beczynnie (*NR(Wolna kopiarka) == 0*) i kolejka do kopiarki szybkiej będzie liczyła 2 lub więcej zleceń (*NQ(Kopiowanie szybkie.Queue) > = 2*).

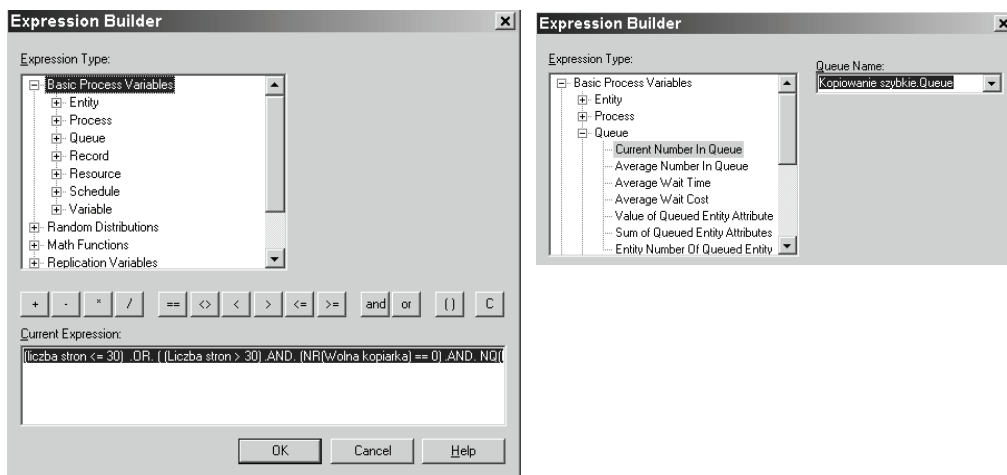
$NR(\text{stanowisko obsługi})$ to funkcja zwracająca wartość 0, gdy wszystkie kanały obsługi na *stanowisku obsługi* są dostępne (czyli *stanowisko* aktualnie nie obsługuje żadnych zgłoszeń).

$NQ(\text{Nazwa kolejki})$ zwraca liczbę zgłoszeń oczekujących we wskazanej *kolejce*.

Cały zapis będzie wyglądał następująco:

```
(Liczba stron <= 30).OR.((Liczba stron > 30).AND.(NR(Wolna kopiarka) == 0) .AND.(NQ(Kopiowanie szybkie.Queue) >=2))
```

Klikając prawym klawiszem myszy w polu *Value* (rys. 8.42), w którym można wpisać wartość numeryczną, wywołujemy narzędzie o nazwie *Expression Builder* (rys. 8.42). Wyświetla się okno dialogowe jak na rys. 8.43 (lewy). Wyrażenie konstruujemy, korzystając z funkcji wyświetlanych w górnym polu (*Expression Type*) i wybierając operatory ułożone nad polem do zapisu wyrażenia (*Current Expression*). Przykładowo, funkcję sprawdzającą długość kolejki znajdziemy w grupie *Basic Process Variable > Queue > Current Number In Queue* (rys. 8.43, prawy). Ponieważ w modelu zaplanowaliśmy wiele kolejek, musimy oczywiście wybrać tę właściwą (pole *Queue Name*).



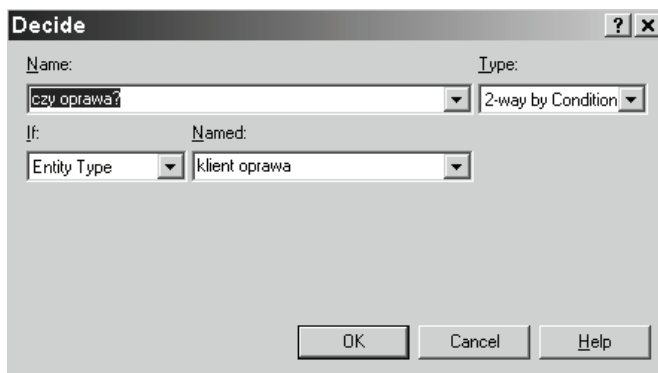
Rys. 8.43. Zawartość okna *Expression Builder*

Operatory logiczne możemy wpisywać z klawiatury albo wybierać z paska dostępnego w środkowej części okna dialogowego. Po zakończeniu wpisywania warunku i uważnym dobraniu liczby nawiasów nasz blok decyzyjny jest już w pełni gotowy.

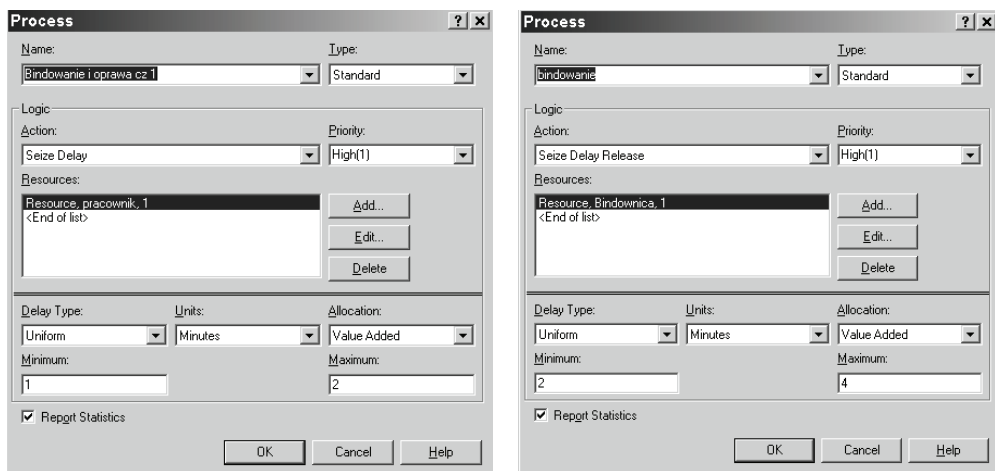
Narzędzie *Expression Builder* można wywoływać w dowolnym bloku modelu i z dowolnego pola numerycznego, czyli takiego, które wymaga wpisania liczby lub wyrażenia algebraicznego.

