SPIS TREŚCI

CZĘŚĆ PIERWSZA

CZĘŚĆ DRUGA

CZĘŚĆ PIERWSZA

Wstęp

Książka zawiera założenia teoretyczne i metodologiczne zastosowania modelowania komputerowego do badań lądowych systemów krajobrazowo-ekologicznych. Podobne aspekty, w zastosowaniu do środowisk wodnych, można znaleść w książce R. Z. Klekowskiego i V. V. Menshutkina (Klekowski, Menshutkin 1996). Poprawiona i rozszerzona wersja tej książki ukazała się w wydawnictwie Towarzystwa Naukowego Katolickiego Uniwersytetu Lubelskiego (Klekowski, Menshutkin 2002).

Różne podejścia do interpretacji krajobrazu, zaczynając od przedstawienia jego wyglądu, do analizy zachodzących w jego obrębie procesów ekologicznych (Schreiber 1990), do interpretacji krajobrazu jako heterogenicznej całości (Richling, Solon 1994, Zonneveld 1984), czy do pragmatycznej deliminacji elementów krajobrazu (Andrzejewski 2001) na poziomie regionalnym (np. województwa) czy lokalnym (np. gminy), nie przeszkadza na stosowanie modeli komputerowych do prognozowania stanu i dynamiki krajobrazu.

Modelowanie krajobrazu rozwija się, ostatnio coraz intensywniej, na pograniczu ekologii, geografii i informatyki. Stało się ono bardzo szeroko stosowaną metodą badań, zwłaszcza w Europie Zachodniej, uzupełniając różne projekty badawcze (Tenhunen et al. 2001).

Ekologia krajobrazu, znajdująca się w stadium gwałtownego rozwoju i poszukiwania najbardziej efektywnych metod badawczych, chętnie stosuje modelowanie matematyczne, w tym głównie techniki komputerowe, jako szczególnie zachęcający sposób badań naukowych. Niemniej jednak takie modelowanie nie może zastąpić metod tradycyjnych, na przykład fitosocjologicznych, gleboznawczych itp. Modelowanie ma natomiast tę przewagę, że potrafi powiązać wyniki obserwacji i eksperymentów w jedną całość w postaci dynamicznego systemu, za którego pomocą możliwe się staje optymalizowanie działalności człowieka i prognozowanie jej skutków w środowisku, także w obrębie krajobrazów.

Oczywiście książka nie ma zamiaru stanowić kompendium wiedzy o modelowaniu krajobrazu. Zamierzamy, za pomocą dobranej serii przykładów, wprowadzić Czytelnika w problematykę i metodologię modelowania krajobrazowego.

Pierwsza część książki zapoznaje z ogólną problematyką modelowania elementów krajobrazu. Przedstawiono metodologiczne podstawy modelowania, traktując krajobraz jako system, oraz analizując jego stan właściwości oraz charakter zmian. Oparta jest ona na przykładach następujących modeli:

(1) Modelu bilansowego części roślinnej ekosystemu lądowego,

(2) Modeli sukcesji zespołu leśnego, opartych na zasadzie modelowania osobniczego (tak zwane modele płatowe),

(3) Modelu krajobrazowego z zastosowaniem matematycznego aparatu automatów komórkowych.

Druga część książki opisuje technikę konstruowania modeli wybranych elementów krajobrazów ekologicznych. Jako narzędzie dla konstruowania takich modeli zastosowano język programowania *Visual Basic 6.0*.

Składowym elementem książki jest płyta CD, na której znajdują się programy modeli omawianych w tekście.

*

Pragniemy złożyć gorące podziękowania za inicjatywę przygotowania pracy oraz wszelką niesioną pomoc Dziekanowi Wydziału Matematyczno-Przyrodniczego KUL, księdzu profesorowi Stanisławowi Ziębie, oraz tym wszystkim, którzy z nami współpracowali, zwłaszcza w pracach terenowych. Składamy podziękowania asystentom Katedry Systemów Krajobrazowych KUL, magistrowi Marcinowi Jóźwinie oraz magistrowi Grzegorzowi Potaczale, za udział w pracach kameralnych oraz pomoc techniczną. Dziękujemy serdecznie panu redaktorowi Stanisławowi Sarkowi za trud doprowadzenia tej pracy do obecnego kształtu.

1. Metodologiczne podstawy modelowania

Głównym narzędziem analizy systemowej w jej zastosowaniach w ekologii krajobrazu jest modelowanie komputerowe elementów krajobrazów (Neveh 2000). Teoretyczną podstawą takiego podejścia jest ogólna teoria systemów zaproponowana przez L. von Bertalanffy'ego (1975). Teoria ta wchłonęła w siebie nie tylko osiągnięcia współczesnej biologii (zwłaszcza fizjologii i psychologii), lecz również filozoficzne idee całościowości systemów złożonych, idee matematyczne o systemach dynamicznych, jak również idee socjologiczne o zachowaniu się złożonych zespołów (rys. 1.1).

Modele stosowane w ekologii, a w szczególności w ekologii krajobrazu, są w różnym stopniu rzeczywiste, dokładne czy ogólne, wobec czego praktycznie niemożliwe jest jednoczesne osiągniecie w konkretnym modelu takiej samej, zwłaszcza wysokiej poprawności wszystkich trzech wymienionych parametrów. Dlatego na rys. 1.2 modele empiryczne, symulacyjne i analityczne znalazły się w różnych miejscach tego trójpodziału.

W postaci najogólniejszej jako s y s t e m rozumie się część wszechświata wyodrębnioną dobrze określonymi granicami od pozostałej jego części, która nazywa się o t o c z e n i e m (ś r o d o w i s k i e m z e w n ę t r z n y m). Środowisko zewnętrzne oddziałuje na system poprzez w e j ś c i a, a system oddziałuje na środowisko poprzez w y j ś c i a. W przypadku obecności takich wzajemnych oddziaływań system jest o t w a r t y, a przy ich braku – z a m k n i ę t y. Wszystkie systemy biologiczne, a więc również krajobrazowe, są systemami otwartymi. Zwróćmy uwagę, że klasyczna termodynamika jest teorią systemów zamkniętych, co bardzo ogranicza jej zastosowania do systemów ekologicznych.

Określenie granic systemu (delimitacja) jest działaniem w dużym stopniu dowolnym. Niemniej jednak nakreślenie takich granic dla krajobrazów jest sprawą szczególnie istotną, gdyż w przeciwieństwie do ekosystemów wodnych takie granice na lądzie są zazwyczaj mocno rozmyte. Tym istotnym zagadnieniem zajmowało się wielu autorów, na przykład D. L. Armand (1975), R. Andrzejewski (2001), M. D. Grodzynśkyji (1993).

Systemy przyrodnicze mają z reguły strukturę hierarchiczną. Oznacza to, że dany system jest częścią składową innego, znajdującego się na wyższym poziomie w hierarchii. Najbardziej holistyczną klasyfikację nauk ekologicznych zaproponował Z. Neveh (2000). Według tej klasyfikacji w systemach krajobrazowych można wyróżnić poziom osobniczy, stanowiący przedmiot autekologii; wyższym poziomem jest poziom populacyjny, czym zajmuje się demekologia. Populacje grupują się w zespoły, obiekty badań synekologii. Zespoły w połączeniu ze składnikami przyrody nieożywionej tworzą ekosystemy, a mozaika ekosystemów jest krajobrazem, stanowiącym obiekt ekologii krajobrazu, natomiast w połączeniu z układami społeczeństwa ludzkiego tworzą noosferę ziemi. To, co dotyczy współdziałania ze społeczeństwem ludzkim, należy do sfery ogólnej ekologii człowieka. Jak to obrazowo określa Z. Neveh (2000), ekologia krajobrazu jest pomostem łączącym ekologię systemów biologicznych ze społeczeństwem ludzkim.

Pojęcie systemu złożonego implikuje wyodrębnienie wewnątrz niego pewnych składników niższego rzędu. Przy braku takiego wyodrębnienia system określamy jako „czarną skrzynkę". Przykładowo składnikami fitocenozy mogą być poszczególne drzewa, a składnikami ekosystemu lądowego – części roślinna i glebowa, przy czym w części roślinnej można wyróżnić korony drzew, ich pnie i systemy korzeniowe, a w części glebowej – materię organiczną i nieorganiczną, jak również wodę gruntową (jak na przykład w pracy Hansen et al. 1990). Widzimy, że język ekologii krajobrazu jest bardziej urozmaicony niż podziały ekosystemowe: producenci – konsumenci – reducenci.

Zajmijmy się pojęciem stanu systemu. Definicja formalna tego pojęcia w teorii systemów jest dość skomplikowana, dlatego też będziemy traktować ją jako intuicyjnie zrozumiałą (podobnie jak chociażby pojęcie punktu w geometrii). Na przykład stan lasu sosnowego przed i po pożarze jest wyraźnie różny, podobnie jak stan krajobrazu w dolinie Wisły za czasów Jagiellonów i obecnie. Stan systemu może być opisany za pomocą różnych metod; przykładowo do opisu krajobrazu najczęściej stosuje się metodę kartograficzną, dla ekosystemu lądowego może to być opis przepływu potoków energii i materii, a dla populacji jeleni – liczebność osobników oraz struktura wiekowa i płciowa tej populacji.

System zmieniający swój stan z upływem czasu nazywa się systemem dynamicznym. Systemy nie zmieniające swego stanu, czyli systemy statyczne, nie mają praktycznie zastosowania do krajobrazów – nawet w stanie klimaksowym fitocenozy nie są systemami statycznymi; lepiej nazywać to stanem równowagi dynamicznej.

Opis zmian stanu systemu w czasie można przeprowadzić na wiele sposobów. Jeżeli rejestrujemy stan systemu w pewych odcinkach czasu, na przykład raz do roku lub raz na dobę, będzie to opis systemu w czasie dyskretnym (nieciągłym). Odcinek czasu między dwoma stanami nazywa się krokiem czasowym systemu. Jeżeli natomiast zakładamy, że zmiany systemu zachodzą nieprzerwanie, to znaczy odcinek czasu między dwoma rejestrowanymi stanami jest nieskończenie mały, jest to system z czasem ciągłym. W rzeczywistym, otaczającym nas świecie czas jest ciągły, a wobec tego wszystkie istniejące w tym świecie systemy też mają czas ciągły. Czasem nieciągłym, dyskretnym, posługujemy się jako wygodną abstrakcją. Niemniej jednak niektóre zjawiska naturalne, przebiegające w czasie ciągłym, mają charakter nieciągły, na przykład pobieranie pokarmu: pijawka może ssać krew tylko raz do roku.

Jeżeli system, przy niezmiennych wejściach (czyli warunkach zewnętrznych), zawsze przechodzi – po określonym czasie – z jednego określonego stanu do innego zawsze takiego samego stanu, to mamy do czynienia z systemem deterministycznym. Układ kierowania samochodem koniecznie musi być deterministyczny: po pokręceniu kierownicą w prawo samochód powinien skręcić w prawo.

Jeżeli system, przy niezmiennych wejściach, po określonym czasie może przejść do różnych stanów, wówczas system nazywany jest stochastycznym. Do jakiego konkretnie stanu przejdzie system, tego zawczasu nie można określić. Co najwyżej można oznaczyć prawdopodobieństwo wystąpienia właśnie takiego stanu. Niestety niekiedy nawet tego nie możemy określić (Ventzel 1980). Klasycznym przykładem systemu stochastycznego jest pogoda. W zasadzie wszystkie systemy naturalne są stochastyczne, a jedynie stopień stochastyczności jest bardzo zróżnicowany, jak ma to miejsce w ruchu planet sytemu słonecznego i przepływie wody w nieuregulowanej rzece. W pierwszym przykładzie stochastyczność można śmiało pominąć przy prognozowaniu dat wystąpienia zaćmień Księżyca i Słońca na wiele lat naprzód. W przykładzie rzeki jest to, z jednej strony, niedopuszczalne, a z drugiej – przy obecnym stanie wiedzy – nieuniknione. Dlatego prognozy powodzi na rok przyszły tak często się nie sprawdzają.

Takie bardzo skrótowe przedstawienie teorii systemów może się okazać niewystarczające dla Czytelnika. Dlatego polecamy do przeczytania dodatkową literaturę: Bertalanffy 1975; Bojarski 1984; Greniewski, Kempista 1963; Greniewski 1969; Greniewski 1969a; Mynarski et al. 1989; Szymański 1991.

A teraz, po bardzo skrótowym zajrzeniu do podstaw ogólnej teorii systemów, przejdźmy do sformułowania pojęcia modelu.

Wyobraźmy sobie dwa systemy, w których pewne właściwości pierwszego z nich (zwane istotnymi) występują także w drugim. W takim przypadku można uznać pierwszy system za oryginał, a drugi za model tego oryginału. Sam zaś proces przenoszenia właściwości jednego systemu do innego nazywamy modelowaniem. Na przykład modelem krajobrazu jest mapa lub fotografia satelitarna tego obszaru.

Pojęcie modelowania jest równie podstawowe jak pojęcie samego systemu. Ludzka świadomość jest niczym innym, jak zbudowaniem modelu otaczającego świata, przy czym jednym z elementów tego modelu jest własne, ludzkie „ja”. Być może człowiek stał się właśnie człowiekiem nie wtedy, gdy uczynił kij i kamień narzędziami pracy i walki, lecz wtedy gdy nauczył się modelować otaczający świat i swoje w tym świecie miejsce.

Właściwie cała istota i sens poznawczej i naukowej działalności ludzkiej polega na tworzeniu modeli różnorakich zjawisk przyrodniczych, technicznych i społecznych. Nauki opisujące dążą do tworzenia statycznych modeli werbalnych. W systematyce roślin lub zwierząt dominują opisy werbalne, uzupełniane rysunkami i fotografiami. Fizyka i astronomia, zwłaszcza w ich odmianach teoretycznych, posługują się aparatem matematycznym do modelowania oryginałów. Sama matematyka jest używana dla konstrukcji modeli w innych dziedzinach wiedzy (tak zwana matematyka stosowana) lub modeli obiektów abstrakcyjnych (czysta matematyka).

Szczególną rolę w modelowaniu odgrywa technika obliczeniowa. K o m p u t e r y pierwotnie służyły jako narzędzia jedynie do obliczeń, czyli do modelowania oryginałów matematycznych, na przykład rozwiązywania systemów równań, poszukiwania wielkości ekstremalnych lub obliczeń statystycznych. W ciągu pięćdziesięciu lat sfera zastosowania komputerów niewyobrażalnie się rozszerzyła. Objęła powszednie życie ludzi. Komputery stały się narzędziem modelowania najprzeróżniejszych przejawów działalności człowieka, a wśród nich także badania krajobrazów. To właśnie jest tematem naszej książki.

W definicji modelowania nie mówi się nic o tym, jakie mianowicie właściwości oryginału należy uznać za na tyle istotne, aby przenieść je do modelu. Nie istnieje wyczerpująca odpowiedź na to pytanie. Dobór właściwości istotnych zależy od celu modelowania, od możliwości technicznych, od poziomu i zakresu wiedzy o strukturze i funkcjonowaniu oryginału i wreszcie od osobowości badacza budującego model. Przy obecnym poziomie ludzkiej wiedzy konstruowanie modeli systemów złożonych (a systemy krajobrazowe niewątpliwie są złożone) jest w znacznym stopniu nie nauką, lecz sztuką (Oktaba 1976; Ventzel 1980).

Zacznijmy od c e l u m o d e l o w a n i a. Najczęstszym motywem konstruowania modelu jest p r o g n o z o w a n i e zachowania się złożonych systemów w przyszłości. Metodologia prognozowania za pomocą modelu polega na tym, że w odróżnieniu od oryginału, istniejącego w czasie realnym, model istnieje w swoim czasie modelowym. Czas ten – w odróżnieniu od realnego – może przyjmować wartości bliskiej lub odległej przyszłości. Przykładem takiego zastosowania modelu fitocenozy jest właśnie, opisana w rozdziale 3, prognoza stanu bukowych i świerkowych lasów w Karpatach.

Rys. 1.3 przedstawia jeden ze sposobów weryfikacji modelu, ze szczególnym uwzględnieniem jego przydatności prognostycznej.

Cykliczne zjawisko (może to być na przykład produkcja pierwotna zależna od światła i temperatury) jest obserwowane i mierzone przez n+4 lata. Model budujemy na podstawie n+3 lat, następnie uruchamiamy i sporządzamy „prognozę" na rok n+4, który też już minął i dla którego mamy takie same pełne dane. Na rysunku linia czerwona oznacza rzeczywisty przebieg zmierzonej zmiennej, a linia czarna – „prognozę". Szary rozszerzający się pasek oznacza przedział ufności (95%).

Weryfikacja wykazała dobrą przydatność modelu dla celów prognostycznych. To, że „prognoza" daje wartości wyższe od zmierzonych w roku n+4, zapewne oznacza, że średnie wartości zmiennej w ciągu lat n+3 były średnio wyższe niż w roku n+4.

Innym celem konstruowania modelu jest kierowanie modelowanym obiektem. W zastosowaniu do systemów krajobrazowych i ich składników takie zastosowanie modelowania jest w pełni uzasadnione, gdyż wszelkie oddziaływanie człowieka na systemy krajobrazowe ma, a przynajmniej powinno mieć, charakter optymalnego kierowania z określonym celem. Z uwagi na praktyczne znaczenie tego zagadnienia, zajmijmy się nim nieco dokładniej.

Kierowaniem nazywamy takie oddziaływanie na system, które dąży bądź do utrzymania jego stanu w określonych granicach, bądź do zminimalizowania albo maksymalizowania określonych funkcji systemu. W pierwszym wypadku mamy do czynienia z kierowaniem homeostatycznym, które jest właściwe dla obszarów chronionych i parków narodowych, w drugim zaś z kierowaniem optymalnym, które ma zastosowanie w leśnictwie lub rolnictwie, kiedy funkcją docelową jest ilość lub wartość rynkowa uzyskanego drewna lub produkcji rolnej.

Kierowanie homeostatyczne występuje szeroko w przyrodzie żywej. Na przykład regulacja biomasy lasu poprzez samozacienianie jest regulacją homeostatyczną z ujemnym sprzężeniem zwrotnym, mechanizm tego sprzężenia należy do funkcjonowania wszystkich systemów biologicznych oraz tworzonych przez ludzką technikę.

Kierowanie zawiera w sobie dwa odrębne elementy: kierujący i kierowany. W przypadku gospodarki leśnej elementem kierującym jest osoba fizyczna lub prawna, do której należy obiekt gospodarczy, a systemem kierowanym – obszar leśny lub cały leśny rejon, albo nawet większe terytorium, jak zlewnia, gdyż wyrąb lasu wpływa na jej system hydrologiczny. Kierowanie optymalne charakteryzuje się tym, że element kierujący dąży do takiego działania, aby doprowadzić określoną funkcję (lub więcej funkcji – w przypadku kierowania wielokryterialnego) do jej (ich) maksimum lub minimum. Na przykład gospodarka leśna ma na celu nie tylko pozyskanie maksymalnej ilości drewna, lecz również zachowanie zasobów leśnych i niedopuszczenie do erozji gleby.

Prawie wszystkie metody kierowania optymalnego zawierają jakąś postać modelu kierowanego obiektu, a sukces metody w ogromnym stopniu zależy od jakości tego modelu i jego adekwatności do oryginału.

Aparat matematyczny kierowania optymalnego został opracowany dość szczegółowo. Klasyczne zadanie rachunku wariacyjnego, dotyczące wyliczenia funkcji krzywej, po której powinien przemieszczać się punkt materialny – bez tarcia, tylko pod wpływem siły ciężkości, w ciągu czasu minimalnego, zostało rozwiązane sposobem analitycznym już w XVIII wieku. Rozwiązanie zadania programowania liniowego w pierwszej połowie XX wieku wzbudziło wiele nadziei, zwłaszcza w dziedzinie ekonomiki. Jednakże dla kierowania układami ekologicznymi, a tym bardziej krajobrazowymi, programowanie liniowe na niewiele się zdaje, gdyż wymaga liniowości, a systemy ekologiczne są z samej swojej istoty nieliniowe.

Kolejnym ważnym etapem na drodze do optymalnego kierowania było zastosowanie teorii gier. Gry antagonistyczne są w istocie modelami działań bojowych i w żaden sposób nie nadają się do opisywania współdziałania człowieka z przyrodą. Co innego w przypadku tak zwanych „gier z przyrodą", w których jeden z uczestników gry nie ma zamiaru wyrządzać szkody przeciwnikowi, lecz strategia jego działania pozostaje nieznana.

Wraz z rozpowszechnieniem komputerów poszukiwanie ekstremum funkcji wielu zmiennych (do czego często sprowadza się zadanie optymalnego kierowania) stało się zadaniem w pełni rozwiązywalnym. Szerokie zastosowanie znalazły gradientowe metody prognozowania matematycznego. Jednakże słabym ogniwem optymalnego kierowania było, i jest nadal, stworzenie adekwatnego modelu kierowanego obiektu. Nawet najlepsze metody matematyczne nie są w stanie wypracować optymalnej strategii walki z erozją i zasoleniem gleby albo eutrofizacją zbiorników zaporowych i jezior, skoro nie umiemy uogólnić, w postaci modeli, całego zasobu wiedzy o przebiegu i prawidłowościach procesów erozji czy eutrofizacji.

Wróćmy jednak do celów konstruowania modeli. Model naturalnego obiektu może być tworzony w celu weryfikacji hipotez naukowych. Chodzi o to, że nasza wiedza o naturalnych obiektach ma z reguły charakter hipotez, które należy przyjąć, odrzucić lub wskazać ograniczoną sferę ich stosowalności. Już w XIX wieku J. von Liebig wysunął hipotezę, że rozwój roślin podlega ograniczeniu przez ten czynnik (na przykład biogen), którego jest najmniej. Badanie licznych modeli wzrostu roślin, od jednokomórkowych planktonowych glonów do drzew, potwierdziło słuszność tej hipotezy i jednocześnie również jej ograniczoność, gdyż w przypadku wzajemnego oddziaływania czynników okazała się ona nie do przyjęcia (przejście do pobierania azotu atmosferycznego przy wysokich koncentracjach fosforu nieorganicznego). Bywają przypadki, kiedy dla opisu tego samego zjawiska powstają dwie alternatywne hipotezy. Wówczas modele zbudowane na podstawie tych hipotez znajdują się w systemie konkurencyjnym, który może trwać bardzo długo. Klasycznym przykładem takiej sprzeczności są modele falowej i korpuskularnej natury i propagacji energii świetlnej.

Wreszcie tworzone są modele dla celów szkoleniowych. Uruchomienie symulatora lotu samolotu lub statku kosmicznego jest niemożliwe bez sporządzenia modeli, zazwyczaj komputerowych, sterowanych pojazdów i ich otoczenia. Również sztabowe gry wojenne lub gry menedżerskie i biznesowe muszą mieć odpowiednie odwzorowanie, często bardzo skomplikowane i kosztowne. Rozrywkowe gry komputerowe dla dzieci i dorosłych konstruowane są w większości przypadków na bazie odpowiednich modeli. Tak na przykład podstawą znanej gry *Sim City* jest J. Forrestera model rozwoju miasta (Forrester 1974).

Zajmiemy się teraz technologią modelowania, czyli zagadnieniem, jak – mając oryginał i wytyczony cel modelowania – zbudować sam model. Dążenie do tego, aby budowane modele były jak najbardziej podobne do oryginałów, powodowało, że pierwsze z nich były budowane z tego samego materiału, co oryginały, i starały sie być jak najbardziej do nich podobne. Na przykład procesy w korycie rzeki są modelowane na zmniejszonej kopii tej rzeki umieszczonej na stole, a niekiedy zajmującej cały rozległy budynek. W ekologii przedsięwzięto próby modelowania populacjii użytecznych ryb za pomocą populacji skorupiaków planktonowych – rozwielitek. Klasycznym przykładem takiego modelowania był amerykański projekt BIOSPHERE-2, który imitował izolowany ekosystem lądowy. Tego rodzaju modele są bardzo kosztowne i mają tę wielką wadę, że czas płynie w nich z tą samą szybkością, co rzeczywisty czas otoczenia. Dlatego praca z takimi modelami może trwać całe lata, a wyniki rzadko warte są poniesionych kosztów. Takie modele w zastosowaniu do obiektów ożywionych nazywają się biologicznymi.

Gdy oryginał jest układem materialnym określonego rodzaju, a model jest też układem materialnym, ale innego rodzaju, taki sposób modelowania nazywa się analogowym. Układ „drapieżnik – ofiara" był tym sposobem modelowany za pomocą sieci elektrycznej, w której natężenie prądu wyznaczało liczebność zwierząt. Ten rodzaj modelowania był realizowany na analogowych maszynach liczących. Takie prototypowe komputery zostały całkowicie wyparte przez maszyny cyfrowe. Dlatego też modelowanie analogowe należy już wyłącznie do historii.

Jeżeli oryginał jest systemem materialnym, a model systemem informacyjnym, to modelowanie nazywamy symbolicznym. Szczególnym przypadkiem modelowania symbolicznego jest modelowanie werbalne, czyli po prostu opis oryginału za pomocą zwykłego języka. Ten najstarszy sposób modelowania zachowuje ugruntowaną pozycję w takich naukach jak systematyka roślin i zwierząt lub historia. Niekiedy modele werbalne nazywane są konceptualnymi, zwłaszcza wtedy gdy przewiduje się późniejsze przejście do innych sposobów modelowania.

Bardzo rozpowszechnioną odmianą modelowania symbolicznego jest modelowanie matematyczne, które opisuje istotne właściwości oryginału jako pewnego obiektu matematycznego. Zależnie od tego, jaki matematyczny obiekt lub konstrukcja jest użyta w modelu, wyróżniamy rodzaje modeli matematycznych: układ prostych równań różniczkowych, modele w postaci układu równań różniczkowych cząstkowych, modele w postaci procesów stochastycznych, końcowych automatów i automatów komórkowych, modele wykorzystujące rachunek predykatów i inne systemy logiczne, modele stosujące aparat logiki rozmytej i logiki wieloznacznej oraz modele logiczno-lingwistyczne .

W literaturze spotyka się niekiedy dość dziwaczne kombinacje wielu typów modeli. Nie sądzimy, aby należało teraz szczegółowo rozpatrywać wymienione powyżej rodzaje modeli. Niezbędne informacje o automatach komórkowych są podane w rozdziale 4, gdzie ten aparat matematyczny jest wykorzystany do konstruowania modeli konkretnych układów krajobrazowych.

Kolejnym, nie mniej ważnym od matematycznego jest modelowanie imitacyjne, w którym jako model pojawia się program komputerowy. W szczególności ów program lub – jak go się także nazywa – algorytm modelujący – może być implementacją pewnego modelu matematycznego. Dla ułatwienia modelowania imitacyjnego rozwinięto specjalne języki (na przykład: *Simula*, *Stella*, *Madonna*), jednakże można w tym celu równie dobrze posługiwać się uniwersalnymi językami programowania wysokiego poziomu (na przykład: *C++*, *Pascal*, *Delphi*, *Visual Basic*). Ważną odmianą modelowania imitacyjnego jest modelowanie indywidualne, które obejmuje dynamiczne tworzenie licznych obiektów oraz ewentualne ich niszczenie. Przykładem takiego podejścia jest model zespołu leśnego opisany w rozdziale 3. Wszystkie modele opisane w tej książce są modelami imitacyjnymi, a ich programy zostały napisane w języku *Visual Basic 6.0* firmy Microsoft.

W modelowaniu krajobrazowym pierwszorzędną rolę odgrywa sposób, w jaki prezentowana jest przestrzeń. Te modele, w których przestrzeń w opisie zmiennych jest całkowicie ignorowana, nazywają się punktowymi. Jeżeli zmienne ulegają zmianom tylko na jednej z osi współrzędnych, mamy do czynienia z modelem jednowymiarowym. W modelowaniu ekosystemów lądowych zazwyczaj najciekawsze jest pionowe rozmieszczenie wartości zmiennych – od skały macierzystej podścielającej glebę do szczytu koron drzew. Model, w których zmienne zmieniają się na dwóch współrzędnych, nazywamy płaskimi. Jest to najczęściej stosowana metoda w modelowaniu krajobrazowym, opartym na zastosowaniu materiału kartograficznego i systemu informacji geograficznej (GIS). Modele objętościowe, w których zmienne zmieniają się na wszystkich trzech współrzędnych, stosowane są w modelowaniu krajobrazowym stosunkowo rzadko.

Teraz kilka zdań o praktyce badań naukowych z zastosowaniem modelowania. Zacznijmy od tego, że modelowanie imitacyjne nie jest metodą uniwersalną w rozwiązywaniu zagadnień naukowych. Jeżeli nie mamy wystarczająco dopracowanej hipotezy o mechanizmach funkcjonowania oryginału, to budowanie modelu nie ma większego sensu. Jedyne, co można wtedy zaproponować, to koncepcja „czarnej skrzynki" i statystyczna obróbka danych. Konstruowanie obecnie modelu rozwoju roślinności na Marsie byłoby nieco przedwczesne, biorąc pod uwagę bieżący stan wiedzy na ten temat.

Niezwykle trudnym zadaniem bywa odpowiedni dobór stopnia uszczegółowienia opisu właściwości zarówno oryginału, jak i modelu. Z jednej strony to bardzo pociągające zbudować prosty model z minimalną liczbą zmiennych. Taki model możemy stosunkowo łatwo przebadać, a wyniki przedstawić w postaci przejrzystej grafiki. Z drugiej jednak strony upraszczając zagadnienie ryzykujemy usunięcie takich właściwości, których brak może wypaczyć całą specyfikę zachowania się modelu. Upraszczając troficzną sieć ekosystemu do łańcucha troficznego, możemy dojść do zupełnie nierzeczywistego wniosku, że ekosystem z nieparzystą liczbą troficznych ogniw jest stabilny, a z liczbą parzystą – niestabilny. Mechanizm takiej zależności jest niezbyt skomplikowany, ale nie ma nic wspólnego z ekologią! W każdym konkretnym przypadku należy więc z wielką uwagą szukać odpowiedniego kompromisu między prostotą a prawdziwością konstruowanych modeli.

Inny problem powstaje przy wyborze stopnia dokładności danych wchodzących w skład modeli imitacyjnych. Współczesny komputer może operować liczbami zawierającymi dziesięć i więcej cyfr znaczących, a dane liczbowe, powstające w wyniku badania naturalnych ekosystemów, w najlepszym przypadku zawierają trzy znaczące cyfry. Dlatego też należy uciekać się do specjalnych sposobów, które zapobiegają przypisywaniu zbyt wysokiej dokładności danym pojawiającym się w komputerze. Do takich sposobów należy metoda dodawania szumu lub zastosowanie arytmetyki liczb rozmytych.

Możliwość posłużenia się wynikami otrzymanymi przy modelowaniu jednego obiektu przy pracy z innymi obiektami jest ważną własnością badań modelowych. Dotyczy ona, w pewnym sensie, stopnia uogólnienia modeli. Modele można zgrupować właśnie na podstawie ich stopnia uogólnienia. Modele portretowe dotyczą wyłącznie jednego obiektu przyrodniczego i nie pretendują do jakiegokolwiek uogólnienia, natomiast modele teoretyczne mogą być zastosowane do całej klasy (grupy, rodzaju) obiektów. Przykładem modelu portretowego może być model zespołu roślinności parku narodowego Sichote-Alińskiego (Suhanov et al. 1994), natomiast modelu teoretycznego – model rozkładu materii organicznej w glebie (Bosatta 1982).

Badania modelowe zazwyczaj zaczynają się od sformułowania koncepcji modelu w postaci werbalnej i graficznej. Następnie należy zdecydować, jaki ma być typ modelu, zestawić wykaz zmiennych i parametrów. Wówczas dopiero, na podstawie dostępnej wiedzy o obiekcie, popartej opiniami ekspertalnymi, można zestawić spodziewane funkcjonalne zależności między zmiennymi. Z reguły wśród dostępnych wyników bezpośrednich badań nie można odnaleźć wszystkich danych niezbędnych do zbudowania modelu, część parametrów więc pozostaje nieokreślona. W pierwszych wariantach modelu te nieokreślone parametry otrzymują wielkości prawdopodobne i przybliżone; ich dalsze precyzowanie odbywa się w procesie kalibracji.

Proces k a l i b r a c j i polega na doborze takiej wartość parametru nieokreślonego, aby uruchomiony model zachowywał się najbardziej zgodnie z oryginałem. W pewnych przypadkach kalibracja może być ostatecznym celem modelowania; chodzi wówczas właśnie o oznaczenie rzeczywistej wielkości nieznanego parametru. Należy jednak pamiętać, że w tych przypadkach, kiedy nieoznaczonych parametrów jest dużo, a danych o funkcjonowaniu oryginału niewiele, kalibracja staje się zawodna i trzeba szukać innych metod oznaczania nieznanych współczynników.

Tak więc pokrótce przedstawiliśmy sposoby sprawdzania hipotez naukowych za pomocą narzędzi modelowania. Pozostaje dodać, że modelowanie nie jest procesem liniowym, lecz cyklicznym. Wyniki modelowania zazwyczaj nie tyle odpowiadają na zapytanie postawione na wstępie, co wskazują luki i błędy popełnione w czasie budowy modelu i cały proces zaczyna się od początku, często od oceny i ewentualnej zmiany samego typu modelu, a następnie skontrolowania algorytmu modelującego.

2. Model bilansowy

2.1. Schematy blokowe modelu bilansowego

Sztuka modelowania w ekologii zaczęła się od bilansów energii w systemach ekologicznych – od osobników do ekosystemów. Podstawy tego kierunku wiążą się z publikacjami klasyków ekologii (Bossel 1985; Straškraba, Gnauck 1885; Straškraba 1994; Jørgensen, Nielsen 1994). Przeważają badania i syntezy dotyczące środowiska wodnego, ponieważ zbiorniki wodne były znacznie łatwiejsze do objęcia zarówno badaniami w terenie, jak i w wyobraźni badaczy, gdyż są wyraźnie odgraniczone od reszty środowiska, a ich zasiedlenie, zwłaszcza w otwartej toni wodnej, jest nieporównanie bardziej równomierne niż w środowisku lądowym. Dopiero Międzynarodowy Program Biologiczny (MPB, lata 1964-1985) zainicjował badania ekosystemów całej biosfery ziemskiej, zwłaszcza wielkich biomów lądowych: prerii, tajgi, tundry, pustyń i gór, jak gdyby podsuwając wyobraźni badaczy istnienie ekologicznych układów wieloekosystemowych, czyli krajobrazów.

Międzynarodowy Program Biologiczny przyniósł poza tym wspólnie zaplanowane gromadzenie danych za pomocą wybranych i interkalibrowanych metod na określonych terenach i akwenach. Dane te były zgromadzone w bazach danych i publikowane w specjalnych wydawnictwach, dostępnych do wykorzystania przez wielkie zespoły. Nawiasem mówiąc, jak dotąd nie udało się powtórzyć tak zaplanowanych i podobnie owocnych badań.

Wspomniane synoptyczne i rozległe, a przez to kosztowne i niepowtarzalne, gromadzenie danych było wielką zachętą do szybkiego rozwoju sztuki (bo to stale jest w dużym stopniu sztuka!) modelowania ekologicznego. Wyniki Międzynarodowego Programu Biologicznego były przedstawiane w postaci bardzo rozbudowanych schematów blokowych. Zawierały one dane o ilości energii nagromadzonej w biomasie poszczególnych bloków, pięter czy ogniw łańcuchów pokarmowych w ekosystemach i krajobrazach, a także dane przepływu tej energii i krążenia pierwiastków, zwłaszcza biogennych. Przykłady takich modeli podano w wielu publikacjach (Bossel 1985; Botkin 1993; Running 1994; Voit, Sands 1996).

W badaniach dynamiki leśnych ekosystemów Bieszczad zastosowano bilansowy model TERECO. Jak widać z rys. 2.1, który prezentuje ogólny schemat blokowy modelu TERECO ekosystemu lasu (zob załączony CD), oznaczanie składników bilansu ekosystemu następuje w odrębnych submodelach: LIGHT – pionowe rozmieszczenie energii świetlnej w ekosystemie; TREE – przyrost i ubytek drzew na skutek naturalnej śmiertelności; BUSH – analogiczny przyrost i ubytek krzaków; GRASS – to samo w odniesieniu do traw; SOIL – procesy mineralizacji i erozji w glebie.

Podstawą modelu TERECO są równania bilansu materii w ekosystemie. Ponieważ krok czasowy modelu wynosi jeden rok, nie uwzględniano biomasy roślin jednorocznych (traw), a jedynie biomasy drzew (BIOM) i krzaków (BIOMBUSH). Stan gleby charakteryzują masy martwej materii organicznej (ORGANIC) i mineralnych związków azotu (NDEP). Zmiany biomasy drzew oznacza się jako:

$$\Delta BIOM = TREE\_GROWTH - TREE\_FRUIT\_PROD - TREE\_DRY - TREE\_FALL - HOLZ$$

gdzie:

TREE_GROWTH – roczny przyrost biomasy drzew,
TREE_FRUIT_PROD – generatywna produkcja drzew,
TREE_DRY – usychanie drzew przy brakach wody,
TREE_FALL – opad,
HOLZ – wyrąb lasu.

Analogicznie oznacza się zmiany biomasy krzaków:

$$\Delta BIOMBUSH = BUSH\_GROWTH - BUSH\_FRUIT\_PROD - BUSH\_DRY - BUSH\_FALL$$

Bilans martwej materii organicznej w glebie oznacza się na podstawie równania:

$$\Delta ORGANIC = GRASS\_MORT - MINR + BUSH\_FALL + BUSH\_LEAFS + BUSH\_DRY + TREE\_LEAFS + TREE\_DRY + TREE\_FALL - ERA$$

gdzie (oprócz już opisanych symboli):

GRASS_MORT – biomasa martwej trawy,
MINR – mineralizacja glebowej materii organicznej,
ERA – ubytek materii organicznej w glebie na skutek erozji.
BUSH_LEAFS – biomasa krzaków
TREE_LEAFS – biomasa liści

Bilans azotu mineralnego w glebie oznacza się w taki sposób:

$$\Delta NDEP = MINR + DUNG + NOX + PRIM - TREE\_NIT\_CONS - BUSH\_NIT\_CONS - NSIK - NERO - GRASS\_NIT\_CONS,$$

gdzie:

DUNG – dopływ azotu w postaci nawozów,
NOX – wiązanie azotu przez organizmy glebowe,
PRIM – dopływ azotu wraz z opadami,
TREE_NIT_CONS, BUSH_NIT_CONS, GRASS_NIT_CONS – zużycie azotu przez drzewa, krzaki i trawy,

NERO – ubytek azotu na skutek erozji gleby,
NSIK – ubytek azotu na skutek spływu powierzchniowego.

Schemat blokowy submodelu przenikania energii świetlnej w ekosystemie znajduje się na rys. 2.2. Jeżeli wysokość drzew (HTREES) jest większa niż wysokość krzaków (HBUSH), to energia słoneczna dosięgająca powierzchni ziemi (SOL) bierze udział w określaniu wielkości fotosyntezy drzew bez zmian (LIGHT_TREE). Natomiast w rzadkich przypadkach, kiedy krzaki są wyższe od drzew, uwzględnione zostaje pochłanianie światła przez krzaki. Pochłanianie światła przez korony drzew (TRANSTREES) jest oznaczane na podstawie wielkości wskaźnika liści (TREE_LEAF_INDEX) za pomocą funkcji FL7 (małe wykresy na rys. 2.2 – 2.5 ilustrują, w sposób bardzo uproszczony, zależności funkcyjne). Sam wskaźnik liści oznacza się wychodząc od masy liści (TREE_LEAFS), która z kolei zależy od stosunku masy liści do masy całego drzewa (LTL). Ostatnia wielkość związana jest funkcją FL10 z biomasą drzewostanu (BIOM).

Pochłanianie światła przez krzaki (TRANSBUSH) obliczane jest analogicznie do schematu obliczania pochłaniania światła przez drzewa, z tym że liczbowe wartości współczynników są inne.

Trawy otrzymują jedynie tę część energii świetlnej (LIGHT_GRASS), która pozostaje po przejściu przez liście drzew i krzaków. Energia słoneczna dochodząca do powierzchni gleby (LIGHT_SOIL) – służąca w submodelu gleby (SOIL) do oznaczania temperatury gleby – jest oznaczana z uwzględnieniem zacieniającego oddziaływania traw (TRANSGRASS).

Na rys. 2.3 przedstawiono schemat blokowy submodelu wzrostu drzew (TREE). Centralną pozycję w tym submodelu zajmuje określenie wielkości fotosyntezy w liściach (PHOTOSYN). Zakłada się, że ta wartość jest ograniczana warunkami świetlnymi (POSY), dostępnością wody (PWS) i mineralnego azotu oraz warunkami termicznymi (DEGDAYS). Warunki świetlne wyznacza funkcja FT2, określająca dopływ energii słonecznej (LIGHT_TREES), a funkcja FT1 wyznacza warunki mineralnego odżywiania zależne od ilości azotu w glebie (NDEP). Ograniczanie fotosyntezy z powodu niedoboru wody zaczyna występować dopiero w przypadku deficytu wilgoci (WBSD), powiązanego przez funkcję FT3 z roczną ilością opadów (RREG). Warunki termiczne fotosyntezy określane są przez liczbę stopniodni w ciągu okresu wegetacyjnego (DAGDAYS). Przyrost biomasy drzew (TREE_GROWTH) określa się jako różnicę między wielkością fotosyntezy i kosztów produkcji liści (TREE_LEAFS), które opadają w końcu okresu wegetacyjnego, oddychaniem liści (LEAF_RESPIRATION) oraz oddychaniem pni, gałęzi i korzeni drzew (STEM_RESPIRATION). Poza tym uwzględnia się możliwość śmierci całego drzewa lub jego części (TREE_FALL) w przypadku wystąpienia warunków zagrażających jego życiu (TREE_STRESS). Takie warunki mogą wiązać się z trującymi zanieczyszczeniami środowiska zewnętrznego, działaniem pasożytów lub klęskami żywiołowymi.

Struktura submodelu wzrostu krzaków (BUSH) jest taka sama jak w przypadku wzrostu drzew. Jedyne odmienności polegają na różnicach liczbowych wielkości współczynników i postaci zależności funkcjonalnych.

Schemat blokowy submodelu wzrostu traw (GRASS) prezentuje rys. 2.4. Biomasa traw, jakie wyrosły w danym roku (BIOM_GRASS), jest określana na podstawie warunków świetlnych panujących na poziomie dolnej warstwy ekosystemu (PHG), dostępności wody (WAWS) i mineralnych substancji odżywczych w glebie (NFG). Warunki świetlne na poziomie traw oznacza się na podstawie stopnia pochłaniania światła przez liście drzew i krzaków (submodel LIGHT) i funkcji naświetlenia F4 (LIGHTGRASS), oczywiście jeżeli w danym ekosystemie występują drzewa i/lub krzewy. Deficyt wilgoci w glebie, wilgoci koniecznej dla dobrego rośnięcia traw (WSO), zależy od ilości opadów w ciągu roku (RREG) w funkcji F1. Zużycie azotu na wzrost traw (GRASS_NIT_CONS) określa się na podstawie ilości azotu w glebie (NDEP) i funkcji F3.

Schemat blokowy submodelu procesów zachodzących w glebie (SOIL) znajduje się na rys. 2.5. Centralną pozycję w tym submodelu zajmuje proces mineralizacji materii organicznej (MINERALIZATION). Szybkość tego procesu jest regulowana przez warunki termiczne (TMIN), zawartość wody w glebie (FMIN) i obecność martwej materii organicznej (DIKF). Temperaturę gleby (TBO) warunkuje średnia temperatura powietrza (TEMP) i bezpośrednie promieniowanie słoneczne docierające do poziomu gleby (LIGHT_SOIL). Wilgotność gleby jest warunkowana przez ilość opadów, co symbolizowane jest poprzez funkcję FS3. Ilość azotu mineralnego uwolnionego w rezultacie rozkładu materii organicznej zależy także – przy takich samych pozostałych warunkach – od początkowej zawartości materii organicznej w glebie (ORGANIC).

Proces erozji gleby zależy przede wszystkim od kąta nachylenia powierzchni gleby, na której znajduje się modelowany ekosystem (SLOPE). Ilość materii organicznej wynoszonej z gleby (ERA) zależy nie tylko od nachylenia (EHA), lecz również od ilości opadów i zawartości materii organicznej w glebie (EMA). Ten sam submodel służy również do określania ilości mineralnego azotu, jaki traci gleba w wyniku erozji (NERO) i wymywania przez stok powierzchniowy (NSIK). Względny udział materii wynoszonej wraz z wodą (RSIK) jest oznaczany na podstawie ilości opadów atmosferycznych.

2.2. Przykłady zastosowania modelu

Dla podstawowego wariantu modelu przyjęto warunki typowe dla regionu Bieszczad: średnia roczna temperatura – 10°C, opady – 1000 mm rocznie, intensywność radiacji słonecznej – 250 Wat/m^2. Przy początkowych zasobach azotu mineralnego w glebie 5 kg/ha i intensywności dopływu azotu z opadami 20 kg/ha/rok model uzyskuje stan stacjonarny po 250 latach; przebieg tego procesu przedstawia rys. 2.6.

W ciągu pierwszych dziesięciu lat następuje gwałtowny rozwój traw, których biomasa osiąga maksymalnie 4 t/ha. Następnie biomasa traw szybko maleje, a biomasa krzaków szybko rośnie, osiągając maksymalnie około 40 t/ha po 50-70 latach od stanu wyjściowego. W tym okresie biomasa drzew osiąga podobne wielkości, lecz ciągle rośnie i osiąga 390 t/ha po 500 latach od początku działania modelu; wtedy biomasa krzaków spada do wielkości bliskich zeru z powodu braku światła. Podobnie biomasa traw maleje do zaledwie 0,14 t/ha. Zawartość materii organicznej w glebie podlega silnym wahaniom w czasie przekształceń struktury ekosystemu, aby później ustabilizować się na poziomie 20 t/ha. Zasób azotu nieorganicznego w glebie podlega mniejszym wahaniom niż zawartość materii organicznej i po początkowym wzroście również ulega stabilizacji.

Zachęcamy wszystkich Czytelników do otworzenia płyty CD dołączonej do książki. (Opis uruchomienia CD jest podany w załączniku B). Pozwala to na natychmiastowe uruchomienie modelu. Tym, którzy chcą wiedzieć, jak to się ma do rzeczywistości, możemy powiedzieć, że tak naprawdę nie ma danych. Nikt nie robił obserwacji w ciągu 500 czy 250 lat. Natomiast w literaturze są oddzielne fragmenty danych z różnych etapów rozwojowych (na przykład krajobrazów leśnych czy łąkowych), które potwierdzają ustalone przez model zależności.

Na przykład w Bieszczadach w procesie sukcesji wtórnej na otwartych przestrzeniach najpierw intensywnie rozwijają się trawy. Po kilku czy kilkudziesięciu latach zarastają krzakami, które dalej zmieniają się na zbiorowiska leśne z dominacją buka czy jodły. W różnych warunkach ekologicznych czas takich zmian będzie różny. W dolinie górskiej części rzeki San łąkowe systemy krajobrazowe istnieją dłużej, zmieniając się w zarośla olszy szarej. Na stokach wśród lasów łąki szybko zarastają bukiem lub jodłą. Natomiast na górnej granicy lasu na wysokościach powyżej 1000 m n.p.m. łąkowe zbiorowiska istnieją bardzo długo, a w strefie subalpejskiej istnieją prawie stale. Lasy zapewne istniały tu w czasie ciepłego okresu atlantyckiego, kiedy górna granica strefy leśnej znajdowała się wyżej od aktualnej; teoretycznie możemy założyć zarastanie takich subalpejskich łąk przez las, pod warunkiem jednak zmian klimatycznych w kierunku ocieplenia.