8.4.5. BUDOWA MODELU, CZĘŚĆ DRUGA. BLOK *BATCH*

W tym podrozdziale omówimy proces bindowania i odbioru przez klientów zrealizowanych zleceń. Po opuszczeniu kopiarek, zgłoszenia rozdzielane są ponownie na dwie grupy. Jedna grupa kierowana jest do wyjścia (przez blok *Batch*), by tam odnaleźć swojego właściciela, natomiast druga grupa musi być jeszcze oprawiona na kolejnym stanowisku. Jak rozróżnić zgłoszenia do oprawy od tych, które oprawy nie wymagają? Pomoże nam w tym typ zgłoszenia, który zdefiniowaliśmy na początku w blokach *Create*. Wystarczy, że sprawdzimy, jakiego typu jest zgłoszenie i będziemy wiedzieli, czy skierować go na oprawę, czy od razu do wyjścia. Badanie przeprowadzimy za pomocą bloku *Decide*, zdefiniowanego tak, jak to pokazano na rys. 8.44: wybieramy typ warunku jako *2-way by Condition* i sprawdzamy w polu *If*, czy typem zgłoszenia (*Entity Type*) jest *klient oprawa*.



Rys. 8.44. Sprawdzanie typu zgłoszenia w bloku *Decide*



Rys. 8.45. Zawartość dwóch kolejnych bloków modelujących proces oprawy

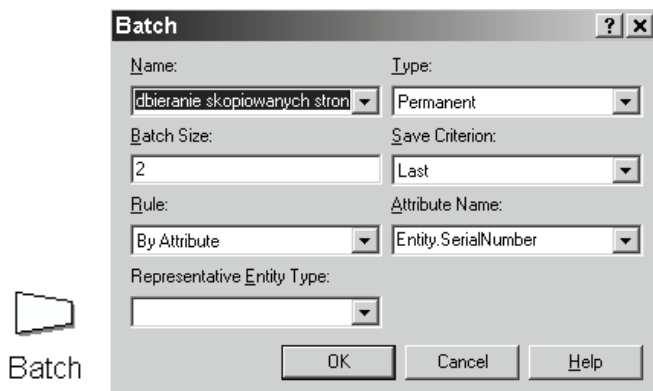
Zlecenie wymagające oprawy kierowane jest na trzy kolejno po sobie następujące bloki *Process*: Bindowanie i oprawa cz1, Bindowanie, Bindowanie i oprawa cz2. W pierwszym procesie potrzebny jest tylko pracownik, który przygotowuje strony do bindowania (składa, sprawdza kolejność, wyrównuje), dlatego w pierwszym bloku *Process* zajmowany jest *Pracownik*, który jednak nie jest zwalniany po skończeniu przygotowań (brakuje polecenia *Release*) (rys. 8.45, lewy). Rozpoczyna się drugi proces, w którym do *Pracownika* przydzielonego do zgłoszenia w pierwszym bloku dołącza *Bindownica* (rys. 8.45, prawy).

Po zakończeniu bindowania, *Bindownica* jest zwalniana, ale *Pracownik* ciągle jest przydzielony do zgłoszenia. Polecenia umieszczone w polu *Action* powodują zajęcie/zwolnienie tylko tych stanowisk, które zostały wskazane w oknie *Resources*. *Seize Delay* w pierwszym bloku *Process* spowodowało zajęcie *Pracownika*, a polecenie *Seize Delay Release* w drugim bloku *Process* spowodowało zajęcie i zwolnienie *Bindownicy*, natomiast nie miało nic wspólnego z *Pracownikiem*. *Pracownik* zwalniany jest dopiero w trzecim bloku *Process* (rys. 8.46), po ukończeniu oprawy (polecenie *Delay Release* w polu *Action*). Każdy z trzech procesów oprawiania jest realizowany w czasie opisanym właściwym rozkładem losowym (tu: jednostajnym *UNIF()*) o różnych parametrach.

Rys. 8.46. Zawartość bloków modelujących proces oprawy

Teraz trzy strumienie mogą spotkać się w bloku *Batch*: klienci z potwierdzeniem uiszczonej opłaty, zlecenia tylko po kserowaniu oraz zlecenia po kserowaniu i oprawie.

Moglibyśmy właściwie zakończyć w tym momencie model, ale zależy nam na tym, aby klienci odebrali własne zlecenie. Chcielibyśmy zatem z powrotem połączyć klienta ze zleceniem, czyli scalić oryginał z klonem. Posłużymy się blokiem *Batch*, którego rolą jest grupowanie (zbieranie) zgłoszeń (rys. 8.47).

Rys. 8.47. Ikona i okno dialogowe bloku *Batch*

W polu *Type* wybieramy *Permanent* (łączenie na stałe); wielkość grupy (*Batch Size*) ustalamy jako 2 (oryginał i klon); zasadę, według której oryginał i klon zostaną dobrane (*Rule*), ustalamy jako *By Attribute* (wg wartości wskazanego atrybutu). Atrybut, którego wartość będzie decydowała o dobraniu się klonu i oryginału to *Entity.SerialNumber*. Powiedzieliśmy, że po sklonowaniu blokiem *Separate*, kopia uzyskuje dokładnie te same cechy, co oryginał. Arena każdemu pojawiającemu się zgłoszeniu nadaje unikalny kolejny numer, który potem jest podstawą śledzenia zgłoszenia podczas przemieszczania przez model. Numer ten też został sklonowany i teraz umożliwi nam połączenie oryginału z kopią (pole *Attribute Name: Entity.SerialNumber*).

Przyjrzyjmy się jeszcze blokowi *Batch*. Nad blokiem może tworzyć się kolejka (o czym świadczy pozioma linia (por. rys. 8.48, lewy)). Dlaczego? Otóż może się zdarzyć (i najczęściej tak będzie), że albo klient będzie czekał na swoje zlecenie, albo zlecenie będzie czekało na klienta. Jeżeli uważnie przyglądnijemy się blokowi *Batch* w trakcie symulacji (rys. 8.48, prawy), zauważymy, że kolejka składa się z ikonek wszystkich typów zgłoszeń.

Rys. 8.48. Kolejka nad blokiem *Batch* w trakcie edycji (lewy) i w trakcie trwania symulacji (prawy)

Pozostał nam do omówienia jeszcze jeden blok. Przed blokiem *Dispose* umieszczamy blok *Record*, ale wykorzystamy go inaczej niż w poprzednim przykładzie. W polu *Type* wybieramy tym razem *Time Interval* (rys. 8.49), dzięki czemu będziemy mogli obliczyć czas, jaki upłynął od wskazanej chwili do chwili aktualnej (por. podrozdz. 8.3.3). Moment, od którego chcemy rozpocząć obliczanie, zapisany jest w atrybucie *Czas Początek*, któremu nadaliśmy wartość w bloku *Assign*. W poleceniu zadania mieliśmy określić średni czas pobytu zgłoszeń w systemie. Arena obliczy nam

automatycznie ten czas, zamieszczając wynik w raporcie, ale wynik będzie podany z podziałem na wszystkie typy zdefiniowanych zgłoszeń. Aby uzyskać wynik ogólny bez konieczności ręcznego przeliczania, posłużymy się właśnie opcją *Time Interval* w module *Record*. Ten sposób zastosowania bloku *Record* możemy wykorzystać do obliczenia dowolnych przedziałów czasu, np. od momentu opuszczenia stanowiska kserowania do momentu opuszczenia stanowiska oprawy. Musimy tylko odpowiednio rozłożyć w modelu bloki *Assign* i *Record*.

Rys. 8.49. Blok *Record* rejestrujący upływ czasu

Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	Wolna kopiarka	Fixed Capacity	1	15	15	2.5		0 rows	✓
2	Szybka kopiarka	Fixed Capacity	1	15	15	2.5		0 rows	✓
3	kasa	Fixed Capacity	1	20	20	0.0		0 rows	✓
4	Bindownica	Fixed Capacity	1	0.0	0.0	0.0		0 rows	✓
5	pracownik	Fixed Capacity	1	0.0	0.0	0.0		0 rows	✓

Rys. 8.50. Blok *Resource*

Queue - Basic Process				
	Name	Type	Shared	Report Statistics
1	Kopiowanie wolne.Queue	First In First Out	<input type="checkbox"/>	✓
2	Kopiowanie szybkie.Queue	First In First Out	<input type="checkbox"/>	✓
3	Oplata.Queue	First In First Out	<input type="checkbox"/>	✓
4	Odbieranie skopiowanych stron.Queue	First In First Out	<input type="checkbox"/>	✓
5	bindowanie.Queue	First In First Out	<input type="checkbox"/>	✓
6	Bindowanie i oprawa cz 1.Queue	First In First Out	<input type="checkbox"/>	✓

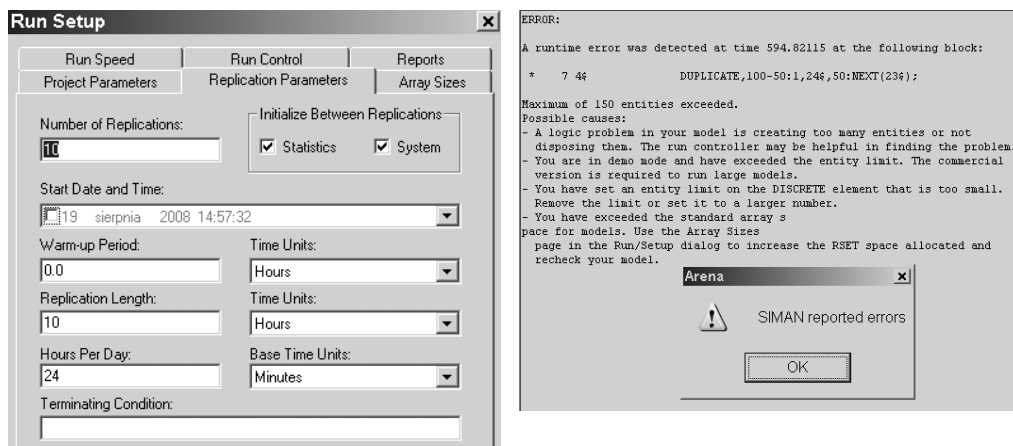
Rys. 8.51. Blok *Queue*

Na koniec sprawdzimy jeszcze zawartość bloków danych *Resource* i *Queue*. W naszym systemie pracuje 5 stanowisk obsługi (rys. 8.50) i może tworzyć się 6 kolejek (rys. 8.51). Zauważmy, że w trakcie procesu oprawy mogą tworzyć się dwie kolejki,

choć wprowadziliśmy trzy bloki *Process*. W bloku *Bindowanie i oprawa* cz2 nie przewidujemy zajmowania stanowiska, a zatem zatrzymanie zgłoszeń brakiem dostępu do stanowiska jest niemożliwe.

8.4.6. URUCHOMIENIE MODELU I ANALIZA WYNIKÓW

Parametry symulacji ustawiamy jak na rys. 8.52 (lewy) i wybieramy opcję *Go*. Niestety, po krótkiej chwili proces symulacji zostaje zatrzymany, a na ekranie wyświetla się komunikat o błędzie, jak na rys. 8.52 (prawy). Sygnalizowany błąd związany jest z przepełnieniem modelu i będzie pojawiał się zawsze (w wersji dydaktycznej *Arena*), kiedy zostanie przekroczony maksymalny limit liczby obiektów przemieszczających się w danej chwili przez model: *Maximum of 150 entities exceeded*.