Za pomocą modelu Czytelnik ma możliwość nie tylko sprawdzenia już opisanych w literaturze zmian zachodzących w krajobrazach Bieszczad (Zarzycki 1963; Jaworski 1994), lecz także przewidzieć nowe, do tej pory nie opisane.

Rys. 2.7 ilustruje zależność ustabilizowanych stanów modelu w zależności od stromizny nachylenia powierzchni; jest to stan, który w tym przypadku pojawia się po około 500 latach od stanu wyjściowego. Należy pamiętać, że na poziomej osi wykresu pokazane są symulacje zmian dynamicznych (jak to się dzieje na rys. 2.6). W zakresie kąta nachylenia od 0 do 20-25 stopni właściwości ekosystemu zmieniają się niewiele. Dalszy wzrost stromizny nachylenia powoduje gwałtowny spadek biomasy drzew i wzrost biomasy krzaków, których przy płaskim gruncie w ogóle brak. Przy kącie nachylenia 40 stopni drzewa w ogóle nie mogą istnieć, a biomasa krzaków szybko maleje i zanika przy nachyleniu 45 stopni. Jedynie trawy utrzymują się przy życiu, a nawet zwiększają biomasę przy większym nachyleniu. Przy nachyleniach większych niż 20-25 stopni wzrastają gwałtownie procesy erozji i wymywania materii organicznej i azotu, co jest głównym powodem przebudowy ekosystemu. W Bieszczadach Polskich te zmiany nie są tak drastyczne jak w Beskidach Ukraińskich, gdzie presja gospodarcza, w postaci całkowitego wycinania lasów naturalnych na stokach, spowodowała nasilenie procesów erozyjnych.

Na rys. 2.8 umieszczono zależność granicznych (klimaksowych) właściwości ekosystemu od temperatury – a więc również od ilości stopniodni. Obniżenie temperatury prowadzi do zamiany dominujących w ekosystemie zespołów drzew przez zespoły krzaków, a przy skrajnie niskich temperaturach – przez trawy. Przy zerowych liczbach stopniodni ekosystem w ogóle przestaje funkcjonować. Ze spadkiem temperatury maleją zasoby materii organicznej w glebie, a rośnie koncentracja azotu organicznego. Ten ostatni symptom wynika z tego, że dopływ azotu z atmosfery jest stały, a nie może być on spożytkowany na wzrost roślin. Podwyższenie średniej rocznej temperatury do 20 stopni prowadzi do wzrostu biomasy drzew do 600 t/ha i pojawienia się zarośli krzaków, mimo zacieniania przez wielkie drzewa o wysokości rzędu 40 m. Biomasa traw też rośnie wraz ze wzrostem temperatury.

Na rys. 2.9 przedstawiono zależność stabilnych stanów ekosystemu od ilości opadów. Obniżanie ilości wody dopływającej w postaci opadów powoduje spadek biomasy drzew, a przy 500 mm rocznie maleje ona do zera i ekosystem nabiera cech stepu. Wiąże się to ze spadkiem ilości materii organicznej w glebie. Wzrost ilości wody prowadzi – przy jednakowych pozostałych warunkach – do spadku biomasy zespołów leśnych, jednakże wówczas wyraźnie rośnie ilość materii organicznej w glebie. Model nabiera przy tym cech ekosystemu bagiennego, gdyż szybkość mineralizacji materii organicznej wyraźnie spada na skutek wysokiej wilgotności gleby.

Przy ilości opadów powyżej 2500 mm rocznie w ekosystemie pozostają tylko trawy i gwałtownie rośnie zasób materii organicznej. Jeżeli do tego dojdzie wzrost temperatury i oczywiście liczba stopniodni, model zaczyna imitować wilgotny las tropikalny z wielkimi biomasami wszystkich komponentów.

3. Modele FORKOME jako przykłady modelu płatowego*

3.1. Specyfika modeli płatowych

Omówione w poprzednim rozdziale modelowanie bilansowo-energetyczne nie jest w stanie sprostać zadaniom imitacji tak złożonych zjawisk, jak na przykład adaptacja do warunków biologicznych i abiotycznych czy przestrzennego rozmieszczenia organizmów żywych w krajobrazie. Dopiero rozwój modelowania osobniczego (ang. *individual-based modelling*) zapewnił istotny postęp poprzez zajęcie się zachowaniem poszczególnych osobników.

W latach sześćdziesiatych ubiegłego wieku rozwój technik komputerowych zbiegł się z rozkwitem metod modelowania procesów ekologicznych (Shugart 1984; Botkin 1993). W wielu ośrodkach naukowych rozpoczęto konstruowanie symulacyjnych modeli krajobrazów, także krajobrazów leśnych (rys. 3. 1).

Przykładem takich modeli są modele luk. Takie modele w literaturze anglojęzycznej były nazywane *gap-models* (od *gap* – luka w drzewostanie) (Shugart 1984). Taka luka powstaje po obumarciu jednego, czy kilku dużych drzew. Proces zarastania luk jest uzależniony od drzew sąsiadujących z powstałą luką (rys. 3.2) i od młodych osobników drzew rosnących w samej luce.

W procesie zarastania luki na jej miejscu powstaje płat drzewostanu. Taki płat jest podstawowym elementem, w którym symulowana jest dynamika krajobrazów leśnych (rys. 3.3). Zamiast *gap-models* takie modele nazywane są obecnie (Shugart, Smith 1996) *patch-models* (od *patch* – jednorodny płat drzewostanu) – modelami płatowymi.

Modele płatowe traktują las jako złożony system ekologiczny. Takie podejście jest w polskim leśnictwie, jak dotąd, słabo rozwinięte (Brzeziecki 1999); przeważają modele empiryczne. Natomiast w wielu krajach Europy i w Stanach Zjednoczonych modele płatowe znajdują szerokie zastosowanie (Botkin et al. 1972; Sullivan, Clutter 1972; Solomon 1974;

* Badania uzupełniające, niezbędne do przygotowania tego rozdziału zostały przeprowadzone w ramach projektu KBN nr 6 P06L 04221

Mitchell 1975; Suzuki, Unemura 1974; Horn 1975; Waggoner, Stephens 1970; Shugart, West 1977; Shugart 1984).

W ostatnim dziesięcioleciu powstały modele uwzględniające pewne wybrane kierunki: procesy glebowe (Pastor, Post 1985), koncepcje fitosocjologiczne (Kienast 1987), strukturę korony drzew (Leemans, Prentice 1989), zjawiska ekofizjologiczne (Friend et al. 1993) i biofizyczne (Bonan, van Cleve 1992; Martin 1992), konkurencję wewnątrzgatunkową (Pawłowski 1996). Wydaje się, że zwiększenie kompleksowości tego rodzaju modeli może polepszyć wyniki prognozowania leśnej sukcesji i dynamiki krajobrazu.

Modele płatowe szeroko uwzględniają analizę logiczną przyrodniczych związków decydujących o losach danego obiektu (Polański 1981). Takie modele pozwalają lepiej zrozumieć mechanizmy jego funkcjonowania (Mozgawa 1985).

Modele te mogą mieć zastosowanie w rozwiązywaniu problemów związanych z coraz bardziej popularną półnaturalną uprawą lasów, tendencją do zastąpienia leśnictwa monokulturowego leśnictwem ekosystemowym (Brzeziecki 1999).

W modelach płatowych, zajmujących pozycję pośrednią między modelami pojedynczego drzewa i modelami całego drzewostanu (Shugart et al. 1980), obliczenia wykonywane są wprawdzie dla każdego drzewa oddzielnie, ale przy uwzględnieniu wpływu najbliższego sąsiedztwa innych drzew, występujących na powierzchni próbnej wielkości od kilku do kilkunastu setnych hektara. Przeważnie jest to wielkość równa 1/12 ha (Botkin et al. 1972).

Opisana wyżej metoda opiera się na założeniu, że ekosystem leśny stanowi przestrzenną mozaikę elementów (różnej wielkości płatów), przechodzących cykl przemian (faz rozwojowych) charakterystycznych dla danego typu lasu (teoria cykliczno-mozaikowa; Boehmer 1997). Takie względnie jednolite płaty mogą mieć różną wielkość, w zależności od przestrzennego zróżnicowania warunków siedliskowych, a także od charakteru zaburzeń naturalnych i antropogennych.

Jeszcze w pracy A. Watta (1947) wyróżniono trzy podstawowe fazy, które kolejno się zmieniają w cyklu odnowy lasu. Pierwsza to faza okna (*gap-phase*), która zaczyna się od obumarcia jednego lub kilku drzew. Drugą jest faza budowy (*building-phase*), w której dominują młode drzewa. Trzecią jest faza dojrzałości (*maturity-phase*), składająca się z dojrzałych drzew.

Terminologia A. Watta jest obecnie szeroko stosowana w literaturze światowej (np. Halle et al. 1978). Las przedstawiany jest jako przestrzenna mozaika plam (Popadjuk et al. 1991), znajdujących się na różnych etapach rozwoju i stanowiących elementy krajobrazu leśnego (rys. 3.4).

Dla mozaikowej struktury lasu zostało przez N. Dylisa (Dylis 1978) zaproponowane pojęcie „parceli biogeocenotycznej". Taka „parcelowa struktura" krajobrazów leśnych była przedmiotem rozważań innych uczonych z byłego Związku Radzieckiego (*Osnowy lesnoj biogeocenologii* 1964; Turkow 1985).

Podkreślając stabilność składu flory całego leśnego obszaru (krajobrazu leśnego), który istnieje jako system asynchronicznie rozwijających się płatów, F. H. Bormann i G. E. Likens (1979) zaproponowali nazywać to zjawisko „zmienną mozaiką stabilnego stanu". Możemy mówić o różnych poziomach mozaikowości (tab. 3.1), spowodowanych przez wiele przyczyn – od wypadnięcia poszczególnych drzew lub ich części do pożarów czy huraganów.

Tab. 3.1. Warianty mozaik krajobrazów leśnych.

ROZMIAR MOZAIK (m^2)	PRZYCZYNY POWSTAWANIA ZMIAN
100 000 – 10 000 000	Huragany, powódź, trzęsienia ziemi, pożary
1000 – 10 000	Sztormy, grupowe wypadanie drzew w wyniku chorób czy innych przyczyn
100 – 1000	Wypadanie oddzielnych drzew
10 – 100	Opad oddzielnych gałęzi

Podstawą wszystkich istniejących obecnie wersji modeli płatowych jest model JABOWA (JAnik, BOtkin, WAllis). Model został skonstruowany na początku lat siedemdziesiątych (Botkin et al. 1972).

Już w 1996 r. amerykanie H. H. Shugart i T. M. Smith (1996) doliczyli się około 40 różnych wersji modeli płatowych, zastosowanych do symulacji różnych typów lasów, reprezentujących zróżnicowane strefy roślinno-klimatyczne: od borealnej po tropikalną. Pod koniec lat osiemdziesiątych ubiegłego wieku modele płatowe pojawiły się także w Europie (Kienast 1987; Leemans, Prentice 1987; Bugmann 1994; Krauechi 1994; Lindner, Sievaenen, Pretsch 1997).

Perspektywiczne również może być zastosowanie modelu ZELIG (Smith, Urban 1988), zwłaszcza jako przykład integracji procesów na poziomie drzewa, drzewostanu i krajobrazu.

Założenia dotyczące zastosowania modelu płatowego w warunkach lasów polskich zostały przedstawione w pracach B. Brzezieckiego (1991; 1999). Brzeziecki (1999) rozdziela procesy symulowane w ramach modeli płatowych na dwie zasadnicze grupy. Pierwszą stanowią zjawiska zachodzące w środowisku drzewostanu. Należą tu między innymi warunki cieplne (niska temperatura w okresie zimy, występowanie przymrozków, długość sezonu wegetacyjnego, temperatura efektywna), bilans wodny siedliska (opady, ewapotranspiracja potencjalna i rzeczywista, ilość wody zawartej w glebie), cykl obiegu azotu i węgla w układzie „drzewostan – gleba", warunki świetlne w drzewostanie i na dnie lasu. Drugą grupę stanowią procesy o charakterze demograficznym zachodzące w drzewostanie: odnowienie naturalne, wzrost i wydzielanie się drzew.

Podział ten, jak określa sam autor, ma w dużym stopniu charakter umowny, ponieważ na przykład warunki świetlne, od momentu pojawienia się pierwszych drzew na symulowanej powierzchni, kształtują się pod wpływem samego drzewostanu. Podobnie jest z ilością dostępnego azotu, który ulega stałym przemianom w układzie „drzewostan – ściółka – gleba".

W modelu Brzezieckiego (Brzeziecki 1999) każdemu z ważniejszych procesów odpowiadają oddzielne moduły, mające względnie izolowany charakter i kontaktujące się z innymi modułami za pośrednictwem odpowiednich wejść i wyjść. Taka struktura modelu ułatwia jego rozwój i doskonalenie. W każdym momencie można zająć się wyodrębnionym procesem i odpowiadającym mu modułem bez konieczności dokonywania znaczących zmian w innych fragmentach modelu. Warto w tym kontekście wymienić także prace J. Szwagrzyka (1988; 1994) oraz model PICEAT, symulujący wzrost monokultur świerkowych w Sudetach (Pawłowski 1996).

Krokiem czasowym stosowanym w modelach płatowych jest przeważnie jeden rok. W tym kroku czasowym obliczane są zmiany podstawowych parametrów, na przykład przyrost pierśnicy drzew, wysokość, liczba drzew itp. Część obliczeń ma charakter stochastyczny. Uzyskuje się to przez dodanie składnika losowego do wartości obliczanych według przyjętych funkcji.

Długość okresu symulacji jest dowolna i zależy od wiarygodności dostępnych scenariuszy rozwoju warunków środowiska. W celu określenia dominujących trendów i uśrednienia wyników obliczenia powtarzane są dla większej liczby powierzchni próbnych.

Większość stosowanych na świecie modeli dynamiki lasów składa się z dwóch submodeli: osobnika i cenozy w całości (Oja 1985). Ma to logiczne wyjaśnienie w tym, że część parametrów zależy od indywidualnych właściwości roślin, a część od wzajemnego położenia w cenozie i krajobrazie. Tutaj warto przypomnieć symulacyjny model lasu różnowiekowego o zróżnicowanym składzie gatunków (Popadjuk et al. 1991; Czumaczenko 1992), który pozwala przejść od modelu osobnika poprzez model leśnej cenozy do modelu krajobrazu leśnego.

3.2. Struktura modeli FORKOME

Modele FORKOME1 i FORKOME2 (Menshutkin, Kozak 1997; Kozak, Menshutkin 1999; Kozak, Menshutkin 2000; Kozak, Menshutkin 2001) zostały opracowane w celu symulacji sukcesji zespołów leśnych z uwzględnieniem poszczególnych drzew. Specyfiką tego podejścia jest traktowanie lasu jako zespołu drzew, z których każde ma indywidualny zestaw cech (gatunek, wiek, pierśnica, wysokość itd.), i podlega indywidualnemu rozwojowi, zależnemu od różnych czynników ekologicznych, takich jak światło, temperatura, wilgotność, konkurencja między drzewami itd.

Konstrukcja modeli FORKOME1 i FORKOME2 oparta jest na modelu FORET (Shugart, West 1977). Warto nadmienić, że podstawowy zarys tego typu modeli został przedstawiony przez D. B. Botkina i współautorów już w 1972 r. (Botkin et al. 1972), a następnie wielokrotnie powtarzany, między innymi także w postaci modelu FORET.

Model FORET symuluje coroczne zmiany biomasy drzew wynikłe z przyrostu biomasy każdego z istniejących na tej powierzchni drzew oraz z pojawiania się tam siewek i ubywania drzew na skutek ich śmiertelności. Wszystkie te procesy są stochastyczne. Podstawowym założeniem przy konstrukcji tego modelu było przyjęcie opisu dynamiki rozwoju drzew poprzez takie równania, których parametry mają dobrą interpretację biologiczną, pozwalającą na wykorzystanie zarówno zgromadzonych dotąd danych pomiarowych, jak i wiedzy dotyczącej opisywanych procesów.

Ogólne zasady modelu FORET można znaleźć w pracy J. Szwagrzyka (1994), a porównanie tego modelu z innymi modelami płatowymi istniejącymi w literaturze przedstawiono w pracy A. Porté i H. H. Barteunka (2002).

W modelach FORKOME1 i FORKOME2 inaczej niż w modelu FORET rozwiązano zależności wodne, temperaturowe itp. Na przykład w bloku temperturowym zastosowano współczynniki umożliwiające symulację różnego rodzaju oscylacyjnych zmian temperatury. W bloku wodnym dodano bilans wodny i transpirację poszczególnych gatunków drzew.

W naszych modelach wydzielane są określone bloki. Jest to dobrze widoczne w modelu FORKOME1 na jednym z początkowych naszych algorytmów (rys. 3.5). Blok WSTAW PARAMETRY dotyczy oceny parametrów każdego drzewa i zbiorowiska leśnego. Do parametrów tych należały między innymi maksymalna średnica drzewa na standardowej wysokości 137 cm od ziemi, maksymalna wysokość, maksymalny wiek oraz minimalna i maksymalna suma temperatur powietrza.

Z uwagi na to, że model jest stochastyczny, badania jego dynamiki wymagają uwzględnienia wielu wariantów (blok NOWY WARIANT). Krokiem czasowym w modelu jest rok.

Model uwzględnia symulacje procesów śmiertelności, reprodukcji i wzrostu drzew w ciągu każdego roku. Śmiertelność drzew jest procesem losowym, który zależy od wieku drzewa i warunków wzrostu w poprzednim roku. Symulacja reprodukcji drzew przedstawiona jest w modelu jako stochastyczny proces uzależniony od gatunków siewek, warunków na powierzchni ziemi oraz średniej temperatury na poziomie ściółki. Tempo wzrostu każdego drzewa (blok WZROST) zależy od jego rozmiarów i gatunku oraz wody, warunków światła, temperatury i zaopatrzenia w pierwiastki biogenne. Po realizacji wszystkich wariantów modelu zostaje przeprowadzona analiza statystyczna weryfikująca otrzymane rezultaty (blok STATYSTYKA). W najprostszym przypadku analiza obejmuje obliczanie wartości średniej i odchylenia standardowego, a w bardziej skomplikowanych przypadkach dotyczy obliczeń seryjnych i funkcji krzyżowo korelacyjnych.

Schemat blokowy przedstawiony na rys. 3.5 został rozszerzony o dodatkowe bloki oraz możliwość wycinania drzew, zmian temperatury powietrza, ilości opadów atmosferycznych w możliwych scenariuszach zmian (Kozak, Menshutkin 2002). W modelu zastosowana została statystyczna metoda Monte-Carlo, pozwalająca na symulacje do 200 wariantów w każdym scenariuszu. Po przeprowadzeniu analizy Monte-Carlo program pozwala na pokazanie i wydrukowanie średniej liczby drzew i średniej biomasy drzew wraz z standardowym odchyleniem dla każdego roku.

W naszych modelach wzrost drzewa zależy od jego gatunku oraz osiągniętych do tej pory przez to drzewo rozmiarów, odniesionych do rozmiarów maksymalnych:

$$\delta\left(D^2H\right) = rLa\left(1 - \frac{DH}{D_{max}H_{max}}\right)$$

gdzie:

r – stała dla gatunku, określająca wydajność fotosyntetyczną aparatu asymilacyjnego,
La – względna powierzchnia listowia danego drzewa (m^2/m^2),
D – pierśnica drzewa na wysokości 1,30 m (cm),
H – wysokość drzewa (m),
D_{max} – maksymalna pierśnica dla danego gatunku (cm),
H_{max} – maksymalna wysokość dla danego gatunku (m),
δ(D2H) – przyrost objętości drzewa.

Konstrukcja równania wzrostu drzew pozwala na określenie parametrów liczbowych na podstawie stosunkowo łatwo dostępnych danych. Takimi parametrami są pierśnica, wysokość i wiek oraz maksymalna pierśnica, maksymalna wysokość i maksymalny wiek możliwe do osiągnięcia przez drzewa danego gatunku.

W literaturze leśniczej i ekologicznej często podawane są informacje o tych cechach, lecz – jak pisze B. Brzeziecki (Brzeziecki 1999) – są to oceny szacunkowe. Dane podawane przez poszczególnych autorów mogą się różnić. Częściowo dane odnoszą się do drzew rosnących poza lasem, na przykład w parkach, przy drogach itd. W takich warunkach niektóre gatunki mogą osiągać rozmiary (dotyczy to zwłaszcza pierśnicy) bardzo mało prawdopodobne w warunkach zwartego drzewostanu. Podobnie przedstawia się sprawa z maksymalnym wiekiem drzew. Jedynie w przypadku wysokości można założyć, że podawane w literaturze wielkości maksymalne zostały zaobserwowane u drzew rosnących w zwarciu, a więc w lesie, bo tylko w takim przypadku drzewa „muszą" maksymalnie rosnąć w górę, walcząc o utrzymanie się w dachu koron i o swobodny dostęp do światła (Brzeziecki 1999).

Celem otrzymania wiarygodnych szacunków wspomnianych parametrów uwzględniono jak największa liczbę źródeł (Karpiński 1949; Faliński, Herezniak 1977; Ellenberg 1986; Leibundgut 1991; Bugala 1991; Prentice, Helmisaari 1991; Dengler 1992; Pacyniak 1992; Bugmann 1994; Kraeuchi 1994; Tomanek 1994).

W obliczeniach zastosowano „zaokrąglone" wartości parametrów, operując bardziej klasami niż wartościami parametrów (Bugmann 1994).

Wysokość drzewa opisana została następującym równaniem:

$$H = 130 + b_2 D - b_3 D^2$$

gdzie:

D – pierśnica drzewa na wysokości 1,30 m (cm),
H – wysokość drzewa (m),
b_2, b_3 – parametry wyliczone dla każdego gatunku.

Wartości parametrow b_2, b_3 wyrażone zostały wzorami (Botkin et al. 1972):

$$b_2 = 2\left(\frac{H_{max} - 130}{D_{max}}\right)$$

$$b_3 = \left(\frac{H_{max} - 130}{D^2{}_{max}}\right)$$

Liczbowe wartości tych parametrów, obliczone dla poszczególnych gatunków drzew (Brzeziecki 1999), zawarte są w tab. 3.2 i 3.3 .

Tab. 3.1. Wielkość parametrów określających maksymalne wymiary: H_{max} – wysokość (m), D_{max} – pierśnica (cm) poszczególnych gatunków drzew, uwzględnianych w obecnej wersji modelu.

GATUNEK	WYSOKOŚĆ	PIERŚNICA
Abies alba Miller	60	150
Acer pseudoplatanus L.	40	150
Alnus incana (L.) Moench	25	50
Betula pendula Roth	35	100
Carpinus betulus L.	35	100
Castanea sativa Miller	35	250
Fagus silvatica L.	45	150
Fraxinus excelsior L.	40	150
Larix decidua Miller	50	150
Picea abies (L.) Karsten	55	150
Pinus sylvestris L.	45	150
Populus tremula L.	40	100
Quercus robur L.	45	250
Tilia cordata Miller	45	250

Tab. 3.2. Parametry równań wzrostu drzew: b_2 i b_3.