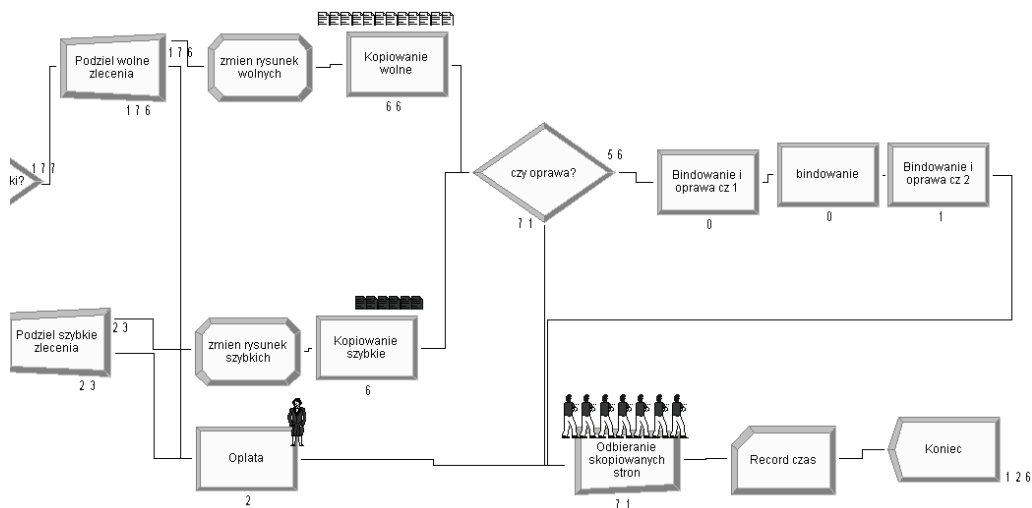


Rys. 8.52. Parametry eksperymentu symulacyjnego i komunikat o błędzie przepełnienia

Rzeczywiście, jeżeli popatrzymy na okno modelu (rys. 8.53) okaże się, że w kolejce do wolnej kserokopiarki czeka 66 zgłoszeń, a po odbiór zleceń oczekuje 71 klientów. Jeżeli dodamy do tego klientów i zgłoszenia znajdujące się w innych blokach modelu, okaże się, że nasz model jest przepełniony. Musimy znaleźć błąd, poprawić go i ponownie uruchomić symulację. Znalezienie błędu przepełnienia nie zawsze jest proste. Często przyczyna błędu wcale nie musi tkwić w module, w którym tworzy się długa kolejka. W naszym przypadku zapomnieliśmy o tym, że w systemie pracuje dwóch pracowników obsługujących kserokopiarki i bindownicę. Wracamy więc do modułu *Resource* (rys. 8.50) i w wierszu *Pracownik*, w polu *Capacity* poprawiamy wartość 1 na 2. Ponownie uruchamiamy model.

Pozostawiamy Czytelnikowi odnalezienie właściwych raportów. Ich analiza wykazuje, że wąskim gardłem jest wolna kserokopiarka, do której średni czas oczekiwania wynosi 7,49 minuty, ale zdarzają się przypadki, kiedy zlecenie czeka w kolejce blisko godzinę (51,71 min), co oczywiście przekłada się na czas oczekiwania klientów po odbiór zleceń (średnio 4 minuty, maksimum 47,6 minut). Za pomocą symulacji można

teraz próbować testować (1) zwiększenie liczby pracowników, (2) wymianę kserokopiarki, (3) zmianę sposobu obsługi zleceń. Analiza stopnia wykorzystania stanowisk wskaże jednak, że średnie wykorzystanie kserokopiarek jest niewielkie. Może zatem inaczej trzeba rozplanować ich pracę? Wydaje się, że jest to dobry moment, aby samodzielnie zgłębiać dalsze możliwości Areny. Odnajdziemy jeszcze tylko w kategorii *User Specified* „czas pobytu w systemie” naliczany blokiem *Record* (12,65 min).



Rys. 8.53. Uruchomienie modelu – długie kolejki

Przedstawione w rozdziale 8 przykłady prezentują bardzo niewielki zakres możliwości Areny. Omówiliśmy działanie większości bloków zamieszczonych tylko w jednym panelu spośród kilkunastu dostępnych. Objasnienia były ponadto ukierunkowane na trzy prezentowane modele i nie wyczerpywały wszystkich elementów i możliwości definicyjnych bloków. Pogłębioną wiedzę o środowisku *Arena* Czytelnik uzyska z takich podręczników, jak np. [Kelton i in. 2007], [Seila i in. 2003], [Altiok i in. 2007] i wielu innych.

PYTANIA SPRAWDZAJĄCE I ZADANIA

Rozdział 1. System, Model i Symulacja

1. W jakim celu prowadzi się badania symulacyjne?
2. Wyjaśnij, co jest istotą badań symulacyjnych?
3. Co odróżnia symulację od metod analitycznych badania systemów rzeczywistych?
4. Jaka jest różnica pomiędzy *symulacją statyczną* i *dynamiczną*?
5. Podaj główne zalety i wady symulacji.
6. Wymień główne podejścia symulacyjne.
7. Wymień etapy badania symulacyjnego.

Rozdział 2. Zbieranie i analiza danych wejściowych

1. Co oznacza sformułowanie *badanie poprawności dopasowania rozkładów losowych*?
2. W jakim celu, w badaniach symulacyjnych posługujemy się metodą reprezentacyjną?
3. Podaj różnice między schematami losowania: (a) indywidualne – zespołowe, (b) nieograniczone – warstwowe, (c) niezależne – zależne.
4. Podaj sposoby wykorzystania w modelu symulacyjnym zebranych danych wejściowych.
5. Na czym polega popełnienie błędu pierwszego rodzaju i błędu drugiego rodzaju w badaniach statystycznych?
6. Czy wprowadzenie do modelu symulacyjnego wartości średnich do opisu zmian o charakterze losowym jest właściwym rozwiązaniem? Dlaczego?

Rozdział 3. Generowanie liczb i zmiennych losowych

1. Za pomocą jakiego rozkładu, w modelach symulacyjnych systemów obsługi modeluje się strumienie zgłoszeń? Jakie muszą być spełnione warunki?
2. Wyjaśnij, w jaki sposób generuje się wartości zmiennej losowej dyskretnej?
3. Wyjaśnij, w jaki sposób generuje się wartości zmiennej losowej ciągłej?
4. Na czym polega metoda rozrzedzania zgłoszeń?