GATUNEK	b_2	b_3
Abies alba Miller	78,3	0,261
Acer pseudoplatanus L.	51,6	0,172
Alnus incana (L.) Moench	77,4	0,387
Betula pendula Roth	67,4	0,337
Carpinus betulus L.	67,4	0,337
Castanea sativa Miller	27,0	0,054
Fagus sylvatica L.	58,3	0,194
Fraxinus excelsior L.	51,6	0,179
Larix decidua Miller	64,9	0,216
Picea abies (L.) Karsten	71,6	0,239
Pinus silvestris L.	58,3	0,194
Populus tremula L.	77,4	0,387
Quercus robur L.	35,0	0,070
Tilia cordata Miller	35,0	0,070

Wzrost zależy od takich czynników ekologicznych jak światło, temperatura, pierwiastki biogenne, woda i inne. Ilość światła, jakie dociera do danego drzewa, jest obliczana jako ubytek radiacji słonecznej spowodowany przez sumaryczne zacieniające oddziaływanie powierzchni liści wyższych drzew:

$$Q(h) = Q_{max} E^{-0.25 LA(h)}$$

gdzie:

LA(h) – łączna powierzchnia liści wyższych drzew (m^2/m^2), jako funkcja wysokości,
Q_{max} – radiacja słoneczna dochodząca do górnej powierzchni koron ($kkal/cm^2$),
Q(h) – radiacja na określonej wysokości (h),
-0.25 – stała.

W modelach zastosowane zostały dwa równania do obliczenia współczynnika redukcji wzrostu związanego z warunkami świetlnymi. Pierwsze dla drzew światłolubnych:

$$r = 1 - e^{-1.136[Q(h) - 0.08]}$$

drugie dla cienioznośnych:

$$r = 1 - e^{-4.64[Q(h) - 0.05]}$$

gdzie:

r – współczynnik świetlnej redukcji wzrostu.
Q(h) – radiacja na określonej wysokości (h)

Wzrost drzewa jest w modelach funkcją warunków termicznych. Zastosowano następujący wzór (Botkin et al. 1972), w którym głównym czynnikiem wpływającym na wzrost drzew jest suma temperatur efektywnych (wyższych niż 5°C):

$$T = \frac{4(DGD - DGD_{min})(DGD_{max} - DGD)}{(DGD_{max} - DGD_{min})^2}$$

gdzie :

T – współczynnik określający wpływ klimatu na wzrost,
DGD – suma temperatur efektywnych dla danego miejsca,
DGD_{min} – minimalna suma temperatur efektywnych konieczna dla występowania gatunku,
DGD_{max} – maksymalna suma temperatur efektywnych konieczna dla wzrostu gatunku.

Model FORKOME1 pozwala na obliczenia zmian klimatu następujących w trakcie symulacji za pomocą współczynników zmiany stopniodni: K_0, K_1, K_2, K_3. Dla sumy stopniodni (sumy temperatur efektywnych, to znaczy temperatur przekraczających 5°C w ciągu okresu wegetacji) zaproponowany jest następujący wzór:

$$K = K_0(1 + K_1 t) + K_2 \mathrm{Sin}\left(\frac{2 * 3{,}14}{K_3} t\right)$$

gdzie:

K – suma temperatur efektywnych,
K_0 – współczynnik stały,
K_1 – kierunek zmian sumy temperatur efektywnych,
K_2 – amplituda oscylacji sumy temperatur efektywnych,
K_3 – okres oscylacji sumy temperatur efektywnych,
t – czas (w latach).

W nawiązaniu do wspomnianych już współczynników temperaturowych dodajmy, że dotyczą one zmian klimatu w rejonach badań. W proponowanej obecnie wersji, niezbędne są informacje o średnich wieloletnich i odchyleniach standardowych średnich temperatur, minimalnej i maksymalnej dla poszczególnych miesięcy roku. Dane te pochodzą ze stacji meteorologicznych położonych najbliżej rejonu badań. Informacje te pełnią rolę danych wejściowych, będących podstawą dalszych obliczeń.

Dla bloku pierwiastków biogennych wykorzystano następującą funkcję:

$$GMF = a + b[RNA] + c[RNA]^2$$

gdzie:

a, b, c – współczynniki wyliczone na podstawie zebranych danych,
GMF – współczynnik modyfikacji wzrostu,
RNA – parametr związany z ograniczeniem wzrostu drzew przez pierwiastki biogenne w glebie.

Wartość RNA jest tym mniejsza, im większa jest łączna biomasa drzew, a zarazem im bardziej pierwiastki biogenne są wyczerpywane z gleby. Przyjmuje się ze w danych warunkach glebowych i ekologicznych biomasa ta nie może przekroczyć pewnej maksymalnej wartości i ze wówczas:

$$RNA = 1 - \frac{B}{B_{max}}$$

gdzie:

B – rzeczywista biomasa drzewostanu
B_{max} – maksymalna biomasa drzewostanu

przy tym $$B = 0{,}1193 \sum_{i=1} D_i^{2.393}$$

Obliczano także prawdopodobieństwo obumarcia drzewa.

Jeżeli $D^{t+1} - D^t < 0{,}1$ cm (gdzie D^t, D^{t+1} to pierśnice w kolejnych latach), wtedy $P_n = 0{,}368$ (gdzie P_n to prawdopodobieństwo obumarcia drzewa) oraz

$$P_n = 1 - \left(1 - \frac{4{,}605}{AGE_{max}}\right)^n$$

gdzie AGE_{max} to maksymalny wiek drzewa.

W modelu FORKOME2 została również uwzględniona transpiracja liści drzew w zależności nie tylko od parametrów meteorologicznych, jak jest to w innych modelach płatowych, a także od gatunku drzewa. Do modelu również zostały wprowadzone zależności między gatunkami drzew a poziomem wód gruntowych oraz zależności tempa wzrostu drzew od ilości wody w glebie (Kozak et al., w druku). Podstawę tego bloku tworzy równanie bilansu wodnego:

$$W(t+1) = W(t) + Prec(t) - Trans(t) - Evapor(t)$$

gdzie:

W(t) – ilości wody w glebie w jednostce czasu t,
Prec(t) – opady atmosferyczne,
Trans(t) – transpiracja,
Evapor(t) – parowanie wody z powierzchni gleby.

3.3. Przykładowe zastosowanie modeli FORKOME

Konstrukcja modeli umożliwia ich zastosowanie w możliwie jak najszerszym zakresie zmienności warunków klimatycznych, glebowych i drzewostanowych. W modelach uwzględniono zarówno gatunki drzew leśnych o dużym znaczeniu lasotwórczym, jak i typowe gatunki domieszkowe.

Perspektywicznym kierunkiem wykorzystania modeli jest przeprowadzanie za ich pomoca różnych eksperymentów symulacyjnych. Wyniki takich eksperymentów krótko przedstawimy w niniejszym paragrafie.

Najpierw podamy przykłady wyników otrzymanych w eksperymentach symulacyjnych w Bieszczadach Polskich w zlewni potoku Głęboki oraz w zlewni potoku Rika w Beskidach Ukraińskich (rys. 3.6), na transektach (rys. 3.7).

Tendencja cyklicznego rozwoju zbiorowisk leśnych została już opisana w literaturze przez H. Shugarta (Shugart 1984) dla lasów amerykańskich. Jest ona zjawiskiem charakterystycznym również dla lasów w Bieszczadach Polskich i w Beskidach Ukraińskich. Wobec tego odtworzenie tej cykliczności w symulacjach komputerowych byłoby ważną przesłanką jakościowej weryfikacji modelu.

Taka cykliczna zmiana jest prognozowana dla powierzchni aktualnie zajętych przez drzewostany bukowe (rys. 3. 8). Wycięcie drzew nie zmienia ogólnej tendencji cyklicznych zamian między bukiem a jodłą w aspekcie ich biomasy i liczebności. Jedynie czas takich zamian między bukiem a jodłą będzie się zmieniał.

Na powierzchni jodłowej (rys. 3.9) prognozowano, że jodłowy drzewostan w średnim wieku 72 lat stanowi stadium leśnej sukcesji na drodze do odnowienia lasu bukowego. Model pozwala przewidywać, że po 100 latach (czasu modelowego) dla liczebności (rys. 3.10) i po 200 latach dla biomasy las jodłowy zmieni się na las bukowy.

Wyniki symulacji pokazują także obserwowaną nietrwałość drzewostanów świerkowych posadzonych w Bieszczadach przez człowieka. Takie symulowane drzewostany (ich absolutny wiek to 40-50 lat), jak pokazują wyniki symulacji, zanikają w ciągu kolejnych 20-30 lat. Na ich miejscu powstają lasy bukowe. W symulowanym procesie regeneracji lasów bukowych, na miejscu antropogennych wtórnych lasów świerkowych (rys. 3.11), widać bardzo szybkie zanikanie świerka i powracanie buka.

W nadchodzących latach w lasach strefy klimatu umiarkowanego przewidywane są zmiany struktury zespołów roślin, powodowane głównie globalnymi zmianami klimatu. Dlatego należy podjąć badania nad wpływem procesów klimatycznych i hydrologicznych na sukcesje w lasach naturalnych i antropogennych. Modele FORKOME pozwalają na symulację skutków ocieplenia klimatu – wtedy w Karpatach prognozowana jest dominacja buka (rys. 3.12b), natomiast przy spadku temperatury powietrza spodziewana jest dominacja jodły (rys. 3.12c).

A co w lasach równinnych? Spróbujmy także pokrótce przedstawić wyniki symulacji komputerowej sukcesji lasu sosnowego na równinie na przykładzie Kampinoskiego Parku Narodowego* w różnych scenariuszach zmian klimatycznych z wykorzystaniem modelu FORKOME2 (Kozak et al., w druku).

Tak naprawdę bardzo mało wiadomo o możliwych zmianach lasów dominujących w Puszczy Kampinoskiej, jak też i w innych regionach Polski. Biorąc pod uwagę fakt, że średnie roczne temperatury powietrza oraz roczne ilości opadów atmosferycznych wykazują na terenie Kampinoskiego Parku Narodowego znaczące wahania (na przykład średnia roczna temperatura powietrza w 1996 r. wynosiła 3,5°C, podczas gdy średnia w latach 1986-1993 miała wartość 7,9°C; ilość opadów w 1996 r. osiągała 662,2 mm, podczas gdy średnia opadów w latach 1986-1993 była równa 501,7 mm – Wierzbicki 1999), uznaliśmy za celową symulację według 4 niżej opisanych scenariuszy możliwych zmian klimatu względem scenariusza „kontrolnego”:

K o n t r o l a – w aktualnych warunkach Kampinosu: średnio roczna temperatura powietrza 7,6°C i opady atmosferyczne 550 mm.

* Pragniemy podziękować dyrekcji i pracownikom Kampinoskiego Parku Narodowego za pomoc okazaną przy prowadzeniu badań.

S c e n a r i u s z p i e r w s z y – ciepło i wilgotnie: zwiększenie o 2°C temperatury i zwiększenie o 200 mm opadów atmosferycznych.

S c e n a r i u s z d r u g i – ciepło i sucho: zwiększenie o 2°C temperatury i zmniejszenie o 200 mm opadów atmosferycznych.

S c e n a r i u s z t r z e c i – zimno i wilgotnie: zmniejszenie o 2°C temperatury i zwiększenie o 200 mm opadów atmosferycznych.

S c e n a r i u s z c z w a r t y – zimno i sucho: zmniejszenie o 2°C temperatury i zmniejszenie o 200 mm opadów atmosferycznych.

Według podanych scenariuszy przeprowadzono także przykładową symulacje sukcesji sosnowego lasu w Kampinosie.

Na podstawie analizy rysunków z symulacji wykonanej według scenariusza kontrolnego (rys. 3.13) można sądzić, że symulowany rozpad starych sosen w ciągu 100 lat spowoduje zwiększenie biomasy dębów. Dominującą biomasę ma dąb szypułkowy, natomiast jego udział w liczebności jest dość zmienny. Ten gatunek pozwala na dość wysoki udział gatunków domieszkowych, które periodycznie oscylują w modelowanym płacie drzewostanu.

Ale wymagający Czytelnik może powiedzieć, że prognozowane w tej jednej realizacji zmiany, zwłaszcza biomasy drzew, mogą być przypadkowe. Dlatego nasz program pozwala zrobić do 20 realizacji i przedstawić uśrednione wartości za pomocą metody Monte-Carlo. Jest to metoda modelowania statystycznego (Shugart 1984) zastosowana przez nas do analizy biomasy.

W tym celu spróbujmy dalej analizować tendencje zmian biomasy drzew na badanej powierzchni przy użyciu metody Monte-Carlo.

Graficzne przedstawienie wyniku zastosowania metody Monte-Carlo dla zmian biomasy drzew na badanej powierzchni lasu sosnowego w Kampinoskim Parku Narodowym (rys. 3.14) potwierdza zmianę lasu sosnowego na las dębowy. W ciągu początkowych symulowanych 100 lat model pokazuje przechodzenie dominacji biomasy od sosny zwyczajnej do dębu szypułkowego. Następnie ten gatunek dębu cały czas utrzymuje dominację biomasy. Jeżeli do 100 lat jego biomasa zwiększa się kosztem zmniejszenia biomasy i zanikania sosny, to od 100 do 500 lat jego biomasa ma tendencje do nieznacznego zmniejszenia na korzyść zwiększenia biomasy lipy (żółty kolor na rysunku). Jak pokazują wyniki z dwódziestokrotnej symulacji, procentowy udział dębu bezszypułkowego zwiększa się od kilku procent na początku symulacji do 10% w 200 roku symulacji. Później jego udział prawie się nie zmienia.

Jako przykład przedstawimy wyniki symulacji tylko w scenariuszu pierwszym przy ociepleniu i uwilgotnieniu klimatu (wyniki symulacji dla innych scenariuszy zostały dokladnie opisane w publikacji Kozak et al. 2003 w druku). W scenariuszu pierwszym model przewiduje szybsze i gwałtowniejsze zaniknięcie biomasy sosny w porównaniu do kontroli. W tym scenariuszu zwiększenie udziału biomasy dębu bezszypułkowego prawie do 45% pod koniec symulacji, gdy gatunek ten dominuje, a dąb szypułkowy zajmuje pod względem tej dominacji drugie miejsce (rys. 3.15). Czyli w tym scenariuszu w porównaniu do kontroli model prognozuje zmniejszenie biomasy dębu szypułkowego i lipy oraz zwiększenie biomasy dębu bezszypułkowego, jaworu, osiki i grabu.

Warto zauważyć, że otrzymane przez nas wyniki symulacji przeprowadzonej dla Bieszczad i Puszczy Kampinoskiej są zgodne z tezą wysuniętą przez D. G. Sprugella (1991), który uważa, że stosunkowo niewielkie zmiany temperatury i opadów mogą spowodować znaczące i długotrwale zmiany w roślinności.

Zmiany klimatyczne będą jednym z głównych problemów, jakim będziemy musieli stawić czoło w nadchodzących latach. Istnieje więc także potrzeba rozwoju narzędzi przewidujących potencjalny wpływ zmian klimatu na ekosystemy. Problemy te już są podejmowane w literaturze (Krauchi 1994; Hong et al. 1999; Brzeziecki 1999). Na przykład symulacje dotyczące powierzchni badawczej Solling w Niemczech za pomocą modelu FORSUM pokazały, że świerk całkowicie zniknie z obszaru Solling i zostanie zastąpiony przez las typu *Luzulo Fagetum* z gatunkami *Acer platanoides*, *Quercus petraea*, *Fagus sylvatica* (Krauchi 1994).

3.4. Perspektywy rozwoju modeli

Dalsza rozbudowa i wykorzystania modeli FORKOME może, według nas, polegać na uwzględnieniu w modelu kolejnych zależności i parametrów oraz na weryfikacji modelu w różnych regionach Polski i państw sąsiednich.

Te dodatkowe parametry mogą dotyczyć dokładniejszej i bardziej szczegółowej klasyfikacji siedlisk (mamy tutaj na myśli klasyfikację uwzględniającą, na przykład, różne warianty temperatury, wilgotności, rodzaje gleby, siedliska itd.) czy na przykład opisu ontogenicznych etapów rozwojowych drzew. W tym ostatnim aspekcie model może wyglądać bardziej naturalnie, ponieważ będą w nim nie tylko drzewa określonej wysokości czy wieku, ale również określonego stadium rozwojowego (na przykład od form juwenilnych do generatywnych starych).

Można też myśleć o innym wariancie, który polegałby na zastąpieniu typów siedliskowych lasu zespołami i podzespołami leśnymi, zdefiniowanymi za pomocą kryteriów fitosocjologicznych z zastosowaniem różnych klasyfikacji, na przykład zgodnej z gatunkami dominującymi klasyfikacji biogeocenotycznej (Osnowy 1964) czy klasyfikacji na podstawie charakterystycznych gatunków roślin według Brauna-Blanqueta (1951).

Perspektywicznie wygląda zmniejszenie charakteru statycznego tych czy innych klasyfikacji i próba wprowadzenia do modelu parametrów zmieniających się w czasie dla odrębnych jednostek klasyfikacyjnych. Tego rodzaju zmiany są z pewnością ważne zarówno dla lasów naturalnych (Bernadzki et al. 1997, 1998; Brzeziecki, Zybura 1998), jak i antropogennych (Kowalski 1992). Mamy dość dużo przykładów takich transformacji dla lasów polskich, na przykład: zmiana siedlisk bagiennych i wilgotnych w mniej wilgotne na skutek zmian klimatu i działań melioracyjnych, wzrost żyzności siedlisk jako przykład regeneracji po ustąpieniu działania czynników degradacyjnych (Faliński 1986) oraz zanieczyszczenia powietrza, wód i gleby substancjami pochodzenia przemysłowego lub rolniczego („kwaśne" deszcze, tlenki azotu). Zmiana typu siedliska o jedną klasę (wzdłuż gradientu żyzności lub wilgotności) jest wystarczająca, aby spowodować zmiany istotne z punktu widzenia symulacji dynamiki krajobrazów leśnych (Brzeziecki 1999).

Perspektywiczne także wydaje się ulepszenie w modelu algorytmu obliczania bilansu wodnego siedliska. Jest to bardzo istotne dla Polski, zwłaszcza z uwagi na możliwość wykorzystania modelu do różnych siedlisk. W literaturze poświęconej modelom płatowym można już znaleźć pierwsze próby rozwiązania tego problemu (Botkin 1993; Kraeuchi 1994). Obiecujące jest wprowadzenie do bloku wodnego, jak pokazały nasze pierwsze wyniki, parametrów dotyczących realnych pomiarów transpiracji drzew. Należałoby spróbować wprowadzić do modelu tego rodzaju adaptacje, które znacznie zwiększyłyby stopień jego ogólności i zakres potencjalnych zastosowań, tym bardziej że problem zmian w bilansie wodnym jest bardzo aktualny nie tylko dla lasów górskich, ale i dla lasów nizinnych, co dobrze widać na przykładzie Puszczy Kampinoskiej w okolicach Warszawy.

Weryfikacja modelu w różnych warunkach siedliskowych pozwoli w przyszłości prognozować zmiany dla większej ilości realnie istniejących płatów, symulując proces dynamiki drzewostanu w sukcesji wtórnej lub regeneracyjnej (Faliński 1986, 1988).

Obiecująco już teraz rysują się perspektywy rozwoju i wykorzystania modelu do analizy krajobrazu, zwłaszcza kiedy patrzymy na krajobraz jako na kompozycje płatów w różnorodnych zmianach sukcesyjnych (Forman, Gordon 1982).

Jak wiadomo, w modelach płatowych nie uwzględnia się na ogół przestrzennego rozmieszczenia drzew w obrębie powierzchni. Z jednej strony upraszcza to przebieg obliczeń, ale z drugiej strony nie pozwala na realną analizę struktury powierzchni. W przyszłości można by uwzględniać przestrzenne rozmieszczenia drzew z dodaniem do modelu nie tylko współrzędnych pnia, ale i projekcji korony itd.

Model można wykorzystać w kierunku sprawdzania wymagań ekologicznych poszczególnych gatunków drzew i analizy zależności drzew od czynników środowiska, stosując przy tym wyliczenia różnego rodzaju współczynników. Wszystko to stwarza między innymi możliwość eksperymentowania i badania wpływu różnych wartości parametrów w procesie sukcesji. Jednym z aspektów może być także badanie parametrów modelu na podstawie wyliczenia współczynników czułości (Jørgensen 1994).

4. Modelowanie z zastosowaniem automatów komórkowych

4.1. Podstawowe pojęcia

Modele płatowe nie są wystarczające, gdy badamy większe obszary przewyższające rozmiary płatów drzewostanowych; mogą to być na przykład zlewnie rzek. Powstaje wtedy pytanie: w jaki sposób charakteryzować procesy zachodzące nie tylko na konkretnej zbadanej powierzchni, ale i poza nią. Wysoki stopień przestrzennego zróżnicowania ekosystemów leśnych powoduje, że badanie prawidłowości i mechanizmów rozwoju lasów wymaga bardzo dużej liczby powierzchni badawczych, co pociąga za sobą wysokie koszty i nie zawsze jest możliwe do wykonania. Rozwiązaniem tych trudności może być zastosowanie automatów komórkowych (Logofet 1999).

Już od dość dawna praktyka badań krajobrazowych (Armand 1975) spowodowała wykorzystanie w modelowaniu teorii automatów komórkowych. Modelowanie przestrzenne rozmieszczenia roślin, jako część badań krajobrazowych, bardzo rozwinęło się wraz z postępem kartografii satelitarnej i geograficznych systemów informacyjnych (GIS – Geografic Information System) (Frelich, Lorimer 1991; Williams 1996).

Nasze liczne rozmowy z kolegami biologami i ekologami pokazały dowodnie, że pojęcie a u t o m a t u kojarzone jest często ze wszystkim (pralką, bronią palną, sprzedażą napojów chłodzących itd.), ale bardzo rzadko z logiką matematyczną i teorią automatów komórkowych. Ponieważ znajomość tego pojęcia będzie konieczna dla zrozumienia przykładów zastosowań takich automatów w modelowaniu krajobrazów ekologicznych (w końcu tego rozdziału i w rozdziale 10 drugiej części książki), zaczniemy od bardzo prostego, konkretnego przykładu.

Załóżmy, że mamy do czynienia z pewnym kawałkiem uprawnej ziemi, powiedzmy działkowym ogródkiem. Możemy tam coś posadzić, na przykład ogórki, ale możemy też nie sadzić niczego. W pierwszym wypadku za rok możemy uzyskać pewien plon, a w drugim – oczywiście – niczego nie uzyskamy. Mówiąc inaczej: nasz system (ogródek) może znajdować się w dwu różnych stanach: stan 1 – są ogórki, stan 2 – nie ma ogórków. Przejście ze stanu 2 do stanu 1 odpowiada operacji posadzenia ogórków, a przejście ze stanu 1 do stanu 2 – sprzątnięciu plonu lub po prostu nadejściu kolejnego roku; nie zebrany plon ulegnie likwidacji.

Graficzna prezentacja omawianej sytuacji znajduje się na rys. 4.1, na którym stany oznaczone są kółkami, a przejścia – strzałkami. Zauważmy, że taki rysunek nazywa się grafem, a cały rozdział współczesnej matematyki nosi nazwę teorii grafów. Grafy są bardzo poglądowym sposobem prezentacji. Spójrzmy uważnie na rys. 4.1. Gospodarując bardzo regularnie, po zbiorze ogórków (gdy mamy stan 2 w kółku) stale sadzimy znów rozsadę (i przechodzimy do stanu 1 w kółku). A co będzie, jeżeli rozsady nie posadzimy? Oczywiście nie będzie łuku (2), a nastąpi pętla (1), czyli po upływie roku możemy ogórki posadzić i w ten sposób „wejść" na łuk (2), albo nie sadzić i „wejść" znów na pętlę (1). Symetryczna sytuacja wystąpi ze stanem (2) i pętlą (2) – po zbiorze plonu.

Dla dalszych rozważań lepszym sposobem jest przedstawienie tych informacji w postaci tak zwanej macierzy przejść. Taką macierzą nazywamy tablicę (tab. 4.1), w której wiersze odpowiadają stanom, a kolumny przejściom. W komórki tablicy wpisywane są te stany, które wynikają z położenia w danym wierszu i odpowiadają danemu przejściu (kolumna tablicy).

Tab. 4.1. Macierz przejść w najprostszym automacie.

	1 – niczego nie robimy, albo jedynie zbieramy plon	2 – sadzimy ogórki
1 – Ogórków nie ma	1	2
2 – Ogórki są	1	2

Wracając do naszego przykładowego ogródka: jeżeli niczego w nim nie ma i my nic nie robimy, to w następnym roku w ogródku będzie pusto (komórka 1,1). Jeżeli w ogródku były ogórki, ale nie zebraliśmy ich i w następnym roku niczego nie posadziliśmy, w ogródku też będzie pusto (komórka 2,1). Niezależnie od tego, czy w poprzednim roku były w ogródku ogórki, czy ich nie było, jeżeli ogórki w tym roku posadzimy, to one tam będą i dadzą jakiś plon (komórki 1,2 i 2,2).

To, co powyżej zostało opisane za pomocą rys. 4.1 i tab. 4.1, nazywa się automatem skończonym (słowo „skończony" dodajemy dlatego, że liczba stanów może być nieskończenie większa, a wtedy będzie to zupełnie inna matematyka). Aby uruchomić nasz automat, należy jeszcze określić stan automatu – w naszym przykładzie najprościej jest założyć, że zaczynamy od pustego ogródka (stan 1). Teraz zbadajmy, co będzie się działo w naszym ogródku, jeżeli w ciągu dwóch kolejnych lat będziemy sadzić ogórki, następnie przez dwa lata nie sadzić, później dwa lata sadzić itd. Oczywiście wynik możemy podać od ręki, bez użycia teorii automatów!

Bądźmy jednak formalistami i każdy krok czasowy wykonajmy z odwołaniem do tab. 4.2.

Tab. 4.2. Kolejne oddziaływania i stany najprostszego automatu.

Czas	0	1	2	3	4	5	6
Oddziaływanie		2	2	1	1	2	2
Stan	1	2	2	1	1	2	2

Wniosek praktyczny będzie trywialny: aby spodziewać się plonu ogórków, trzeba koniecznie zasadzić w ogrodzie ogórki. Natomiast istotne dla nas jest prześledzenie na przykladzie powyższej tablicy działania automatu skończonego.

Zwróćmy uwagę, że powyższy przykład automatu jest deterministyczny: każdemu stanowi wyjściowemu i każdemu oddziaływaniu (wejściu automatu) odpowiada jeden i tylko jeden stan następny. Takimi właśnie automatami deterministycznymi są na przykład komputer i pralka. Po komendzie Run komputer zacznie wykonywać program, a po włączeniu pralki zacznie ona prać; jeżeli to nie nastąpi, to komputer i pralka są zepsute.

W naturze sprawy przebiegają zupełnie inaczej. Posadzenie ogórków wcale nie oznacza, że zbierzemy plon – pogoda może być w lecie zimna i deszczowa, rozsada mogła być kiepska, a urodzaj mógł zostać rozkradziony. Rys. 4.2 przedstawia graf przejść automatu, w którym nieuniknione są zdarzenia nieprzewidywalne, i możemy jedynie wskazać, jakie jest prawdopodobieństwo zajścia danego przejścia. Tak więc prawdopodobieństwo zebrania plonów, pod warunkiem, że ogród był uprawiany w ubiegłym roku (pętla przy stanie 2), wynosi 0.9. Jeżeli jednak ogród był zapuszczony, prawdopodobieństwo plonu spada do 0.7 (przywrócenie zdatności uprawowej ogrodu sprawi pewien kłopot).

Tab. 4.3. Macierz przejść automatu stochastycznego.

	1 – Ogórków w tym roku nie było (albo plon przepadł?)	2 – Ogórki w tym roku były; plon zebrano
1 – Ogórków w ubiegłym roku nie było	**0.3**	**0.7**
2 – Ogórki w ubiegłym roku były	**0.1**	**0.9**

Pożyteczne wydaje się nam pewne rozszerzenie naszego przykładu. Sytuacja przedstawiona na rys. 4.2 może być rozpatrywana jako ł a ń c u c h M a r k o w a. Jeżeli będziemy wytrwale pracować rok w rok w naszym ogrodzie, to graniczne (finalne) prawdopodobieństwa naszego sukcesu (stan 2) lub porażki (stan 1) możemy łatwo obliczyć, rozwiązując układ liniowych algebraicznych równań, z którymi zetknęliśmy się w szkole średniej.

Zasady tworzenia takich równań są następujące: w lewej części graniczne prawdopodobieństwo p_1 jest mnożone przez prawdopodobieństwo przepływów z danego stanu, a po prawej prawdopodobieństwo przepływów prowadzących do danego stanu pomnożone jest przez odpowiednie graniczne prawdopodobieństwo p_2.

$$0.7\ p_1 = 0.1\ p_2$$
$$p_1 + p_2 = 1$$

Przypominamy, jak to należy zrobić:

$$p_2 = (0.7/0.1)\ p_1 = 7\ p_1$$
$$p_1 + 7\ p_1 = 1$$
$$8\ p_1 = 1$$
$$p_1 = 1/8 = 0.125$$

Wynik: $p_1 = 1/8 = 0.125$, $p_2 = 7/8 = 0.875$ – całkiem niezły, biorąc pod uwagę wszystkie kłopoty, jakie dokuczają ogrodnikowi.

Teoria automatów skończonych i łańcuchów Markowa jest raczej sprawą skomplikowaną i przywołaliśmy ją tutaj na tym prościutkim przykładzie jedynie dla pobieżnego zorientowania Czytelników w istocie problemu.

Przejdźmy do przykładu nieco bardziej złożonego, a co ważniejsze – ilustrującego realny problem badawczy (czego nie można powiedzieć o „uprawie ogórków”) – problem sukcesji w zespole roślinnym.

Tab. 4.4. Macierz przejść automatu stochastycznego imitującego sukcesję zespołu roślinnego.

STANY ZESPOŁU ROŚLINNEGO	1	2	3	4	5
1. Zespół łąkowy	0.3	0.7	0.0	0.0	0.0
2. Zespół krzaków	0.3	0.0	0.7	0.0	0.0
3. Młody las	0.3	0.4	0.0	0.3	0.0
4. Dojrzały zwarty drzewostan	0.5	0.0	0.0	0.0	0.5
5. Dojrzały drzewostan różnowiekowy	0.3	0.0	0.0	0.3	0.4

Graniczne prawdopodobieństwa stanu zespołu przy takiej macierzy przejść (tab. 4.4) wynoszą: $p_1 = 0.317$, $p_2 = 0.308$, $p_3 = 0.216$, $p_4 = 0.087$, $p_5 = 0.072$.

Zespół roślinny może znajdować się w pięciu różnych stanach: (1) zespół łąkowy, (2) zespół krzaków, (3) młody las, (4) dojrzały zespół leśny –

zwarty drzewostan, (5) dojrzały zespół leśny różnowiekowy – z polanami na miejscu obumarłych lub ściętych drzew. Spróbujmy przedstawić to w postaci obrazków (rys. 4.5).

Przyjmijmy, że naturalny szereg sukcesyjny (ciągłe linie ze strzałkami na rys 4.5) przechodzi od stanu 1 do stanów 2, 3 i 4 i dalej oscyluje między stanami 4 i 5. Przyjmijmy, że taki naturalny szereg sukcesyjny odpowiada pierwszemu (X = 1) zewnętrznemu oddziaływaniu na układ, kiedy brak jest oddziaływań antropogennych. Drugi wariant zewnętrznego oddziaływania (X = 2) obejmuje umiarkowany wyrąb lasu (przerywane linie ze strzałkami na rys. 4.5). W tym przypadku stan (5) będzie trwały, ponieważ naturalne przerzedzanie się lasu zostanie zastąpione przez selektywny wyrąb. Wyrąb młodego lasu (stan 3) doprowadzi do jego wyginięcia i przejścia do zespołu krzaków (stan 2). Wycinanie krzaków prowadzi do odtworzenia zespołu łąkowego (stan 1), który przy systematycznym wycinaniu krzaków będzie stanem stałym (na rys. 4.5 pętla przy kółku oznaczającym zespół łąkowy).

Trzeci wariant oddziaływania zewnętrznego (X = 3) odpowiada wyrębowi całkowitemu (linie punktowane na rys. 4.5). Przy takim wyrębie trwałym będzie jedynie stan zespołu łąkowego.

Opisanym powyżej zdarzeniom można nadać postać tablicy lub macierzy przejść (tab. 4.5). Zapełnienie przekątnej macierzy odpowiada stanom układu stałym w czasie. W komórkach, stanowiących elementy macierzy, zapisane są te stany wejść, przy których możliwe jest przejście ze stanu wyjściowego (wiersz macierzy) do stanu docelowego (kolumna macierzy).

Tab. 4.5. Macierz przejść automatu.

Stan docelowy		1	2	3	4	5
Stan wyjściowy	1	3	1,2			
	2	2,3		1		
	3	3	2		1	
	4	3				1,2
	5	3			1	2

Omawiany układ ma wyjście w postaci ilości wywiezionego z lasu drewna. Oczywiście przy braku wyrębu wyjście będzie zawsze zerowe (Y = 0). Przy wyrębie umiarkowanym (X = 2) maksymalną ilość wywożonego drewna można uzyskać w przypadku, kiedy zespół znajduje się w stanie 4: zwarty drzewostan; oceńmy wyjście w takim przypadku jako Y = 2 (umownych jednostek). Przy wyrębie całkowitym ilość wywiezionego drewna będzie większa. Temu wszystkiemu można nadać postać tablicy albo macierzy wyjść (tab. 4.6). W komórkach, stanowiących elementy macierzy, zapisane są stany wyjść, przy odpowiadającym im stanom wejść (wiersz matrycy) i stanowi automatu (kolumna matrycy).

Tab. 4.6. Tablica automatu skończonego.

Stan automatu		1	2	3	4	5
Stan wejścia	1	0	0	0	0	0
	2	0	0	1	2	1
	3	0	0.5	2	3	2

Posługując się powyższymi macierzami przejść i wyjść, można rozwiązać proste zadanie dotyczące działania modelowanego zespołu. Stan początkowy: zespół łąkowy ($S_0 = 1$). W ciągu pierwszych pięciu lat nie było wyrębu ($X_0 = 1$, $X_1 = 1$, $X_2 = 1$, $X_3 =1$, $X_4 =1$), następnie w ciągu dwóch kolejnych lat był wyrąb selektywny ($X_5 = 2$, $X_6 = 2$), a później nastąpił wyrąb całkowity ($X_7 = 3$). W ciągu ostatnich dwóch lat nie było wyrębu ($X_8 = 1$, $X_9 = 1$). Należy określić końcowy stan zespołu i całkowitą ilość wywiezionego drewna. Tutaj należy powiedzieć, że kolejny krok czasowy może być na tyle długi, że pozwala na przejście do kolejnego stanu sukcesji.

Aby rozwiązać takie zadanie, nie potrzeba komputera, a nawet kieszonkowego kalkulatora. Posługując się kolejno tablicami 4.5 i 4.6, otrzymujemy szereg stanów i wyjść układu, przedstawionych w tab. 4.7. Na przykład podczas kroku czasowego 5 układ znajduje się w stanie 4, a wejściowe oddziaływanie wynosi wtedy 2. W tab. 4.5 znajdujemy, że kolejny stan będzie 5, a w tablicy 4.6 odnajdujemy, że wyjście będzie wynosić 2.

Tab. 4.7. Kolejne stany i wyjścia automatu przy danych wejściach i stanie początkowym.

CZAS	0	1	2	3	4	5	6	7	8	9	Σ
Wejście (X)	1	1	1	1	1	2	2	3	1	1	
Stan (S)	1	2	3	4	5	4	5	5	1	2	
Wyjście (Y)	0	0	0	0	0	2	1	2	0	0	5

Po takim wstępnym zapoznaniu się, możemy sformułować definicję końcowego automatu komórkowego (Melichow 1971).

Automat A zawiera pięć obiektów: skończony zbiór $X = \{x_i\}$, $i \in I = \{1,2,3,...m\}$, nazywany wejściowym alfabetem automatu; skończony zbiór Y

= $\{y_j\}$, $j \in J = \{1,2,3,...n\}$, nazywany wyjściowym alfabetem automatu; skończony zbiór $Q = \{q_k\}$, $k \in K = \{1,2,3,...p\}$, nazywany alfabetem stanów; element $q_0 \in Q$ jako początkowy stan automatu; odwzorowania F zbioru Q samego na siebie. Odwzorowanie F każdemu $q \in Q$ i każdej wejściowej literze $x \in X$ przyporządkowuje stan $q \in Q$, co wyznacza funkcję przejść $\varphi = (q, x)$, i literę wyjściową $y \in Y$, wyznaczając funkcję wyjść $\psi = (q,x)$.

Specjalnie przytaczamy definicję automatu końcowego w takiej postaci, w jakiej jest ona podawana w podręcznikach matematyki, aby zapobiec ewentualnym szokom, jakie mogą się przytrafić biologom lub geografom po napotkaniu czegoś takiego. Za symbolami z teorii mnogości kryją się proste i zrozumiałe sprawy, wprowadzone w przykładzie sukcesji zbiorowiska leśnego. Niezwykłe użycie terminu „alfabet" nie znaczy nic poza przeliczeniem symboli (w przykładzie zastosowaliśmy liczby arabskie, ale równie skutecznie można by wykorzystać inne symbole). Przykład funkcji $\varphi = (q, x)$ przedstawiony jest w tab. 4.5, a funkcji $\psi = (q,x)$ w tab. 4.6.

Modele z zastosowaniem automatów komórkowych są podstawowym narzędziem prognozowania, zwłaszcza modelowania takich krajobrazów, które mają wysoki stopień mozaikowości. Urok automatów komórkowych kryje się, po pierwsze, w możliwościach łatwego obejrzenia wyników w postaci macierzy przejść, a po drugie – w możliwościach odzwierciedlenia struktur przestrzennych. Automaty komórkowe dają również możliwość zastosowania łatwiejszej i bardziej zrozumiałej struktury programów w porównaniu do równań różniczkowych.

Pojęcie automatów komórkowych powstało poza biologią czy ekologią, chociaż o możliwościach zastosowania w biologii takich matematycznych narzędzi pisał jeden z twórców cybernetyki – J. von Neuman (1966).

Przed analizowaniem konkretnych przykładów modelowania krajobrazu z zastosowaniem automatów komórkowych należy zdefiniować pojęcie automatu komórkowego.

Matematyczny aparat modelu zależy od rodzaju czasu i stanu oryginału w modelu (Cerazan 1998). Jeżeli czas i stan w modelu traktujemy jako wartości nieciągłe, wtedy matematyczny aparat, opisujący ten model, możemy traktować jako „automat".

Automat komórkowy stosowany w modelowaniu krajobrazu jest połączeniem pięciu obiektów:

1. Regularnej siatki złożonej z komórek;
2. Skończonej liczby komórek;
3. Reguł wyznaczania sąsiadów;
4. Funkcji przejść, która wyznacza stan komórki w następnym momencie czasu, w zależności od stanu komórki i jej sąsiadów w danym momencie;

5. Początkowej konfiguracji stanów komórek. W zasadzie automat komórkowy (A) można przedstawić jako następujący zbiór obiektów:

$$A = \langle L, S, \delta, f \rangle$$

gdzie:

L – mapa, która wyznacza wzajemne położenie komórek
S – stan automatu
δ – sposób wyznaczania sąsiadów
f – funkcja przejść

Mapa jest zapisywana przeważnie we współrzędnych prostokątnych. Nie ma żadnych ograniczeń struktury przestrzeni modelowanej poza skończoną liczbą komórek. Ilość stanów automatu też jest liczbą skończoną.

Sposób wyznaczania komórek sąsiednich może być różny. Na przykład przy podziale powierzchni na kwadraty w prostokątnym układzie współrzędnych prostokątnych sąsiednimi mogą być te kwadraty, które mają wspólne boki. W tym wypadku każda komórka ma najwyżej czterech sąsiadów. Jest to sposób wyznaczania sąsiadów według J. von Neumana (rys. 4.6). W innym wypadku można analizować osiem sąsiednich komórek. Jest to sposób wyznaczania sąsiadów według Moore.

Funkcja przejść wyznacza stan komórki w czasie (t + 1) w zależności od stanu tejże komórki i jej sąsiadów w momencie (t) (rys. 4.7) oraz w zależności od wpływów wewnętrznych w tym samym momencie. Sama funkcja przejść może być deterministyczna bądź stochastyczna. W ostatnim przypadku mamy do czynienia z tak zwanym przestrzenno-czasowym łańcuchem Markowa (Guttorp 1995). Problemy konstruowania funkcji przejść w modelach krajobrazowych zostały dokładnie przeanalizowane w publikacji Childress et al. 1996.

4.2. Modelowanie krajobrazów za pomocą aparatu automatów komórkowych na przykładzie Bieszczad

Lądowe układy ekologiczne charakteryzują się szczególnie wysokim stopniem mozaikowatości. Dlatego w konstruowaniu modeli tych środowisk należy poszukiwać adekwatnych sposobów (Logofet 1999). Jednym z takich sposobów może być zastosowanie aparatu modeli komórkowych.

Jak już wsopmniano, pojęcie automatu komórkowego nie miało początkowo żadnego związku z biologią, chociaż J. von Neuman (1966), jeden z twórców cybernetyki, wskazywał na możliwości zastosowania takich automatów w biologii. Znaczącą rolę w popularyzacji teorii automatów komórkowych w dziedzinie biologii odegrała gra komputerowa *Life* (Gardner 1971). Jednakże konkretne zastosowania ekologiczne pojawiły się dopiero w latach dziewięćdziesiatych ubiegłego wieku (Hassel et al. 1991; Colasanti, Grime 1993; Balzter et al. 1998). Metoda automatów komórkowych znalazła udane zastosowanie w dobrze rozwiniętej dziedzinie modelowania poświęconej następstwom leśnych pożarów (Karafyllidis, Thanailakis 1997).

Zachętą do stosowania automatów w modelowaniu ekologicznym jest przede wszystkim przejrzystość wyników, prezentowanych w postaci grafów przejść, zwłaszcza w sytuacji względnie niewielkiej liczby stanów dyskretnych. Automaty komórkowe mają poza tym możliwość odwzorowania struktur przestrzennych, zazwyczaj płaskich, co jest szczególnie ważne przy zagadnieniach ekologii krajobrazowej. Należy dodać, że programy modeli automatowych są lżejsze i prostsze niż układy równań różniczkowych. Jednakże „coś za coś": modele automatowe „płacą” swoją dyskretnością stanów; nawiasem mówiąc, fitocenolodzy już się do przyzwyczaili przez stosowanie skali Braun-Blanqueta.

Zastosowanie automatów komórkowych w modelowaniu krajobrazów zilustrujemy na przykładzie wybranych wyników badań (program *Celauto* – zob. CD dołączony do książki) prowadzonych w rejonie Bieszczad, graniczącym z Ukrainą na wschodzie i Słowacją na południu. Rejon ten jest bardzo zróżnicowany krajobrazowo – od alpejskich łąk powyżej 1000 m n.p.m. do łęgowych lasów w dolinie rzeki San. Próby posługiwania się scalonymi charakterystykami całego regionu okazały się zbyt schematyczne (Menshutkin, Klekowski, w druku). Konieczne jest uszczegółowienie, zwłaszcza uwzględnienie przestrzennych niejednorodności rozmieszczenia ekologicznych i ekonomicznych właściwości tego rejonu.

Podstawą modelu jest mapa topograficzna byłego województwa krośnieńskiego w skali 1:100 000 (1998 r.); niekiedy używano, jako uzupełnienia, map 1:25 000. Teren Bieszczad został podzielony na kwadraty 2×2 km. Model zawiera łącznie 743 takie kwadraty. Stan każdej komórki określony zostaje na podstawie tego, jaki procent powierzchni komórki jest zajęty przez las, łąkę, pola uprawne, osiedla wiejskie lub miejskie. Oceny powierzchni dokonywano z dokładnością do 1 km^2. Ustalono 28 możliwych stanów komórki, które kodowano w następujący, wygodny dla modelowania, sposób:

FFFF – całą powierzchnię komórki zajmuje las;
FFFA – trzy czwarte powierzchni komórki zajmuje las, a resztę łąki lub pola uprawne;

FFFV – trzy czwarte powierzchni zajęte przez las, a reszta przez osadę typu wiejskiego;
FFFT – trzy czwarte powierzchni zajmuje las, resztę osada typu miejskiego;
AAAA – całą powierzchnię komórki zajmuje łąka lub uprawy polowe;
VVVV – cała powierzchnia komórki jest pokryta przez osadę typu wiejskiego;
TTTT – całą powierzchnię komórki pokrywa osada typu miejskiego;
FAVT – komórkę zajmuje las, łąka lub uprawy polowe, osady typu miejskiego i wiejskiego (w równych proporcjach).
I tak dalej.

Zasada czterosylabowego kodu polega na tym, że symbol F powtarza się w nim tyle razy, ile ćwiartek komórki zajęte jest przez las, symbol A – ile ćwiartek komórki zajęte jest przez łąki i ziemię orną, symbol V – ile ćwiartek komórki zajętych jest przez osady typu wiejskiego, a symbol T – ile ćwiartek komórki zajętych jest przez osiedla miejskie. Nie wszystkie możliwe kombinacje wydają się realistyczne. Z tego względu na przykład FFFT (miasto otoczone lasem) i podobne nie były uwzględnione.

Po wprowadzeniu mapy (L) i możliwych stanów automatu (S) należy wprowadzić sposób określania sąsiadów komórki (δ). W tym przypadku przyjmiemy, że sąsiadami są te komórki, które mają wspólne boki, to znaczy przyjmiemy układ z czterema sąsiadami.

Pozostaje jeszcze zdefiniować funkcję przejść automatu (f). W tym celu należy uwzględnić historię tego regionu. Na rys. 4.8 przedstawiono stan tego regionu w 1936 r. na podstawie danych udostępnionych nam (bardzo dziękujemy!) przez pana magistra M. Augustyna (Oddział Karpacki Międzynarodowego Centrum Ekologii PAN w Ustrzykach Dolnych). W wyniku gwałtownych wydarzeń politycznych i gospodarczych oraz przesunięć granic państwowych i administracyjnych w latach 1939-1953 nastąpiło znaczne zmniejszenie liczebności zaludnienia i prawie całkowita zmiana składu demograficznego. Liczne komórki, w których przeważał komponent wiejskiego osiedla (na przykład FAVV, AAVV), przesunęły się w kierunku przewagi lasu (na przykład FFAV, FFFA, FFFV). Pozwoliło to określić empirycznie prawdopodobieństwo zalesienia komórki w ciągu jednostki czasu, uzależniwszy ten proces od wysokości położenia komórki nad poziomem morza; chodzi o to, że wierzchołki gór, tak zwane połoniny, nie mogą porastać lasem z powodów geobotanicznych. Innym uwarunkowaniem jest bliskość rzeki: w dolinach rzek zalesianie się przebiega szybciej niż na działach wodnych. Natomiast pobliże dużych osiedli spowalnia zalesiania lub całkiem je zatrzymuje. Ważnym zdarzeniem dla regionu było utworzenie Bieszczadzkiego Parku Narodowego, co zatrzymało wzrost liczby osad na jego terytorium.