5. Do opisu jakich zjawisk rzeczywistych stosuje się w symulacji *stacjonarny proces Poissona*? Podaj przykład.
6. Na czym polega i do czego służy metoda *transformacji odwrotnej*?
7. Wyjaśnij, dlaczego wartości uzyskane za pomocą generatora liczb losowych noszą nazwę *liczb pseudolosowych*?
8. Jaka jest różnica między liczbą losową i zmienną losową?
9. Dlaczego zmienna losowa o rozkładzie równomiernym odgrywa kluczową rolę w modelu symulacyjnym? Jak uzyskujemy takie zmienne losowe?
10. W jaki sposób w modelach symulacyjnych systemów obsługi modeluje się strumienie zgłoszeń?
11. Do opisu jakiego procesu stosuje się w symulacji *niestacjonarny proces Poissona*? Podaj przykład.
12. Wyjaśnij, jaka jest różnica pomiędzy liczbami losowymi, a tymi, które uzyskujemy za pomocą generatora?
13. Kiedy stosujemy metodę rozrzedzania zgłoszeń?
14. Wygeneruj 200 niezależnych obserwacji z rozkładu normalnego o średniej 10 i odchyleniu 2. Policz 95-procentowy przedział ufności dla średniej. Narysuj histogram i porównaj go z wykresem funkcji gęstości rozkładu normalnego o parametrach (10,2).
15. Na czym polega pierwszy etap trójstopniowej analizy zasadności T. Neylora? Jakie techniki mogą być stosowane w ramach tego etapu?
16. Kiedy ciąg liczb można uznać za losowy?
17. Co oznacza *niestacjonarność* przybywania zgłoszeń do systemu?
18. Wykorzystaj poniższe liczby losowe do symulacji poprawności odpowiedzi na 10 pytań typu Prawda/Fałsz pewnego testu.
Liczby losowe: 0.52 0.06 0.50 0.88 0.53 0.30 0.10 0.47
0.99 0.37 0.66 0.91 0.35 0.32 0.00 0.84 0.57 0.00
a) Załóż, że liczby od 0.00 do 0.49 odpowiadają odpowiedzi *PRAWDA*, natomiast liczby z zakresu od 0.50 do 0.99 odpowiadają odpowiedzi *FALSZ*.
b) Załóż, że liczby parzyste odpowiadają odpowiedzi *PRAWDA*, natomiast liczby nieparzyste odpowiadają odpowiedzi *FALSZ*.

Rozdział 4. Symulacja zdarzeń dyskretnych

1. Dlaczego o modelach dyskretnych mówi się, że są to *modele kolejkowe*? Podaj 3 przykłady takich modeli.
2. Wymień znane Ci techniki symulacji dyskretniej.
3. Co to są zdarzenia bezwarunkowe i warunkowe? Podaj po jednym przykładzie.
4. Wyjaśnij pojęcie *zasobu* i *kalendacza zdarzeń* w symulacji dyskretniej.
5. Wyjaśnij pojęcie *zgłoszenia* i *zdarzenia* w symulacji dyskretniej.
6. Wymień reguły, według których zgłoszenia mogą być wybierane z kolejki.
7. Co to jest *kalendacz zdarzeń* i jak wpływa pobranie z niego zdarzenia na zegar symulacyjny?
8. Jak odbywa się sterowanie czasem symulacji w symulacji dyskretniej?
9. Co jest charakterystyczne dla prowadzenia symulacji metodą zdarzeniową?

10. Wyjaśnij, co oznaczają *wielokanałowe stanowiska obsługi*? Podaj dwa przykłady.
11. Jaką rolę pełni *kalendarz zdarzeń* w symulacji dyskretnej?
12. Co oznacza zwrot *przeskakiwanie czasu* w symulacji dyskretnej?
13. Wymień znane Ci techniki symulacji dyskretnej.
14. Rozważmy system obsługi *Stacja benzynowa*. Klientów obsługuje jeden pracownik. Przeprowadź ręczną symulację procesu obsługi klientów. Odstępy czasu między przybyciem kolejnych klientów są następujące: 4,7; 4,6; 6,9; 1,1;... Czas obsługi (operacje finansowe): 5,2; 2,5; 8,2; 6,8;... Analiza wyników zadania pozwala stwierdzić, że:
 - a) maksymalna długość kolejki to.....
 - b) wykorzystanie stanowiska to.....
 - c) średni czas pobytu klientów w kolejce to.....
 - d) średni czas pobytu w systemie to.....
15. Rozważmy system obsługi *Bankowe Okienko Kasowe*. Klientów obsługuje jeden pracownik. Przeprowadź ręczną symulację procesu obsługi klientów. Odstępy czasu pomiędzy przybyciem kolejnych klientów są następujące: 4,8; 4,6; 5,9; 1,1;... Czas obsługi (operacje finansowe): 5,0; 2,0; 11,2; 6,8;... Analiza wyników zadania pozwala stwierdzić, że:
 - a) maksymalna długość kolejki to.....
 - b) średni czas pobytu w kolejce to.....
 - c) średni czas pobytu w systemie to.....
 - d) wykorzystanie stanowiska obsługi to.....

Rozdział 5. Weryfikacja i walidacja modelu

1. Na czym polega i czego dotyczy trójstopniowa analiza zasadności T. Naylora?
2. Na czym polega pierwszy etap trójstopniowej analizy zasadności T. Naylora? Jakie techniki mogą być stosowane w ramach tego etapu?
3. Na czym polega drugi etap trójstopniowej analizy zasadności T. Naylora? Jakie techniki mogą być stosowane w ramach tego etapu?
4. Na czym polega trzeci etap trójstopniowej analizy zasadności T. Naylora? Jakie techniki mogą być stosowane w ramach tego etapu?
5. Jaki jest cel etapu *Weryfikacji i Walidacji* modelu symulacyjnego?
6. Co oznacza stwierdzenie „Wyników procesu symulacji nie należy odbierać w kategoriach *Prawda–Fałsz*”?
7. Na czym polega *Wielopłaszczyznowa Weryfikacja i Walidacja* modelu symulacyjnego?
8. Jaki jest cel etapu *Weryfikacji i Walidacji* modelu symulacyjnego?

Rozdział 6. Planowanie eksperymentów symulacyjnych i analiza wyników

1. Dlaczego w symulacji stochastycznej wykonuje się wiele powtórzeń?
2. Dlaczego w eksperymencie typu *terminating* nie uwzględniamy warunków początkowych?