Rys. 4.9 przedstawia graf przejść tworzonego automatu. Na przykład jeżeli w zwartym lesie (stan FFFF) pojawi się polana osiągająca powierzchnię 1 km^2, odpowiada to łukowi grafu od stanu FFFF do stanu FFFA. Każdy wierzchołek zorientowanego grafu (graf zorientowany to taki graf, w którym łuk między dwoma wierzchołkami ma oznaczony kierunek) przedstawionego na rys. 4.9 ma pętlę, co oznacza, że dowolny stan układu może trwać w ciągu kroku czasowego. Łuki przejść grafu są symetryczne, co oznacza, że procesy zachodzące w układzie są odwracalne. Na przykład jeżeli w zwartym lesie pojawi się polana (FFFF → FFFA), to jest możliwe zarośnięcie tej polany (FFFA → FFFF). Cała istota funkcji przejść polega na tym, że każdemu łukowi tego grafu należy przypisać pewne prawdopodobieństwo, które zależy od stanu sąsiednich komórek, kierunku przejścia, stałych właściwości lokalizacji komórki (na przykład wysokość n.p.m. lub ekspozycja stoku) i scenariusza rozwoju regionu.

Funkcji przejść automatu, jak również graficznej prezentacji wyników nadano postać programu napisanego w języku *Visual Basic*. Zamiast czarno-białych rysunków zastosowano ilustracje kolorowe, co pozwoliło jednocześnie zaprezentować wszystkie składniki układu, włącznie z siecią hydrologiczną i transportową, a także rzeźbą terenu. Ponieważ funkcja przejść automatu ma postać stochastyczną, każdy wariant działania modelu – z takim samym stanem początkowym i niezmiennymi oddziaływaniami zewnętrznymi – był powtarzany wielokrotnie (zwykle 30 razy, ale można obrać inną liczbę), a rezultaty poddawano obróbce statystycznej (była to więc odmiana metody Monte Carlo).

Rozpatrzono różne scenariusze rozwoju regionu Bieszczad:

S c e n a r i u s z p i e r w s z y przewidywał zachowanie takich funkcji przejść i oddziaływań zewnętrznych, jakie miały miejsce w latach dziewięćdziesiątych. Oznacza to, że na obszarze Parku Narodowego nie wprowadza się żadnych antropogennych oddziaływań, a granice Parku nie zmieniają się. Poza terenem Parku trwa proces zalesiania ugorów, ale jest on mniej intensywny niż na terytorium Parku.

S c e n a r i u s z d r u g i zakłada intensywny rozwój regionu, oparty na prywatyzacji gruntów na obszarach poza Parkiem Narodowym oraz wzroście inwestycji w działach produkcji i usług. W algorytmie funkcji przejść automatu komórkowego znajduje to wyraz w zwiększeniu prawdopodobieństwa przejść ze stanów zawierających symbol A (łąka, pole orne) do stanów zawierających symbol V (osiedle), z jednoczesnym zmniejszeniem prawdopodobieństwa przejść odwrotnych. Podlega intensyfikacji proces przechodzenia osiedli typu wiejskiego do osiedli miejskich, jak również proces zmniejszania się powierzchni lasów w pobliżu wzrastających osiedli.

S c e n a r i u s z t r z e c i zawiera próbę uwzględnienia wpływu globalnych zmian klimatu. Wobec rejonu Bieszczad znajduje to wyraz w podwyższeniu temperatury i zmniejszeniu opadów. W algorytmie funkcji przejść automatu komórkowego odpowiada to zwiększeniu prawdopodobieństwa przejścia stanów zawierających symbol F (las) do stanów zawierających symbol A (łąka i pole orne), pod warunkiem że przez daną komórkę lub przez komórkę sąsiednią przepływa rzeka. Sens ekologiczny takich zmian funkcji przejść polega na ginięciu lasów olchowych rosnących w dolinach rzecznych.

Tab. 4.8. Procentowy podział powierzchni regionu Bieszczad w różnych momentach czasowych (suma we wszystkich kolumnach, oprócz pierwszej, jest mniejsza od 100% z powodu zbudowania w 1964 r. zbiornika Solina, którego powierzchnia wynosi około 1% terenu).

	Stan w 1936 r.	Stan w 1998 r.	Prognoza dla 2100 r.		
			Scenariusz pierwszy	Scenariusz drugi	Scenariusz trzeci
Powierzchnia zajęta przez lasy	42.6	63.7	73.9	65.5	66.9
Powierzchnia zajęta przez łąki, nieużytki (?) i ziemie orne	47.7	29.5	14.5	11.9	21.9
Powierzchnia zajęta przez osady typu wiejskiego	9.3	5.5	9.5	20.4	9.5
Powierzchnia zajęta przez osady typu miejskiego	0.4	0.2	1.1	1.1	1.1

Tab. 4.8 zawiera wyniki działania modelu wychodzące z tego samego stanu początkowego według danych z 1998 ro. i rozgrywane według trzech opisanych wyżej scenariuszy. Rozwój zdarzeń według scenariusza pierwszego doprowadza do istotnego wzrostu powierzchni lasów, a następnie zmniejszenia powierzchni nieużytków i ziemi ornej. Jednocześnie następuje wzrost miast i osiedli typu wiejskiego.

Prywatyzacja (scenariusz drugi) prowadzi, przy przyjętych zmianach w funkcji przejść, do wyraźnego wzrostu powierzchni zajętej przez osady typu wiejskiego. Ogólna powierzchnia lasów pozostaje prawie taka sama jak w 1998 r., gdyż znaczne zmniejszenie powierzchni lasów w północnej części Bieszczad jest skompensowane wzrostem powierzchni lasów na terenie Parku Narodowego.

Imitacja globalnych zmian klimatu (scenariusz trzeci) prowadzi do spowolnienia wzrostu powierzchni lasów – w porównaniu do rozwoju sytuacji w scenariuszu pierwszym. Redukcja powierzchni lasów w dolinach rzecznych jest przy tym w znacznym stopniu skompensowana przez wzrost powierzchni lasów bukowych, zajmujących stoki gór do wysokości 1000 m n.p.m.

Zasadniczym niedostatkiem zaproponowanego modelu jest brak zróżnicowania pojęcia lasu, albowiem w Bieszczadach są lasy bukowe, jodłowe, świerkowe, modrzewiowe, olchowe, a nawet klonowe, kształtujące rozmaite fitocenozy. Badania geobotaniczne były prowadzone w Bieszczadach bardzo intensywnie, doprowadzając do modeli zjawisk sukcesyjnych (Wilkie, Finn 1988; Green 1989; Satoh 1990).

Próby włączenia typu zespołu roślinnego do zakresu pojęcia stanu automatu doprowadziły do wyraźnego skomplikowania funkcji przejść, wobec czego autorzy odłożyli ten zabieg do dalszego etapu rozbudowy modelu.

Inny kłopot polega na tym, że zastosowany w modelu podział na kwadraty 2×2 km nie jest optymalny. Heksagonalna siatka współrzędnych, z rozbiciem płaszczyzny na regularne sześciokąty, jest wprawdzie bliższa rzeczywistości, lecz sprawia dodatkowe trudności przy obróbce kartograficznej. Najbardziej pociągającym sposobem byłoby wykorzystanie naturalnych granic między biogeocenozami lub obiektami antropogennymi. Jest to jednak sposób nadzwyczaj pracochłonny z powodu zmian powierzchni poszczególnych komórek. Wymaga opracowania specjalnego oprogramowania współpracującego z systemem informacji geograficznej (GIS). Dodatkową komplikacją jest to, że granice naturalnych biocenoz z reguły nie są zgodne z granicami leśnictw, gmin i innych jednostek administracyjnych, dla których gromadzone są informacje statystyczne (Andrzejewski 2001).

Granice rejonu Bieszczad wyznaczają granice państwowe według sytuacji z 1951 r.; jedynie granice północna i północno-zachodnia określone zostały w znacznym stopniu dowolnie. Jednakże przy opracowywaniu historycznych materiałów kartograficznych należy uwzględniać i to, że zarówno granice, jak i strategia wykorzystywania zasobów przyrody ulegają istotnym zmianom nawet w stosunkowo krótkim okresie stulecia.

W modelu niewystarczająco rozwinięta jest jego część hydrologiczna. Zbiorniki zaporowe Solina i Myczkowce zajmują wprawdzie niewielką część powierzchni regionu, tym niemniej wpływają w sposób istotny nie tylko na znajdujące się w pobliżu naturalne i sztuczne obiekty, lecz także na sytuację hydrologiczną w całym regionie, szczególnie w ekstremalnych sytuacjach powodzi, które nie są rzadkim zjawiskiem w tej części Europy.

Opracowywany w Międzynarodowym Centrum Ekologii PAN imitacyjny model zbiornika zaporowego Solina (Klekowski, Menshutkin, w druku) może być traktowany jako submodel w modelu regionu Bieszczad. Mimo bardzo wielkich metodycznych różnic między modelowaniem wodnych i lądowych układów ekologicznych zestawianie takich modeli jest w pełni możliwe i może okazać się owocne.

Przebieg procesów atmosferycznych nad regionem, transfery strefowe i południkowe, zmiany koncentracji dwutlenku węgla i transport różnego rodzaju zanieczyszczeń nie są bez znaczenia dla dynamiki regionu. Skomplikowana orografia Bieszczad stwarza warunki do zróżnicowania mikroklimatycznego, lokalnych wiatrów i innych zjawisk, które mogą w istotny sposób wpływać na przebieg procesów w lokalnych ekosystemach i krajobrazach. Dlatego celowe wydaje się rozważenie związków między modelem regionu i odpowiednim modelem lokalnych zróżnicowań klimatycznych i przemieszczeń atmosferycznych.

Być może należałoby nieco zmodyfikować klasyczną definicję automatu komórkowego i związanego z nią pojęcia przestrzenno-czasowych łańcuchów Markowa, jeżeli chcemy stosować te pojęcia do układów biogeocenoz krajobrazowych i lądowych. Na przykład po uwzględnieniu sieci hydrologicznej pojęcie sąsiada nie będzie już tak oczywiste jak w klasycznym przypadku. Mapa (L), którą przy klasycznym podejściu przyjmuje się jako niezmienną w czasie, w realnej sytuacji może stać się zależna od stanu komórek. Nie trzeba w tym celu zmieniać granic państwowych; wystarczą globalne zmiany klimatu lub inwestycyjnej polityki państwa.

5. Potencjalne kierunki dalszego rozwoju modelowania krajobrazu

Rozpatrzmy wybrane możliwe drogi rozwoju modeli krajobrazowych i modeli ekosystemów lądowych. Przykład modelu bilansowego ekosystemu lądowego (rozdział 2) pokazuje, jak trudno dla takiego modelu dostarczyć wiarygodne wartości wszystkich parametrów. Dotyczy to zwłaszcza procesów zachodzących w glebie przy współdziałaniu systemów korzeniowych roślin z grzybami i bakteriami. Wielki stopień heterogenności układów glebowych nie dopuszcza możliwości posługiwania się wartościami średnimi, jak to się dzieje przy modelowaniu ekosystemów wodnych.

Jednym z możliwych wyjść z takiej sytuacji jest zastosowanie m o d e l i l o g i c z n o - l i n g w i n i s t y c z n y c h. Pojęcie zmiennej lingwinistycznej wprowadził do nauki L. Zadeh (1975), który zastosowanie zmiennych lingwinistycznych i rozmytej logiki określił jako wyliczanie za pomocą słów (Zadeh 1999). Pomysł, że liczby zamienić można sformułowaniami słownymi, jest bliski wyobrażeniom fitocenologów, o czym przypomina skala Brauna-Blanqueta (Braun-Blanquet 1951), służąca do oceny obfitości gatunków roślin w zespołach. W monografiach o krajobrazach zwykle poświęca się dużo uwagi skalowaniu i porównaniu skal rozmaitego typu (Armand 1975). Jednakże L. Zadeh poszedł znacznie dalej i opracował aparat matematyczny służący do konstruowania modeli, które zawierają sformułowania słowne i współzależności między nimi, lecz brak w nich wielkości liczbowych.

Główna idea modeli logiczno-lingwinistycznych polega na tym, żeby nie usiłować ściśle ilościowo zobrazować wszystkich procesów, jakie zachodzą w oryginale, a wprowadzić do modelu tylko to, co o konstrukcji i zachowaniu się modelu wiedzą eksperci, czyli ludzie dobrze znający oryginał. A ludzie, jak wiadomo, myślą za pomocą obrazów i słów, a nie za pomocą liczb. Tyczy się to nawet zawodowych matematyków, nie mówiąc już o ludziach innych zawodów.

Spróbujemy na prostym konkretnym przykładzie pokazać, jak można budować logiczno-lingwinistyczne modele ekosystemów lądowych w sytuacji ostrego niedoboru danych ilościowych (Menshutkin, Fisher 1995). Rys. 5.1 przedstawia schemat blokowy modelu ekosystemu w wyższej części Karkonoszy. Każda zmienna w tym modelu (nazwy zmiennych znajdują się w objaśnieniu rysunku) ma od 4 do 16 wielkości, które są opisana za pomocą sformułowań słownych. Na przykład ekspozycja stoku

(E) może mieć znaczenia: „północny", wschodni", „południowy" i „zachodni"; odczyn gleby (PHSOIL): „neutralny pH = 7.7-7.4", „lekko kwaśny pH = 7.4-6.4", „kwaśny pH = 6.7-5.4", „bardzo kwaśny pH = poniżej 5.4"; intensywność mineralizacji w glebie materii organicznej (MINERALIZATION): „bardzo niska", „niska", „średnia", „wysoka", „bardzo wysoka" itd.

Model składa się z dwóch części: bazy wiedzy i maszyny wnioskującej i jest w istocie odmianą systemu ekspertalnego (Forsyth 1984; Mulawka 1996). Baza wiedzy tego modelu składała się z opinii, których przykłady są przytoczone poniżej.

„Jeżeli zawartość wody w glebie (WSOL) jest bardzo wysoka, to wtedy w warunkach dowolnych koncentracji materii organicznej w glebie (ORGANIC) i dowolnych temperaturach gleby (TSOL) intensywność mineralizacji materii organicznej w glebie (MINERALIZATION) będzie bardzo niska".

„Jeżeli opady atmosferyczne (P) są wyższe od średniej wieloletniej i intensywność radiacji słonecznej niższa lub równa średniej wieloletniej, to wtedy zawartość wody w glebie (WSOL) będzie wysoka".

„Jeżeli roślinność jest typu (TR) *Piceetum hercynicum typicum*, a śmiertelność roślin (MORT) jest wysoka, to wtedy masa ginącej części roślinności (FALL) jest wysoka".

„Jeżeli koncentracja glinu w glebie (Al) jest wysoka, to wtedy biomasa bakterii w glebie (BACTERIA) jest niska".

„Jeżeli zawartość ciężkich metali w opadach atmosferycznych (HM) jest bardzo wysoka, to wówczas przy dowolnych wielkościach zawartości wapnia w opadach (Ca), magnezu w opadach (Mg) i koncentracji glinu w glebie (Al), śmiertelność roślin (MORT) jest wysoka."

Liczba takich stwierdzeń w bazie wiedzy omawianego modelu przekraczała 200.

Funkcja maszyny wnioskującej polegała na wyszukiwaniu potrzebnych stwierdzeń w bazie wiedzy i rozwiązywaniu ewentualnych konfliktów między nimi. Za pomocą tego modelu sporządzono prognozę rozwoju roślinności wyższej partii Karkonoszy, w warunkach różnych wariantów oddziaływań antropgennych, w szczególności nasilenia kwaśnych deszczy.

Inny przykład nietradycyjnego podejścia do modelowania układów krajobrazowych prezentuje praca V. V. Menshutkina i R. Z. Klekowskiego (2001), którzy badali układ ekonomiczno-ekologiczny Bieszczad. Rys. 5.2 prezentuje schemat blokowy tego modelu. W odróżnieniu od modelu poprzedniego, w którym wszystkie wartości zmiennych były dyskretne, w tym modelu zmienne mają wartości ciągłe, określane w zakresie od 0 do 1. W modelu zastosowano aparat matematyczny logiki rozmytej (Klir, Folger 1988), która jest uogólnieniem wieloznacznej logiki Łukasiewicza (1961).

Na przykład intensywność spływu rzecznego (FLOW) określana jest jako zależność od ilości opadów (PREC) i powierzchni lasów w regionie (FOR) ze wzoru:

$$FLOW = PREC \vee \neg FOR$$

gdzie:

$\neg$ – symbol logicznego działania przeczenia,
$\vee$ – symbol logicznego działania dysjunkcji.

W tab. 5.1 przytoczone są wzory obliczania tych działań. Na przykład przy opadach nieco wyższych w danym roku od średniej wieloletniej (PREC = 0.6) i stosunkowo wysokiej lesistości regionu (FOR = 0.7), intensywność spływu rzecznego – w wariancie stochastycznym działań logiki rozmytej – zostanie oceniona jako:

$$FLOW = 0,6 + (1 - 0,7) - 0,6 * (1 - 0,7) = 0,72$$

co określa spływ rzeczny jako wyraźnie większy od średniej.

Tab. 5.1. Wykonanie operacji logiki rozmytej.

	WARIANT PROGOWY	WARIANT STOCHASTYCZNY
Przeczenie ($z = \neg x$)	$z = 1 - x$	$z = 1 - x$
Dysjunkcja ($z = x \vee y$)	$z = \max(x,y)$	$z = x + y - x y$
Koniunkcja ($z = x \wedge y$)	$z = \min(x,y)$	$z = x y$

Spośród innych przykładów zastosowania metod sztucznej inteligencji do problemów bliskich modelowaniu krajobrazowemu wymienimy prace: Starfeld, Bleloch 1983; Sterbacek et al. 1990; Lembach 1994; Tuma et al. 1996.

6. Modelowanie krajobrazowe jako podstawa do podejmowania decyzji społeczno-administracyjnych

Od kilkudziesięciu już lat w społecznej opinii krajów rozwiniętych ugruntowuje się przekonanie o konieczności coraz szerszego uwzględniania racji ekologicznych przy projektowaniu i realizacji działań gospodarczych powodujących zmiany w stanie środowiska naturalnego, a zwłaszcza wywołujących zagrożenia i uszkodzenia tego środowiska.

Zgodnie z wszechobejmującym prawem Parkinsona często, z pewnością zbyt często, działania ograniczają się do powołania urzędów i komisji dość udanie pozorujących odpowiednie działania, zwłaszcza poprzez ładne nazwy tych urzędów i komisji. Są już jednak przykłady bardzo zachęcające. Jeden ze współautorów tej książki uczestniczy w wielkim programie naukowo-aplikacyjnym, mającym na celu zapewnienie optymalnej gospodarki zasobami wodnymi i zaopatrzeniem w wodę metropolii Sankt Petersburga. Podstawowym narzędziem badawczym i oparciem dla procesu podejmowania decyzji społeczno-administracyjnych jest modelowanie komputerowe wszystkich procesów przyrodniczych i gospodarczych toczących się na przestrzeni obejmującej źródła wody: ogromne jezioro Ładoga, rzeka Newa wraz z systemem zaopatrzenia w wodę i zrzutem ścieków czteromilionowej metropolii oraz wschodni akwen Zatoki Fińskiej, do której trafiają wszystkie ścieki z tej metropolii. Na ten temat ukazały się dwa obszerne opracowania książkowe (Menshutkin 1997; Alimow, Umnov 1998). W dodatku z uwagi na wielki obszar i delimitację granic jako zlewni stosunkowo krótkiej, ale bardzo obfitej w wodę rzeki są to niewątpliwie badania i modelowania krajobrazowe.

Główną zaletą modelowania w tworzeniu i wyborze scenariusza działań gospodarczych jest to, że modele mogą prognozować skutki zamierzonych działań na wiele lat, a nawet stuleci naprzód. Model FORKOME, omówiony w rozdziale 3, umożliwia symulację składu gatunkowego drzew, ich liczby i biomasy, a także skutków różnych scenariuszy wyrębu. Prognozy skutków ocieplenia klimatu mogą sięgać raczej okresu kilku tysięcy lat!

Drugą korzyścią z modelowania, nie mniej ważną niż uzyskiwanie błyskawicznych odpowiedzi, jest brak ujemnych, a nawet katastrofalnych skutków rzeczywistego eksperymentowania na środowisku. W komputerze możemy wyciąć wszystkie lasy w Bieszczadach czy gdzie indziej, zobaczyć, co z tego wyniknie, a w rzeczywistości ani jedna gałązka nie zostanie złamana wskutek naszego szaleństwa wyrębowego. Wielka szkoda, że na ekranie komputera nie można było jeszcze zobaczyć, jakie będą skutki zbudowania, za bardzo wielkie pieniądze, tamy na wschodnim krańcu Zatoki Fińskiej, która miała ochronić wspominany Sankt Petersburg od powodzi wywoływanych przez sztormowe wiatry zachodnie. Tamę prawdzie zbudowano, ale nie dokończono – i nie będzie dokończona, bo nie przyniosła spodziewanych dobrodziejstw.

Powróćmy jednak do modelu FORKOME. Nie ruszając zatem nawet jednej gałązki, możemy określić scenariusze gospodarki leśnej, która ma uzyskać albo dużo masy drzewnej (produkcja celulozy, papieru), albo drzewa budulcowego, albo pełnić rolę rekreacyjną itd.

Możliwości, jakie daje modelowanie, mogą zmienić całą „filozofię prowadzenia badań" i ich logistykę. Dotychczas badania terenowe zaczynaliśmy zwykle od zbierania materiałów i informacji na wybranej, ograniczonej liczbie stanowisk – co trwało nie krócej niż rok – z zamiarem zaplanowania na tej podstawie „badań właściwych", kilku- lub wieloletnich, na większej liczbie stanowisk. Często polegało to na mierzeniu i zbieraniu wszystkiego, co się dało, najczęściej zgodnie ze specjalnościami stojących do dyspozycji wykonawców. Często w projektach (na przykład w grantach) dodawano modne stwierdzenie, że zebrane materiały pozwolą skonstruować model komputerowy (ładniej: matematyczny) badanego obiektu czy procesu. Znane nam przykłady niczego takiego nie umożliwiły!

Sądzimy, że modelowanie może być użyteczne, jeżeli zaraz na początku, zanim pobierzemy pierwszą planową próbę, określimy minimalny zestaw informacji, jakie będą niezbędne dla wymarzonego modelu, i odpowiemy sobie na brutalne pytanie: czy jesteśmy w stanie taki zestaw uzyskać? To nie zawsze muszą być nowe badania! W postaci publikowanej i nie opublikowanej istnieje już wiele potrzebnych nam danych. Wiele jak najbardziej wiarygodnych i wystarczająco dokładnych informacji znajduje się w „bazach wiedzy" zgromadzonych w głowach naszych kolegów-ekspertów. Właśnie na podstawie tej *ad hoc* zebranej bazy danych należy sporządzić próbny model i dopiero wtedy, jeżeli ocenimy go jako sensowny, rozpocząć gromadzenie nowych danych, które natychmiast spowodują wzrost kosztów!

To podjęcie roli *advocatus diaboli* nie miało na celu zniechęcania Czytelników do wykonywania badań własnymi rękami. Warto jednak pamiętać, że właśnie krajobraz jest szczególnie skomplikowanym, całościowym systemem o strukturze hierarchicznej, która na dodatek ciągle ulega zmianom spowodowanym przez działalność człowieka (Zięba 1998, 2002).

CZĘŚĆ DRUGA

Wprowadzenie

Pierwsza część książki zawierała jedynie opisy tego, jak działają modele zespołów leśnych, ekosystemów lądowych i całych kompleksów krajobrazowych. Można było sprawdzić, czy dołączone do książki programy rzeczywiście działają, a także przeprowadzić za ich pomocą własne eksperymenty. Na przykład w modelu ekosystemu lądowego ustawić intensywność promieniowania słońca na zerze i przekonać się, że wymrze wszystko, co żywe.

W drugiej części postaramy się nie tyle opowiedzieć, ile p o k a z a ć, jak zbudować takie modele, o jakich była mowa w części pierwszej. Na początek warto jednak powiedzieć co nieco o samych komputerach i językach programowania.