3. Dlaczego w eksperymencie typu *terminating* nie wyznaczamy *okresu rozgrzewania* modelu?
4. Z którymi etapami procesu symulacji związane są *metody redukcji wariancji*? Wymień poznane cztery metody.
5. Na czym polega planowanie eksperymentu symulacyjnego?
6. Dlaczego w *steady-state simulations* definiujemy *okres rozgrzewania* się modelu? Jak wyznaczamy długość tego okresu?
7. W jakich typach eksperymentów symulacyjnych musimy uwzględnić warunki początkowe? Jak to robimy?
8. W jakich eksperymentach symulacyjnych definiujemy *okres rozgrzewania* się systemu? Jak wyznaczamy długość tego okresu?
9. W jaki sposób w stochastycznym modelu symulacyjnym oceniamy precyzję oszacowania wartości średnich?
10. Wyjaśnij krótko, co oznacza termin *eksperyment symulacyjny*?
11. W jakim celu stosuje się technikę *CRN*?
12. Na czym polega analiza 2^k czynnikowa planowania eksperymentów?
13. W jakim celu wyznacza się *przedział ufności dla średniej*?

Rozdział 7. Metoda Monte Carlo

1. Wymień etapy przebiegu symulacji Monte Carlo.
2. Dlaczego w metodzie Monte Carlo musimy wielokrotnie powtarzać eksperyment symulacyjny?
3. W jaki sposób można w arkuszu kalkulacyjnym *Excel* generować liczby losowe?
4. Do jakiej grupy podejść symulacyjnych można zaliczyć metodę Monte Carlo?
5. Tomek i Ania są przedstawicielami handlowymi. Miesięczne zarobki Tomka (X) układają się zgodnie z rozkładem równomiernym (w tys. zł) pomiędzy 1,0 i 5,0. Zarobki Ani (Y) można opisać rozkładem symetrycznym trójkątnym pomiędzy 1,0 i 5,0. Co miesiąc Tomek i Ania porównują zarobki. Jeżeli $X/Y < 0,3$ Tomek stawia obiad. Jeżeli $X/Y > 0,7$ obiad stawia Ania. W przeciwnym razie dzielą się rachunkiem po połowie. Należy zbudować model symulacyjny i sprawdzić, jak często Tomek kupuje Ani obiad w ciągu roku. Należy przeprowadzić 100 powtórzeń, z których każdy obejmuje 1 rok.
6. Wyjaśnij, w jaki sposób wprowadzisz do modelu Monte Carlo zmienną losową, za pomocą której będziemy symulowali *awaryjność drukarki*? Masz do dyspozycji dane zebrane w ostatnim roku:

Dni z awariami poważnymi	40
Dni z awariami drobnymi	90
Dni z usterkami	20
Dni bez awarii	210

7. Wymień kroki, za pomocą których wprowadzisz do modelu Monte Carlo zmienną losową symulującą popyt na pewien produkt? Masz do dyspozycji dane zebrane w kolejnych kwartałach ostatnich 5 lat:

Liczba kwartałów	Popyt
10	1000
5	800
2	500
3	600

8. Dealer Peugeota, uważa, że w 2007 roku popyt na samochody X będzie się kształtował na poziomie od 180 do 230 sztuk. Dealer otrzymuje jeden egzemplarz samochodu X za 25 000 zł, a sprzedaje go za 40 000 zł. Jeżeli nie wszystkie samochody zostaną sprzedane, to część niesprzedanych samochodów może zostać sprzedana w następnym roku po cenie promocyjnej 30 000 zł. Pozostałe samochody zwracane są producentowi. Za każdy zwrócony egzemplarz dealer dostaje 15 000 zł. Dealer rozważa zamówienie 180, 200, 220, 230, 240, 260 sztuk samochodów. Jak duże powinno być zamówienie? Należy zbudować model i przeprowadzić symulacje. Dla zaproponowanej wielkości zamówienia obliczyć prawdopodobieństwo zysku w wysokości co najmniej 4 000 000 zł. Wielkość sprzedaży promocyjnej w poprzednich latach: 30% z prawdopodobieństwem 0,5; 50% z prawdopodobieństwem 0,3; 20% z prawdopodobieństwem 0,2.
9. Biuro turystyczne Artemis uważa, że w 2007 roku popyt na wyjazdy do Egiptu będzie się kształtował na poziomie od 800 do 1200 miejsc w sezonie, ale najbardziej prawdopodobna wydaje się wielkość 1000. Biuro kupuje jedno miejsce za 1600 zł, a sprzedaje je za 2500 zł. Jeżeli nie wszystkie miejsca zostaną sprzedane, to część niesprzedanych miejsc może zostać sprzedana po cenie promocyjnej 1800 zł. Pozostałe miejsca są zwracane, ale biuro odzyskuje tylko połowę wpłaconej przez siebie kwoty. Biuro rozważa zamówienie 800, 900, 1000, 1050, 1100, 1150, 1200 miejsc. Jak duże powinno być zamówienie? Biuro chciałoby uwzględnić w symulacji koszt niezrealizowanego popytu w wysokości 900 zł za miejsce. Należy zbudować model, przeprowadzić 250 powtórzeń symulacyjnych, zaproponować wielkość zamówienia i oszacować ryzyko. Dla zaproponowanej wielkości zamówienia obliczyć prawdopodobieństwo zysku w wysokości co najmniej 800 000 zł. Wielkość sprzedaży promocyjnej w poprzednich latach (liczona od niesprzedanych miejsc): 20% z prawdopodobieństwem 0,25; 30% z prawdopodobieństwem 0,5; 40% z prawdopodobieństwem 0,1; 50% z prawdopodobieństwem 0,15.