Ludziom współczesnym, zwłaszcza studentom, nie trzeba już wyjaśniać, co to takiego ten komputer, gdzie i jak naciskać klawisze na klawiaturze, jak posługiwać się myszą, ani też, co trzeba i czego nie należy robić, aby nie zakazić swojego komputera wirusami. Komputer znakomicie redaguje dokumenty, poprawia błędy ortograficzne i gramatyczne, retuszuje grafikę i robi jeszcze wiele innych pożytecznych rzeczy. Szybko przywykliśmy do tych wygód komputeryzacji i przestaliśmy się nad tym zastanawiać.

W stworzenie tych wszystkich systemów operacyjnych, gier komputerowych, programów do obróbki grafiki komputerowej, systemów wyszukiwania danych we wszelkich bazach danych włożona została olbrzymia praca programistów.

Dziedzina modelowania, a zwłaszcza modelowania krajobrazowego nie rokuje szybkiego konstruowania programów i modeli uniwersalnych. Wprawdzie nowe języki modelowania pojawiają się jeden za drugim (na przykład *Simula* czy *Stella*), lecz powiedzmy sobie szczerze, że najnowsze narzędzia automatyzacji modelowania to w najlepszym razie wczorajszy dzień nauki, a rzeczywiście nowe podejścia można tworzyć licząc wyłącznie na własne siły i umiejętności.

Dlatego właśnie podejmujemy ryzykowną próbę pokazania „kuchni” modelowania – tego właśnie, czego praktycznie nigdy nie pokazują publikacje, ograniczając się co najwyżej do odwołania się do precedensowych zastosowań.

Współczesne języki programowania, do których należy *Visual Basic*, działają na z a s a d z i e p r o g r a m o w a n i a o b i e k t o w e g o. Znajomość tej zasady pozwala traktować reguły programowania nie tylko jako zbiór pożytecznych wskazówek w rodzaju książki kucharskiej, lecz jako zwarty i logiczny system, jak również pozwala na utworzenie własnego arsenału sposobów i chwytów ułatwiających tworzenie nowych programów i modeli.

Obiektem (ang. *Object*) może być konstrukcja zbudowana z danych i komend bardzo różnych typów i o różnym przeznaczeniu. Na przykład obiektem może być pozycja menu, lub przycisk wraz z przypisanymi mu zestawami komend lub wykazem danych. Obiektem jest także baza danych czy obraz graficzny albo tablica sporządzona w programie *Excel*. Jeszcze ważniejsze jest to, że oprócz wspomnianych powyżej standardowych obiektów języka *Visual Basic* możemy stworzyć nieograniczoną liczbę obiektów własnych, które będą działać nie gorzej niż te standardowe, opracowane prze firmę Microsoft.

Dla modelowania taka możliwość tworzenia nowych klas obiektów jest bardzo ważna, gdyż upraszcza proces modelowania. Na przykład w programie FORKOME, prezentowanym w rozdziale 3, takim nowym obiektem jest TREE – drzewo. Każdy obiekt, zarówno zbudowany ze standardowych elementów języka *Visual Basic*, jak i zrodzony z potrzeb i wyobraźni programisty, może być zapisany w kodzie, a następnie użyty w utrwalonej lub dowolnie zmodyfikowanej postaci również w innym programie czy modelu. To wielka oszczędność czasu i pracy programisty!

Każdy obiekt ma określony zestaw cech (ang. *Properties*). Na przykład Obraz (Picture Box) ma właściwości: szerokość (Width), wysokość (Hight), widoczność w danej chwili (Visible), kolor tła (Back Color), rodzaj ramki (Border Style) i wiele innych, które wyliczone są w podręcznikach dla programistów.

Nowe obiekty też posiadają właściwości. Na przykład wspomniany obiekt TREE ma następujące właściwości: gatunek drzewa, średnica na wysokości 130 cm od ziemi, wysokość i wiek. W trakcie wykonywania programu właściwości obiektów mogą się zmieniać, na przykład na obrazie mogą pojawiać się najróżniejsze kolory, a drzewo może zwiększać wysokość i średnicę pnia.

Działania na obiektach zmieniające ich właściwości nazywają się metodami (ang. *Methods*). Na przykład obraz można wyczyścić, czyli usunąć (Cls), umieścić na nim punkt określonego koloru i w określonym miejscu (Pset), rozpoznać, jakim kolorem jest zabarwiony punkt o określonych współrzędnych (Point), wykreślić linię (Line), narysować okrąg (Circle) i wykonać wiele innych działań. Dla obiektu TREE jako metody możemy zaproponować: pojawienie się nowego drzewa z nasiona, wypadnięcie (śmierć) drzewa z różnych przyczyn, zmiana szybkości wzrostu pod wpływem warunków zewnętrznych itp. Wygoda takiego podejścia polega na tym, że po jednokrotnym „wyjaśnieniu komputerowi”, co to jest prawo Liebiga, i – oczywiście po sprawdzeniu, że komputer właściwie nas zrozumiał – można być pewnym, że przywoławszy w odpowiedni sposób nazwę „Liebiga", spowodujemy wykonanie przez komputer właśnie tego, co zostało powiązane z tym hasłem.

Tak więc posługując się metodą modelowania obiektowego nie używamy sposobów najwłaściwszych dla komputera i nie staramy się opisać zmiennych procesów w krajobrazie za pomocą wyrażeń algebraicznych, lecz „wymuszamy" na komputerze zrozumienie, co to takiego jest „drzewo", czym brzoza różni się od modrzewia, czy też – ile azotu zużywa się w procesie fotosyntezy. Oczywiście początkowe straty czasu na utworzenie naszych własnych obiektów są bardzo duże, ale później zwracają się z lichwiarskim zyskiem, gdyż rozmowa z komputerem będzie się odbywać w tym samym języku, jakim geobotanik rozmawia z geobotanikiem albo limnolog z limnologiem, bez tłumaczy i pośredników.

Przykłady takiego podejścia zostały już opisane w wielu publikacjach (na przykład: Sequeire et al. 1997; Acock, Reynolds 1997; Timlin, Pachepsky 1997).

7. Model pierśnica – wysokość drzewa DIAMHIGH

Dla zrozumienia reszty książki tak, jak chcieliby tego autorzy, należy teraz mieć przed sobą włączony komputer z systemem operacyjnym Windows 95 (lub wyższym) i programem *Visual Basic 6.0.* Do legalnej kopii *Visual Basica 6.0* jest dołączona obszerna instrukcja; prosimy użyć jej celem uruchomienia programu. Polecamy także bardzo gorąco niewielką książeczkę: Sławomir Żaboklicki. *Visual Basic. nauka programowania w szkole*. Wydawnictwo „Interbook": Szczecin 2002, 86 ss. Jest to bardzo dobry podręcznik! Szczególnie cenne jest polskie słownictwo, które zastosowaliśmy również w naszej książce. To samo wydawnictwo wydało też *Słownik informatyczny dla każdego*.

Visual Basic (*VB*), tak jak i inne języki programowania wysokiego poziomu, ma pewną właściwość, która podobno pomaga w użytkowaniu tych programów: prawie każdy kolejny krok w zapisywaniu i uruchomianiu może być wykonany na kilka sposobów! Co gorsza: wygląd ekranu po włączeniu *VB* zależy od tego, jakie ustawienie programu pozostawił poprzedni użytkownik w momencie wyłączenia. To są, oczywiście, „choroby dziecięce", ale nie powinny zniechęcać początkujących użytkowników. W dalszym opisie przyjmujemy, że *VB* w waszym komputerze ma ustawienia firmowe lub takie ustawienia wprowadził wasz „guru" komputerowy.

Po włączeniu *VB* na ekranie pojawiło się o k n o p o w i t a l n e New Project. W zakładce New podświetlony jest typ programu: Standard EXE. Klikamy lewym przyciskiem myszy [Open] ([ ] – w takich nawiasach będziemy umieszczać nazwy przycisków na ekranie i klawiszy na klawiaturze). Otwiera się o k n o g ł ó w n e (rys. 7.1). Na ciemnoszarej (na waszym ekranie kolory mogą być inne!) przestrzeni znajduje się biały prostokąt o k n a f o r m u l a r z a (Form Designer) z napisem u góry: Project1 – Form1 (Form). Wewnątrz okna formularza znajduje się mniejszy prostokąt o k n a a p l i k a c j i (z napisem u góry: Form1. Z lewej strony ekranu widać o k n o p r z y b o r n i k a (ToolBox).

Powiększmy rozmiary obszaru roboczego w oknie Form1. Umieśćmy wskaźnik myszy na jednej z krawędzi okna formularza Project1 – Form1 (Form), aby zmienił się we wskaźnik zmiany rozmiaru: ↕. Kliknimy lewym przyciskiem myszy i trzymając przycisk zwiększamy okno do granicy szarego prostokąta. Zróbmy to samo z drugą krawędzią okna. Następnie umieśćmy wskaźnik zmiany rozmiaru w prawym dolnym narożniku okna Form1 i zwiększymy je do granic okna formularza.

Jesteśmy gotowi do rozpoczęcia pisania naszego pierwszego programu.

Załóżmy, że chcemy obliczyć wysokość drzewa (H) w zależności od jego średnicy według wzoru:

$$H = (B_2 * D - B_3 * D^2) + 1.3$$

gdzie:

D – średnica drzewa na wysokości piersi człowieka (pierśnica),
B_2 i B_3 – współczynniki.

Aby być bliżej rzeczywistości będziemy obliczać wysokość dębu szypułkowego (*Quercus robur* L.) o następujących współczynnikach (Botkin et al. 1972):

$$H = (35 * D - 0.07 * D^2) + 1.3$$

Zaczniemy od utworzenia na obszarze roboczym aplikacji Form1 p r z y c i s k u p o l e c e n i a (CommandButton) o nazwie „START". Robimy to w następujący sposób:

Jeżeli okno przybornika (Toolbox) nie jest widoczne na ekranie, należy: kliknąć pozycję menu [View], a w rozwiniętym submenu kliknąć [Toolbox]. Teraz przeniesiemy na obszar roboczy przycisk polecenia. Klikamy ikonę CommandButton w oknie przybornika (poziomy prostokąt bez napisów; zapamiętajmy: „zawieszenie" wskaźnika myszy nad ikoną wyświetla jej opis). Zwalniamy przycisk, a następnie klikamy w wybranym miejscu na obszarze roboczym i nie odpuszczając (zwalniając) przycisku myszy rysujemy kursorem punktowany prostokąt potrzebnej wielkości. Odpuszczamy przycisk. Zarówno rozmiary, jak i położenie przycisku można łatwo zmienić: klikamy jeden z uchwytów na krawędzi przycisku i zmieniamy wielkość albo klikamy w obrębie przycisku i przesuwamy go. Ufff! Te działania są znacznie prostsze od ich opisu!

Na utworzonym przycisku znajduje się napis Command1.

W ten sposób utworzyliśmy obiekt programu *VB*, któremu należy nadać „imię własne" i przypisać inne niezbędne właściwości. Klikamy prawym (a nie lewym!) przyciskiem myszy w obrębie przycisku. Pojawia się submenu (Cut-Copy-...-Properties). Klikamy [Properties], czyli c e c h y; w prawym oknie pojawia się spis cech. Znajdujemy wiersz n a z w y [Name] i klikamy go (szybko!) dwa razy, a następnie na klawiaturze naciskamy [Delete]. Z prawego okna zniknął napis Command1; wpisujemy tam cmbStart; (cmb to skrót od CommandButton). Uwaga: jeżeli wiersza z nazwą Name nie widać od razu, należy za pomocą przycisków [▼] i [▲] (z prawej strony okna z cechami) odnaleźć wiersz [Name].

Tak więc nadaliśmy obiektowi-przyciskowi nazwę; należy teraz wpisać na przycisku informację o jego funkcji. Znajdujemy w spisie wiersz [Caption], klikamy go dwa razy, naciskamy na klawiaturze [Delete] i wpisujemy z prawej START. Równocześnie zmienia się napis na przycisku (rys.7.1), a u dołu pojawia się aktualne polecenie – Caption.

Utworzyliśmy „opakowanie" obiektu, a teraz dodamy zawartość. Klikamy dwa razy przycisk START. Na ekranie pojawia się zawartość procedury (u góry jej nazwa: cmbStart):

```
Private Sub cmbStart_Click ( )
End Sub
```

Pierwszy wiersz powyższej procedury wskazuje, że to, co będzie wpisane poniżej, powinno zostać wykonane (policzone) po naciśnięciu przycisku START. Wpisujemy za mrugającym kursorem nasz wzór dla wyliczenia wysokości dębu:

```
Private Sub cmbStart_Click()
   H = (B2 * D – B3 * D ^ 2) / 10 + 1.3 ' Nowo wpisany wiersz
End Sub
```

W języku *VB* symbol: * oznacza mnożenie, natomiast symbol:^ podnoszenie do potęgi. Naciśnijmy [Enter]; wzór zostanie odpowiednio sformatowany. A co będzie, jeżeli wzór zawiera błąd? Spróbujmy: wpisujemy dwa razy ^^ i naciśnijmy [Enter]. Wzór wyświetlił się na czerwono i pojawił się komunikat: Compile error. Naciskamy [OK], poprawiamy wzór i naciskamy [Enter].

Co dalej ze wzorem na wysokość dębu? Trzeba komputerowi wyjaśnić, że H, D, B2, i B3 to liczby zmiennoprzecinkowe, jak na przykład 2,23 lub 0,00280, a nie liczby całkowite (bez ułamka).

Musimy umieć wywoływać na ekran okno tekstu programu lub okno formy. W oknie z prawego brzegu ekranu pod napisem u góry Project – Project1 znajduje się pasek menu z trzema prostokątami. Kliknięcie lewego prostokąta włącza okno tekstu programu Project1-Form1(Code), a kliknięcie środkowego – włącza okno formy Project1-Form1(Form). Takie same rezultaty uzyskamy, klikając w głównym menu [Viev], a następnie [Code] lub [Object].

U góry formy są dwa wąskie okna – lewe: cmbStart [▼], prawe: Click. Klikamy w lewym przycisk [▼]. Pojawia się menu, klikamy General i za migającym kursorem wpisujemy:

```
Option explicit
Dim H, D, B2, B3 As Single
```

Naciskamy [Enter]

Powiedzieliśmy komputerowi, że H, D, B2 i B3 to są liczby, a nie cokolwiek innego. Chodzi o to, że komputer potrafi operować nie tylko liczbami, lecz również słowami i innymi bardziej złożonymi konstrukcjami, ale o tym później. Teraz należy nadać współczynnikom B2 i B3 konkretne wartości, charakterystyczne właśnie dla dębu szypułkowego, a nie dla klonu czy buka. Również średnicę dębu (D) należy skonkretyzować, na przykład 20 cm. W oknie z lewej u góry znów klikamy [▼], a w pojawiającym się menu klikamy Form i wpisujemy za kursorem:

```
Private Sub Form_Load ( )
  D = 20
  B2 = 35
  B3 = 0.07
End Sub
```

Teraz po uruchomieniu programu tym symbolom zostaną przypisane podane wyżej wielkości. Jeżeli nie zapiszemy powyższej procedury, *VB* przypisze tym symbolom wartości zerowe.

Utworzymy jeszcze jeden przycisk, który nazwiemy CLOSE. Robimy to w identyczny sposób jak przy tworzeniu przycisku START. Nadamy przyciskowi n a z w ę (Name) cmbClose i n a p i s (Caption) CLOSE. Przechodzimy do okno tekstowego programu (jak opisano wyżej).

Stwórzmy procedurę, która zadziała po naciśnięciu przycisku o nazwie cmbClose. Klikamy dwa razy przycisk [CLOSE]. Na ekranie pojawia się zawartość procedury z migającym kursorem. Dopisujemy do programu następującą procedurę:

```
Private Sub cmbClose_Click ( )
  End
End Sub
```

Działanie tej procedury jest bardzo proste: zatrzymuje działanie programu.

Przechodzimy na ekran formy. Należy teraz uwidocznić wyniki naszych obliczeń. *VB*, podobnie jak i w innych sytuacjach, proponuje do tego różne sposoby. Wybieramy zastosowanie etykiet (Label). W przyborniku (Toolbox) ikoną etykiety jest prostokąt z literą A. Przeniesiemy go, zupełnie tak samo jak poprzednio przycisk polecenia (CommandButton) z ramki przybornika na pole formy. *VB* automatycznie napisze w ramce Label1 i nada nazwę Label1. Zmienimy tę nazwę. W spisie Properties należy kliknąć Name i pole z prawej zmienić na lblHEIGHT. Z pola etykiety usuńmy Label1, klikając dwa razy właściwość Caption i na klawiaturze naciskając [Delete].

Temu, co ukaże się w polu etykiety, możemy nadać ładniejszy wygląd. Na środku krawędzi etykiety powinny być widoczne małe kwadraty (okno etykiety jest aktywne) – jeżeli ich nie widać, należy kliknąć pole etykiety.

Kliknijmy cechę Alignement (wyrównanie) i z prawej strony [▼]. Na rozwiniętej liście kliknijmy opcję Center.

Dodajmy ramkę – kliknijmy cechę Border Style i [▼], a na liście [1-Fixed Single].

Ustawmy rodzaj czcionki – trzeba znaleźć cechę Font i kliknijmy ją dwa razy. W oknie Czcionka kliknijmy Arial, styl Pogrubiony i rozmiar 26. Klikamy [OK].

Wróćmy do tekstu programu. Dopiszmy polecenie odsyłające wynik obliczeń do zrobionej właśnie etykiety. W głównej procedurze ustawiamy kursor na końcu drugiego wiersza (`H = ... 1.3`) i naciskamy [Enter]. Za kursorem wpisujemy nowy wiersz jak poniżej:

```
Private Sub cmbStart_Click()
  H = (B2 * D – B3 * D ^ 2) / 10 + 1.3
  lblHEIGHT.Caption = Format(H, „#####.#”)
End Sub
```

Operator `Format` przekształca liczbę w wiersz i ogranicza liczbę znaków po kropce (tutaj: jeden znak). Zwróćmy uwagę, że nazwa obiektu (`lblHEIGHT`) oddzielona jest od jego właściwości za pomocą kropki.

Uruchamiamy program klikając Run w głównym menu *VB* i Start w rozwijającym się submenu. Wyświetli się ekran formy; kliknijmy przycisk [START]. Otrzymany wynik: wysokość dębu = 690 cm, czyli prawie 7 m, jest w pełni prawdopodobny. W opisanej postaci nasz model dobrze zilustrował sposób konstruowania najprostszego programu „obiektowego”, ale jego przydatność praktyczna jest niewielka. Dodamy więc kilka ulepszeń.

Pierwsze ulepszenie polega na dodaniu poziomego paska przewijania (HorizontalScrollBar), który pozwoli na wprowadzanie średnicy (D) wprost z ekranu. W tym celu należy znaleźć w przyborniku (ToolBox) ikonę paska przewijania. Przypomnijmy, że „zawieszenie” kursora nad ikoną wyświetli po krótkiej chwili jej nazwę. W okolicy połowy wysokości przybornika są poziomo obok siebie dwie ikony pasków; kursor na lewym poziomym wyświetli nazwę HScrollBar, a na prawym, poziomym – VScrollBar (VerticalScrollBar, czyli pionowy pasek przewijania). Dalej należy postępować dokładnie tak samo, jak przy tworzeniu przycisków START lub STOP, to znaczy zakreślić kursorem odpowiednie miejsce na polu roboczym formy, kliknąć prawym przyciskiem myszy utworzony prostokąt, następnie kliknąć Properties i w rubryce Name wpisać hscDIAMETER, w rubryce Min wpisać 5, w rubryce Max natomiast 200. W ten sposób określiliśmy minimalną i maksymalną średnicę dębów, z którymi będziemy mieli do czynienia.

Umieśćmy jeszcze na polu roboczym etykietę, która będzie wyświetlała średnicę dębu: lblDIAMETER; robimy to analogicznie do opisanego już umieszczania etykiety lblHeight.

Teraz uzupełnimy nasz program poleceniami, które spowodują, że przemieszczanie na pasku przewijania jego środkowej części ruchomej (suwaka) będzie powodować zmiany liczb w oknie etykiety lblDIAMETER. Przejdźmy do okna tekstu programu i dopiszmy następującą procedurę:

```
Private Sub hscDIAMETR_Change()
  D = hscDIAMETR.Value
  lblDIAMETR.Caption = Format(D, „###")
End Sub
```

Funkcja `Format` przekształca liczbę określającą średnicę dębu w wiersz, który pojawia się w polu etykiety lblDIAMETER. Zanim uruchomimy program, posłużymy się paskiem przewijania. Ustawmy suwak w skrajnie lewym położeniu – etykieta średnicy wyświetli 5 cm. Uruchomimy program klikając Run w menu *VB* i Start; wysokość dębu = 1.9 m. Ekran będzie wyglądał jak na rys. 7.2.

Moglibyśmy jeszcze na przykład dodać bazę danych zawierającą współczynniki dla innych gatunków drzew i dopisać procedurę wprowadzania tych współczynników, ale... dla poznania sposobu budowy bardzo prostego programy to, co zrobiliśmy do tej pory, wystarczy.

Napiszmy pełny tekst naszego programu:

```
Option Explicit
  Dim D, H, B2, B3 As Single
Private Sub Form_Load()
  B2 = 35
  B3 = 0.07
End Sub
Private Sub cmbStart_Click()
  H = (B2 * D – B3 * D ^ 2) / 10 + 1.3
  lblHEIGHT.Caption = Format(H, „###")
End Sub
Private Sub cmbStop_Click()
  End
End Sub
Private Sub hscDIAMETR_Change()
  D = hscDIAMETR.Value
  lblDiameter.Caption = Format(D, „###")
End Sub
```

Na zakończenie zapiszmy nasz program. *VB* zapisuje w dwóch osobnych plikach (1) kod formy i obiekty oraz (2) listę elementów (okno Project). Kliknijmy w menu File, a w otwartym submenu Save Project As; otworzy się okno Save File As. Wybierzmy folder (albo utwórzmy nowy). Wpiszmy nazwę, na przykład Piersnica1.frm (inne wersje projektu można zapisywać Piersnica2.frm itd. Kliknijmy [Zapisz]. Otworzy się okno Save Project As. Wpiszmy nazwę Piersnica1. Kliknijmy [Zapisz].

8. Model obliczeń parametrów drzew – TREEPAR

Ten rozdział zawiera modyfikację i dalsze rozwinięcie programu, jaki był rozpatrywany w rozdziale 7. Dla modeli płatowych, omawianych w rozdziale 3, konieczne są parametry B_2 i B_3, które można oznaczyć, jeżeli są znane maksymalna wysokość drzewa (H_{max}) i maksymalna średnica (D_{max}), na podstawie wzorów (Botkin et al. 1972):

$$B_2 = 2\left(\frac{H_{max} - 130}{D_{max}}\right)$$

$$B_3 = \left(\frac{H_{max} - 130}{D_{max}^2}\right)$$

Zaczniemy od utworzenia procedury, która wylicza współczynniki B_2 i B_3 na podstawie powyższych wzorów, a wyniki wyświetla na etykietach lblB2 i lblB3.

```
Private Sub CALCULATION()
   B2 = 2 * (Hmax – 130) / Dmax
   B3 = (Hmax – 130) / (Dmax ^ 2)
   lblB2.Caption = Format(B2, „##.#")
   lblB3.Caption = Format(B3, „#.###")
End Sub
```

Zakładamy, że Czytelnicy nauczyli się z poprzedniego rozdziału, jak tworzyć etykiety i procedury, jak i co klikać i naciskać, i nie będziemy się tym dalej w tekście zajmować.

Dane obejmujące maksymalne średnice i wysokości wybranych gatunków drzew zawarte są w tab. 3.1 w rozdziale 3 pierwszej części książki.

Wprowadzimy te dane do programu w postaci macierzy. Tworzony program będzie miał za zadanie, po wprowadzeniu nazwy gatunkowej, wyświetlać współczynniki B2 i B3, a także pokazywać średnice tego drzewa.