Rozdział 8. Arena – środowisko symulacji dyskretnej

1. Ryzy papieru przybywają na halę co EXPO(10) minut w partiach (partie są jedno- dwu- i trzejelementowe, ale najczęściej występują partie złożone z trzech elementów). Około 1% ryz papieru jest zawracanych i opuszcza na-

- tychmiast halę. Na hali stoją dwie przycinarki. Około 60% ryz kierowanych jest do przycinarki nr 1, a reszta do przycinarki nr 2. Czas trwania czynności na przycinarce nr 1 to TRIA(7, 9, 10) minut. Czas trwania czynności na przycinarce nr 2 to TRIA(12, 13, 15) minut. Należy zbudować model i uruchomić go dla 5 dni (1 dzień = 8 godzin) i 10 powtórzeń. Jaki jest czas pobytu w systemie od momentu wejścia do momentu opuszczenia przycinarki (1 lub 2)? W jakim stopniu wykorzystane są obie przycinarki? Ile wyrobów kierowanych jest na przycinarkę 1 i 2?
2. Pacjenci przychodzą do przychodni na badania. Matki z dziećmi (1, 2 lub 3 dzieci, ale najczęściej 2) pojawiają się zgodnie z rozkładem wykładniczym o średniej 15 minut, a pacjenci dorośli zgodnie z rozkładem wykładniczym o średniej 10 minut. Wszyscy pacjenci udają się najpierw do rejestracji, w której pracują dwie osoby. Czas rejestracji to (3, 5, 8) minut w rozkładzie trójkątnym. Po opuszczeniu rejestracji 80% pacjentów kierowanych jest na badania ogólne, a 20% na badania specjalistyczne. Badania ogólne wykonywane są przez trzy pielęgniarki (jedna pielęgniarka wykonuje badanie u jednego pacjenta) w czasie TRIA(7, 8, 10) u osoby dorosłej i TRIA(10, 12, 15) minut u dziecka, natomiast badania specjalistyczne wykonuje lekarz wraz z (inną) pielęgniarką w osobnym pokoju i trwa to od 15 do 20 minut. Po wykonaniu badań specjalistycznych pacjent-dziecko musi pozostać na obserwacji (na korytarzu) jeszcze od 3 do 5 minut (Wskazówka – należy wykorzystać słowo kluczowe DELAY). Należy uruchomić symulację dla jednego 10-godzinnego dnia, wykonując 10 powtórzeń (bazowe jednostki = minuty). Jaki jest czas pobytu od momentu wejścia do momentu zakończenia badań specjalistycznych? Ilu pacjentów wykonuje badania ogólne? Jak długa jest kolejka do badań specjalistycznych?
 3. W banku znajduje się trzy stanowiska kasowe oraz dwa stanowiska obsługi klientów. Proces przychodzenia klientów do banku można opisać za pomocą rozkładu wykładniczego o średniej 2 minuty. Klienci przychodzą pojedynczo lub dwójkami. 5% klientów po wejściu rozmyśla się i opuszcza bank. Około 80% pozostałych klientów to klienci pragnący pobrać lub wpłacić gotówkę, a 20% to klienci kierujący się do stanowisk obsługi. Klienci po wejściu do banku ustawiają się w (jednej) kolejce do okienek kasowych lub w (innej) kolejce do stanowisk obsługi. Pobranie/wpłata gotówki zajmuje TRIA(5, 7, 9) minut, natomiast operacje przy okienku obsługi zajmują TRIA(10, 15, 17). Należy przeprowadzić symulację dla 7-godzinnego dnia pracy, wykonując 10 powtórzeń. Ilu klientów opuściło bank? Jak długa jest kolejka do kasy? Ile czasu trwa pobyt klientów wpłacających/wypłacających?
 4. Do baru typu Fast Food klienci przybywają pojedynczo według rozkładu wykładniczego o średniej 5 minut albo dwójkami według rozkładu wykładniczego o średniej 6 minut (wskazówka – proszę założyć, że to są dwa strumienie). Około 1% klientów samotnych i 15% par rezygnuje i opuszcza bar. Klienci najpierw zajmują miejsca w jadalni, w której jest 20 krzeseł (proszę założyć, że jadalnia

to wielokanałowe stanowisko obsługi). Zajmowanie miejsc w jadalni trwa 2 minuty. Po zajęciu stolików klienci idą kupić obiad. Klienci, którzy przyszli dwójkami rozdzielają się i kupują obiad oddzielnie. Czas „zamawiania” to rozkład trójkątny o parametrach 1, 2, 4 minuty, natomiast czas „płacenia” – trójkątny (1, 2, 3) minuty. Następnie klient wchodzi do jadalni i siada na wcześniej zajętych miejscach. Zjedzenie przepysznej potrawy zajmuje mu niezwykle przyjemne $TRIA(15,25, 35)$ minut. Później klient opuszcza bar (zwalniając miejsce). W barze pracuje 4 pracowników. Należy uruchomić symulację dla 10 powtórzeń po 7 godzin każde. Ilu klientów zrezygnowało z posiłku? Jak długa jest kolejka do stanowiska zamawiania/płacenia? Ile czasu upływa od przyścia do zakończenia posiłku?

LITERATURA

- Aczel A.D., *Statystyka w zarządzaniu*, Wydawnictwo Naukowe PWN, Warszawa 2005.
- Altioik T., Melamed B., *Simulation modeling and analysis with Arena*, Academic Press, Elsevier Inc65., 2007.
- Balcerak A., Pelech A., *Praprzodkowie symulacyjnych gier kierowniczych*, [w]: Symulacja Systemów Gospodarczych, Antałówka'99, WSZiF, Warszawa 1999.
- Balci O., *Principles and Techniques of Simulation Validation, Verification and Testing*, [w]: Proceedings of 1995 Winter Simulation Conference, IEEE Piscataway, New Jersey, s. 147–154, 1995.
- Balci O., *Validation, Verification and Testing Techniques throughout the Life Cycle of Simulation Study*, [w]: Proceedings of 1994 Winter Simulation Conference, Association of Computing Machinery, s. 215–220, 1994.
- Balci O., *Verification, Validation and Accreditation of Simulation Models*, [w]: Proceedings of 1997 Winter Simulation Conference, s. 135–141, 1997.
- Chyliński A., *Metoda Monte Carlo w bankowości*, Twigger S.A., Warszawa 1999.
- Evans J.R., Olson D.L., *Introduction to simulation and risk analysis*, Prentice Hall, 2002.
- Fishman G.S., Kiviat P.J., *The Statistics of Discrete Event Simulation*, [w:] Simulation 10, s. 185–195, 1968.
- Fishman G.S., *Symulacja komputerowa. Pojęcia i metody*, PWE, Warszawa 1981.
- Fishman G.S., *Monte Carlo. Concepts, Algorithms, and Applications*, Springer-Verlag, New York 1996.
- Friedrich D., Goldberg A., *A methodological framework for the validation of predictive simulations*, [w:] European Journal of Operational Research 124, s. 55–62, 2000.
- Gajda J.B., *Prognozowanie i symulacja a decyzje gospodarcze*, Wyd. C.H.Beck, Warszawa 2001.
- Gordon G., *Symulacja systemów*, WNT, Warszawa 1974.
- Greń J., *Statystyka matematyczna, Modele i Zadania*, PWN, Warszawa 1982.
- Grossman T.A., *Teachers' forum: spreadsheet modeling and simulation improves understanding of queues*, [w:] Interfaces 29(3); s. 88–103, 1999.
- Hora S.C., *Spreadsheet modeling of G/G/c queuing system without makros or add-ins*, [w:] INFORMS Transactions on Educations 3(3); s. 86–89, 2003.
- Harrington H.J., *Simulation modeling methods*, McGraw-Hill, Inc., 2000.
- Hillier F.S., Hillier M.S., Schmedders K., Stephens M., *Introduction to Management Science. A Modeling and case Studies Approach with Spreadsheets*, McGraw-Hill, Inc., 2008.
- Hoover S.V., Perry R.F., *Simulation. A problem solving approach*, Addison Wesley Publishing Company, Inc., 1989.
- Ingolfsson A., Grossman T.A.Jr., *Graphical spreadsheet simulation of queues*, [w:] INFORMS Transactions on Educations 2(2), s. 27–39, 2002.
- Jackson M., Staunton M., *Zaawansowane modele finansowe z wykorzystaniem Excela i VBA*, Helion, Gliwice 2004.
- Kelton W.D., Sadowski R.P., Sturrock D.T., *Simulation with Arena*, McGraw-Hill, Inc., 2004.
- Kelton W.D., Sadowski R.P., Sturrock D.T., *Simulation with Arena*, McGraw-Hill, Inc., 2007.

- Kleijnen J.P.C., *Statistical validation of simulation models*, [w:] European Journal of Operational Research, 87, s. 21–34, 1995.
- Kleindorfer G.B., Ganeshan R., *The philosophy of science and validation in simulation*, [w:] Proceedings of 1993 Winter Simulation Conference, s. 50–57, 1993.
- Krupa K., *Modelowanie symulacyjne i prognozowanie. Systemy ciągłe*, Wydawnictwa Naukowo-Techniczne, Warszawa 2008.
- Law A.M., Kelton W.D., *Simulation modeling and analysis*, McGraw-Hill, Inc., 1991.
- Law A.M., Kelton W.D., *Simulation modeling and analysis*, McGraw-Hill, Inc., 2000.
- Lawrence J.A., Pasternack B.A., *Applied Management Science*, John Wiley & Sons, Inc., 2002.
- Mielczarek B., *Dyskretne modelowanie symulacyjne systemów ochrony zdrowia*, [w:] Symulacja Systemów Gospodarczych, Instytut Organizacji i Zarządzania Politechniki Wrocławskiej, Wrocław 2002, s. 65–74, 2002.
- Mielczarek B., *Aspekty losowości w modelach symulacyjnych*, [w:] Symulacja Systemów Gospodarczych, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2005, s. 133–141.
- Mielczarek B., *Metoda Monte Carlo w nauczaniu symulacji – niesłusznie pomijane podejście?* [w:] Modelowanie symulacyjne systemów społecznych i gospodarczych. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, s. 11–20, 2006.
- Naylor T.H., *Modelowanie cyfrowe systemów ekonomicznych*, PWN, Warszawa 1974.
- Oakshott L., *Business modeling and simulation*, Pitman Publishing, 1997.
- Patrykiewicz A., *Wprowadzenie do metody Monte Carlo*, Wydawnictwo Uniwersytetu Marii Curie-Skłodowskiej, Lublin 1993.
- Pidd M., *Computer simulation in management science*, John Wiley & Sons, Ltd., 2004.
- Powell S.G., Baker K.R., *The art of modeling with spreadsheets*, John Wiley & Sons, Inc. 2004.
- Rao L., Owem L., Goldman D., *Development and application of a validation framework for traffic simulation models*, [w:] Proceedings of 1998 Winter Simulation Conference, s. 1079–1086, 2004.
- Render B., Stair R.M., Hanna M., *Quantitative Analysis for Management*, Pearson Education, Inc., Upper Saddle River, New Jersey 2003.
- Ross S.M., *Simulation*, Academic Press, An Elsevier Science Imprint, 2002.
- Sargent R.G., *Validation and verification of simulation models*, [w:] Proceedings of 2004 Winter Simulation Conference, s. 17–28, 2004.
- Seila A.F., Ceric V., Tadikamalla P., *Applied Simulation Modeling*, Thomson Brooks/Cole, 2003.
- Snopkowski R., *Symulacja stochastyczna*, Uczelniane Wydawnictwa Naukowo-Dydaktyczne, Kraków 2007.
- Steczkowski J., *Metoda reprezentacyjna w badaniach zjawisk ekonomiczno-społecznych*, PWN, Warszawa–Kraków 1995.
- Szapiro T. (red.), *Decyzje menedżerskie z Excelem*, PWE, Warszawa 2000.
- Uziako J., *Istotne kwestie etapu pozyskiwania i analizy danych wejściowych w procesie budowania modelu symulacyjnego*, [w:] Modelowanie symulacyjne systemów społecznych i gospodarczych. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2006, s. 21–32.
- Winston W.L., Albright S.Ch., *Practical management science. Spreadsheet modeling and applications*, Duxbury Press, Wadsworth Publishing Company, 1997.
- Winston W.L., *Microsoft Excel. Analiza i modelowanie danych*. APN Promise Sp. z o.o., Warszawa 2005.
- Wit R., *Metody Monte Carlo. Wykłady*, Wydawnictwa Politechniki Częstochowskiej, Częstochowa 2004.
- What is Scientific Sampling*. American Telephone and Telegraph Company, 1959.
- Zabawa J., Mielczarek B., *An attempt to comparison of inventory model implementations in Arena and Extend environments*, [w:] System Modelling Control. SMC. XII International conference, Institute of Mechatronics and Information Systems. Technical University of Lodz, 2007.
- Zeigler B.P., Praehofer H., Kim T.G., *Theory of Modeling and Simulation*, Academic Press, 2000.
- Zieliński R., *Metody Monte Carlo*, Wydawnictwa Naukowo-Techniczne, Warszawa 1970.