VB, jak zwykle, daje wiele sposobów rozwiązania tego zadania. Wybierzemy wprowadzanie gatunku drzewa za pomocą przycisków (Option Button). W tym celu zaczniemy od utworzenia ramki (Frame) i nazwiemy ją Species. Funkcja tej ramki polega na tym, że spośród wszystkich przycisków znajdujących się w ramce działa tylko ten, który został naciśnięty jako ostatni. Nie musimy więc kasować już nie potrzebnych przycisków.

Teraz wewnątrz ramki Species ulokujemy 14 przycisków z nazwą gatunków drzew. Łacińska nazwa gatunku jest cechą przycisku o nazwie Caption, a nazwę obiektu-przycisku nadamy od opt(0) do opt(11), to znaczy z indeksami od 0 do 11. Indeksy można wstawiać ręcznie lub automatycznie, ale w tym ostatnim przypadku należy uważać, żeby indeksy były prawidłowo nadane.

Dla przycisków poleceń START i STOP wykorzystamy „prefabrykaty" z poprzedniego programu, lecz zmienimy procedurę cmbStart na polecenia poszukiwania w matrycy danych o maksymalnej średnicy i wysokości drzewa na zasadzie: który przycisk został w tym momencie naciśnięty.

```
Private Sub cmbStart_Click()
Dim i As Integer
For i = 0 To 11
     If opt(i) = True Then
        Hmax = Hmax_db(i + 1)
        Dmax = Dmax_db(i + 1)
        Exit For
     End If
Next i
   lblH.Caption = Format(Hmax, „###")
   LblDIAMETR.Caption = Format(Dmax, „###")
   Call CALCULATION
End Sub
```

Ta procedura działa bardzo prosto: przegląda indeksy gatunków drzew, począwszy od 0, i kiedy natrafi na naciśnięty przycisk (`opt(i) = True`), zapamiętuje charakterystykę drzewa i wychodzi z cyklu (`Exit For`). Następnie te charakterystyki wysyłane są na etykiety do obejrzenia i przesyłane do procedury `CALCULATION`, którą omówiliśmy wcześniej.

Teraz o obrazkach drzew. *VB* ma specjalny typ obiektów o nazwie Image, służący do lokowania wizerunków. Same wizerunki można tworzyć na przykład za pomocą programu *Paint*, który jest składnikiem systemu *Windows*, lub – jak to było tym razem – pobrać wizerunki drzew z jakiegoś podręcznika botaniki za pomocą skanera, który przekształca obrazek w plik rastrowy (bmp). Przykład wizerunku dębu znajduje się na rys. 8.1.

Sporządziliśmy macierz wizerunków od img(0) do img(11), z indeksami koniecznie w tej samej kolejności, co indeksy przycisków, aby po naciśnięciu przycisku [Betula pendula] ukazywał się wizerunek brzozy, a nie innego drzewa. Związek obiektu z plikiem zawierającym rysunek powstaje za pomocą cechy Picture. Na przykład:

```
Image(1).picture = c:/My_documents/Alnus_incana.bmp
```

Obrazki nie powinny pojawiać się na ekranie wszystkie naraz. Dlatego trzeba im przypisać właściwość widzialności Visible jako False. Utworzymy obiekt ImgMAIN, w którym będziemy przekazywać potrzebny obrazek i który będzie stale widoczny. Umieścimy ten obiekt w prawej części formy; pozostałe obrazy można umieszczać gdziekolwiek, gdyż nie są widoczne w czasie działania programu. Teraz do procedury poszukiwania należy wpisać operator przekazania obrazka do widzialnej prezentacji.

```
ImgMAIN.Picture = img(i).Picture
```

Na koniec udoskonalimy program tak, żeby było można obliczać współczynniki B2 i B3 dla dowolnych kombinacji Hmax i Dmax. W tym celu dodamy dwa poziome paski przewijania i przypiszemy im następujące procedury:

```
Private Sub hscDIAMETR_Change()
   Dmax = hscDIAMETR.Value
   LblDIAMETR.Caption = Format(Dmax, „###")
   Call CALCULATION
End Sub
Private Sub hscHEIGHT_Change()
   Hmax = hscDIAMETR.Value
   LblDIAMETR.Caption = Format(Hmax, „###")
   Call CALCULATION
End Sub
```

Zwróćmy uwagę, że zawarty w powyższym programie sposób przechowywania i poszukiwania danych jest przydatny jedynie w takich przypadkach, gdy mamy do czynienia z niewielkimi macierzami. Kiedy ilość danych przekracza kilka dziesiątków pozycji, należy posłużyć się programami przeznaczonymi specjalnie dla budowy baz danych – na przykład *Excel* czy *Access*, wchodzącymi w skład pakietu *Microsoft Office* i mającymi wygodną łączność z programem *Visual Basic.*

9. Uproszczony model systemu leśnego jako zbioru drzew – FOREST

Ten rozdział jest pomyślany jako przygotowanie do konstruowania programów bardziej skomplikowanych. Zapoznamy się w nim ze sposobami tworzenia menu, nowych obiektów w rodzaju GRASS (trawa) i TREE (drzewo), jak również z najprostszymi działaniami w tworzeniu grafiki komputerowej.

Niech obiekt GRASS ma takie cechy jak współrzędne (`coor`) i wysokość (`height`), a obiekt TREE ma takie cechy jak przynależność gatunkowa (`species`), współrzędne (`coor`) i wysokość (`height`).

Dla naszych zadań dydaktyczno-ilustracyjnych w pełni wystarczy zadeklarowanie macierzy składających się z 10 drzew i 100 źdźbeł trawy.

```
Private Type GRASS
  coor As Single
  height As Single
End Type
Private Type TREE
  coor As Single
  height As Single
  species  As Integer
End Type
Dim tr(10) As TREE
Dim g(100) As GRASS
Dim imax As Integer
```

Obecnie zajmiemy się sporządzeniem menu. Mamy włączony *VB*, a na ekranie włączony jest widok *Form* (a nie *Code*). Klikamy [Tools] w głównym menu *VB*. W submenu klikamy [Menu Editor]. Na ekranie Menu Editor znajdują się między innymi okna Caption i Name; w pierwszym trzeba umieścić to, co będzie napisane na formie, na przykład Initial State, w drugim zaś nazwę procedury, która będzie wykonana po kliknięciu na utworzoną pozycję menu, na przykład `mnu_Initial_State`:

```
Private Sub mnu_Initial_State_Click()
Dim i As Integer
     tr(1).species = 1
     tr(1).coor = 15
     tr(1).height = 30
     tr(2).species = 2
```

```
       tr(2).coor = 15
       tr(2).height = 30
     Call Drow_Tree(1)
      Call Drow_Tree(2)
      For i = 1 To grass_initial
      Call Create_GRASS(100 * Rnd, 10)
       Grass2.Line  (g(i).coor,  hmax)-(g(i).coor,  hmax  –
g(i).height)
     Next i
  End Sub
```

Procedura `mnu_Initial_State` tworzy dwa drzewa oraz 100 (`grass_initial`) źdźbeł trawy. Utworzenie źdźbła następuje w procedurze `Create_GRASS`, a jego obraz graficzny jest bardzo prosty – odcinek pionowej linii prostej o długości odpowiadającej wysokości źdźbła.

```
  Private Sub Create_GRASS(ByVal XX As Single, ByVal HH As
Single)
      imax = imax + 1
      g(imax).coor = XX
      g(imax).height = HH
  End Sub
```

Oprócz źdźbeł traw procedura `mnu_Initial_State` tworzy też dwa drzewa (rys. 9.1) – jedno liściaste (`tr(1).species = 1`), a drugie iglaste (`tr(2).species = 2`). Osobna procedura – **`Draw_Tree`** – wykonuje wizerunki drzew:

```
  Private Sub Draw_Tree(ByVal i As Integer)
  Dim X, H, h1, h2 As Single
  Dim k As Integer
  H = tr(i).H
  X = tr(i).X
   Select Case tr(i).species
     Case 1
       Grass2.Line (X, hmax)-(X, hmax – 0.4 * H)
       Grass2.Circle (X, hmax – H + 0.3 * H), 0.2 * H, , , , 2
     Case 2
       Grass2.Line (X, hmax)-(X, hmax – H)
    For k = 1 To 5
        h1 = 0.1 * k
        h2 = h1 + 0.2
        Grass2.Line  (X,  hmax  –  H  +  h1  *  H)-(X  –  0.1  *  H,
hmax – H + h2 * H)
        Grass2.Line  (X,  hmax  –  H  +  h1  *  H)-(X  +  0.1  *  H,
hmax – H + h2 * H)
     Next k
   End Select
```

```
End Sub
```

Rysowanie liściastego drzewa sprowadza się do nakreślenia pionowej linii i elipsy (rys. 9.2). Zwraca uwagę, że w metodzie Circle po współrzędnych środka (w nawiasie) występuje wartość promienia, a ostatnim argumentem jest spłaszczenie elipsy. Trzy wolne miejsca oznaczają, że parametry koloru linii oraz początku i końca łuku przyjmuje się jako domyślne (kolor czarny, a linia elipsy ciągła). Szczegółów na temat rysowania kół i elips prosimy szukać w podręczniku *VB*. Zachęcamy Czytelników do przećwiczenia rysowania kół, łuków i elips.

Drzewo iglaste to pionowa linia i kilka pochyłych linii imitujących gałęzie (rys. 9.3). `h1` i `h2` to względne współrzędne pionowe początku i końca gałęzi.

Zwraca uwagę, że druga gałąź jest, wobec pnia, lustrzanym odbiciem pierwszej, a w operatorze `Line` zmienia się tylko znak w przyroście współrzędnej horyzontalnej. Parametr cyklu (`k`) decyduje o liczbie gałęzi, którą można zwiększyć.

Kliknięcie [Run] uruchamia procedurę `mnuRun`, która wykonuje przyrost trawy i drzew. Przyrosty te są stałe w czasie. Polecenie `Grass2.Cls` czyści ekran przed rysowaniem nowych obrazów.

```
Private Sub mnuRun_Click()
  Dim i As Integer
  tr(1).H =  tr(1).H + 5
  tr(2).H =  tr(2).H + 6
  For i = 1 To imax
   g(i).H = g(i).H + 3
  Next i
  Grass2.Cls
  Call Drow_Tree(1)
  Call Drow_Tree(2)
  For i = 1 To grass_initial
    Grass2.Line(g(i).coor,  hmax)  -  (g(i).coor,  hmax  –
    g(i).height)
  Next i
End Sub
```

Procesy ginięcia roślin i ich rozrodu będą tematem kolejnych modeli.

10. Poznajmy się bliżej – model STOCHASTIC

Już w rozdziale 1 były rozpatrywane układy stochastyczne. Ten rozdział ma praktycznie zaznajomić z komputerową imitacją zdarzeń losowych (model STOCHASTIC – zob. CD dołączony do książki). *VB* ma wbudowaną funkcję liczby losowej (`Rnd`) o równomiernym rozkładzie od 0 do 1. Wykorzystując formę programu z poprzedniego rozdziału, stworzymy populację traw z równomiernie rozłożonymi wysokościami:

```
h(i) = hmax * Rnd
```

Zbudujemy funkcję rozkładu wysokości źdźbeł trawy. W tym celu należy całą pulę wysokości poszczególnych źdźbeł rozbić na klasy wysokości (na przykład na `dmax = 10`) i policzyć liczbę źdźbeł trafiających do każdej klasy. Można to zrobić na przykład tak:

```
Private Sub DISTRIBUTION()
   Dim i As Integer
   Dim k As Integer
   Dim hm As Integer
   hm = 0
   Dis.Cls
   Dis.ScaleTop = 0
   Dis.ScaleLeft = 0
   Dis.ScaleWidth = dmax
For k = 1 To dmax
   d(k) = 0
Next k
For i = 1 To imax
   For k = 1 To dmax
     If h(i) > (hmax / dmax) * (k – 1) And h(i) <= (hmax /
dmax) * k Then
        d(k) = d(k) + 1
      End If
    Next k
Next i
For k = 1 To dmax
   If hm < d(k) Then
      hm = d(k)
   End If
Next k
Dis.ScaleHeight = 1.2 * hm
For k = 1 To dmax
```

```
  Dis.Line (k – 1, Dis.ScaleHeight)-(k, Dis.ScaleHeight –
d(k)), RGB(0, 255, 0), BF
Next k
End Sub
```

W powyższej procedurze obliczana jest funkcja rozkładu wielkości `h(i)`, a następnie prezentowany jest histogram tego rozkładu na wykresie o nazwie `Dis`. Prosimy zwrócić uwagę na to, żeby macierz `d(k)`, w której następuje podliczanie źdźbeł, na początek zapełnić zerami. Jeżeli tego nie zrobimy, to procedura zadziała prawidłowo jedynie pierwszy raz, a wszystkie kolejne jej wywołania będą błędne, gdyż w macierzy pozostawać będą wyniki poprzednich obliczeń.

Pionowa skala rysunku `Dis` jest zmienna, gdyż nie możemy z góry przewidzieć, ile będzie źdźbeł w klasach wysokości. Dlatego wstępnie oblicza się maksymalną liczbę źdźbeł w klasie `hm`, a skalę pionową (`Dis.ScaleHeight`) robi się o 20% większą od tej liczby, aby słupki histogramu nie opierały się o górny brzeg wykresu.

Po uruchomieniu programu przekonamy się, że przy liczbie źdźbeł 100 wysokość słupków będzie w istotny sposób różnić się od rozkładu teoretycznego. Dopiero zwiększenie próby do 1000 lub jeszcze lepiej do 10 000 źdźbeł spowoduje, że wysokość słupków będzie praktycznie jednakowa. Zmianę wielkości próby w tym programie osiąga się po prostu zmieniając wielkość `imax`.

W naturze, tak w ogólności, nie spotykamy równomiernego rozkładu źdźbeł; znacznie bliższy rzeczywistości jest rozkład normalny. Aby uzyskać normalny rozkład w naszym programie komputerowym, możemy posłużyć się znanym wzorem:

$$\varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}}$$

Jednakże bardziej eleganckie będzie zastosowanie centralnego twierdzenia granicznego teorii prawdopodobieństwa, które głosi, że przy zsumowaniu wystarczająco dużej liczby niezależnych losowych wielkości o takich samych zasadach rozkładu otrzymuje się losową wartość o rozkładzie zbliżonym do normalnego. Praktycznie wystarczy sześć takich wartości (w zapisie *VB*):

```
h(i) = hav+ Sqr(2) * sigma * (Rnd + Rnd + Rnd + Rnd + Rnd
+ Rnd – 3)
```

gdzie:

`hav` – średnia wysokość źdźbła,
`sigma` – dyspersja,
`Sqr(2)` – po prostu pierwiastek z 2.

Przełączanie z rozkładu równomiernego na normalny i z powrotem łatwo zrobić, stosując – znaną już z rozdziału 7 – technikę przycisków, tak jak to widać na rys. 10.1.

Przy małych wartościach `hav` i wielkich `sigma` można otrzymać dość dużo źdźbeł o ujemnej wysokości. Czegoś takiego – jak dotąd – w naturze nie zaobserwowano. Dlatego koniecznie trzeba wprowadzić do programu zakaz pojawiania się ujemnych źdźbeł! W takim jednak przypadku będziemy mieli do czynienia nie z normalnym rozkładem, a z tak zwanym rozkładem normalnym uciętym.

11. Model losowy populacji traw – GRASS

W rozdziale 9 rozważano proces wzrostu trawy, ale zakładano przy tym, że źdźbła trawy mogą zwiększać swoje rozmiary w sposób losowy. Uczyńmy ten model bardziej złożonym przez wprowadzenie procesów kiełkowania trawy z nasiona i umieranie osobnika (zob. CD dołączony do książki).

Zaczniemy od tego, że spróbujemy wyjaśnić komputerowi, co to jest „trawa" – jest to coś, co rośnie wraz z współrzędną `coor` i ma wysokość `height`. Tak też zapiszemy:

```
Private Type GRASS
   coor As Single
   height As Single
End Type
```

Słowa `As Single` oznaczają, że zarówno wysokość trawy, jak i jej współrzędne są liczbami posiadającymi część ułamkową. Zakładamy, że `nmax = 100` źdźbeł w pełni wystarcza dla naszych eksperymentów. Informacja o tym wygląda następująco:

```
Dim g(nmax) As GRASS
```

A więc przygotowaliśmy w pamięci komputera miejsce dla 100 źdźbeł, ale tych źdźbeł jeszcze tam nie ma. Aby stworzyć nowe źdźbło, trzeba wskazać miejsce, gdzie ma ono rosnąć (`XX`) i jaką ma mieć wysokość (`HH`). Jeżeli założymy, że mamy już `imax` źdźbeł, to stworzenie nowego źdźbła można powierzyć następującej procedurze:

```
Private Sub Create_GRASS(ByVal XX As Single, ByVal HH As
Single)
    If imax < nmax Then
       imax = imax + 1
       g(imax).coor = XX
       g(imax).height = HH
    End If
End Sub
```

W nagłówku procedury opisano szczegółowo typ argumentów i sposobów ich wywoływania (`ByVal` – to informacja dla *VB*, żeby zachował kopię oryginalnego argumentu i zwrócił ją nie zmienioną, gdy procedura zakończy działanie, i to nawet jeżeli zmienna została zmodyfikowana w procedurze). W zasadzie można tego nie robić, ale „strzeżonego Pan Bóg strzeże".

Jeżeli chcemy zacząć pracę naszego programu z 25 źdźbłami (`i_initial = 25`), wysokość każdego 10 mm (`h_initial = 10`) i rozmieścić te źdźbła losowo na powierzchni 100 cm (`space = 100`), to w procedurze, którą wywołujemy klęknięciem [Initial State] w menu, należy wpisać:

```
For i = 1 To i_initial
  Call Create_GRASS(space * Rnd, h_initial)
Next i
```

Pierwszy krok zrobiony. Teraz jest odpowiedni moment, aby omówić strukturę całego programu. Rys. 11.1 prezentuje schemat blokowy algorytmu modelującego. Zwróćmy uwagę na kształt bloków, który nie jest dowolny, lecz odpowiada międzynarodowym zasadom graficznej prezentacji schematów programów. Prostokąt z zaokrąglonymi kątami oznacza początek lub koniec programu. Prostokąt o ostrych kątach oznacza obliczenia lub inne działania nie związane z rozgałęzianiem procesu kalkulacji. Wydłużony sześciokąt oznacza cykl, a romb – operator warunkowy. Jeżeli warunek zapisany wewnątrz rombu spełnia się, wówczas strzałka kierująca ma napis Yes. W przeciwnym przypadku na strzałce jest napis No.

Załóżmy, że model ma przejść `tmax` kroków czasowych. Temu działaniu odpowiada na schemacie blokowym podłużny sześciokąt z napisem While t < max, a w samym programie następujący fragment:

```
For t = 1 To tmax
  Call MODEL_ONE_STEP
Next t
```

Procedura `MODEL_ONE_STEP` zawiera opis wszystkiego, co dzieje się w modelu w ciągu jednego kroku czasowego. Źdźbła trawy giną, rosną i rozmnażają się. Zacznijmy od umierania. Aby imitować śmierć źdźbła trawy oznaczonej indeksem `i`, trzeba ją wykreślić ze spisu źdźbeł. Najprościej jest to zrobić, wstawiając na jej miejsce w spisie ostatnie źdźbło i zmniejszając liczbę źdźbeł o jeden.

```
Private Sub Delete_Grass(ByVal i As Integer)
  If imax > 0 Then
    g(i).x = g(imax).x
    g(i).h = g(imax).h
     imax = imax – 1
  End If
End Sub
```

Zwróćmy uwagę na to, że procedura będzie cokolwiek robić, jeżeli w spisie jest choćby jedno źdźbło; jeśli go nie ma, to cóż ma ginąć? Dla normalnego człowieka sprawa wydaje się i bez tego jasna, ale komputer wymaga takiego wytłumaczenia, bo bez tego zaczną pojawiać się „ujemne trawy" i podobne dziwa.

Załóżmy, że każde źdźbło ginie w ciągu jednego kroku czasowego z prawdopodobieństwem `mort`. Wtedy procedura `MODEL_ONE_STEP` ma postać:

```
Private Sub MODEL_ONE_STEP
 Dim i As Integer
 i = 1
 Do
   If mort > Rnd Then
      Call Delete_Grass(i)
   Else
      Call Growth_Grass(i)
     i = i + 1
   End If
   If i = imax Then
      Exit Do
   End If
 Loop
 Call Peprod_Grass
End Sub
```

Zwróćmy uwagę na to, jak w modelu imitowane jest losowe zdarzenie śmierci źdźbła. Jeżeli prawdopodobieństwo zdarzenia jest większe niż bieżąca wartość `Rnd`, to zdarzenie zachodzi; w przeciwnym razie wszystko pozostaje po staremu. W naszym przypadku, jeżeli zachodzi zdarzenie śmierci źdźbła z indeksem `i`, włącza się procedura `Delete_Grass(i)`, w przeciwnym wypadku źdźbło może dalej spokojnie rosnąć (procedura `Growth_ Grass(i)`). Zauważmy, że w tym przypadku dla przeglądu wszystkich źdźbeł nie można zastosować polecenia `For...To...Next`, gdyż z góry nie wiadomo, ile źdźbeł pozostanie pod koniec kroku czasowego. Dlatego zastosowano polecenie `Do...Loop` z wyjściem z cyklu po przeglądzie wszystkich źdźbeł. Szczegółowe informacje o operatorach cyklu można znaleźć w instrukcji *VB*, którą radzimy studiować jednocześnie z budowaniem z nami tego i innych modeli.

Pozostaje nam jeszcze zapoznanie się z procedurami wzrostu i pojawienia się traw.

```
Private Sub Growth_GRASS
 Dim i as Integer
  For i = 1 To imax
     g(i).h = g(i).h + k_growth * Rnd
   Next i
 End Sub
```

```
Private Sub Reprod_GRASS
 Dim i as Integer
   For i = 1 To k_reprod
       Call Create_GRASS(space * Rnd, h_initial)
    Next i
 End Sub
```

W tych procedurach `k_growth` oznacza maksymalny przyrost źdźbła w ciągu jednego kroku czasowego, a `k_reprod` – liczbę źdźbeł rosnących w tym samym czasie.

Radzimy Czytelnikom przeprowadzić samodzielnie serię eksperymentów z tym programem. Na przykład warto zobaczyć, co się stanie, jeżeli trawa nie będzie się rozmnażać (`k_growth = 0`) lub będzie się rozmnażać bardzo intensywnie (`k_growth = 50`). Przekonacie się, że cała trawa rzeczywiście zginie, kiedy prawdopodobieństwo śmiertelności będzie równe jedności.

Nasz model odznacza się wieloma uproszczeniami, które Czytelnik może poprawić i zmodyfikować po swojemu. Na przykład w modelu zakłada się, że rozmieszczenie źdźbeł jest równomierne. W naturalnym środowisku tak nie bywa, gdyż jedno źdźbło od drugiego musi byś w pewnej minimalnej odległości. Przekonajcie się, jak bardzo skomplikuje się model po wprowadzeniu takiej, zdawałoby się, nieznacznej poprawki. W modelu szybkość wzrostu trawy i jej śmiertelność są wielkościami stałymi i niezależnymi zarówno od czasu, jak i wielkości łąki. Tak też się nie zdarza w naturze. Warto spróbować wprowadzić do modelu warunki środowiska zewnętrznego: temperaturę, wilgotność, ilość biogenów i ich wpływ na parametry wzrostu i śmiertelności. Jeszcze ciekawsze będzie wprowadzenie do modelu zależności zwrotnych, tak charakterystycznych dla warunków naturalnych. Na przykład szybkość wzrostu zazwyczaj maleje wraz ze wzrostem rozmiarów rośliny, a śmiertelność się zwiększa wraz ze wzrostem gęstości łąki. A gdy do modelu wprowadzony zostanie szereg gatunków traw o różnych właściwościach, to z całą pewnością powstanie model konkurencji i sukcesji, pełen różnorodności i nieoczekiwanych wyników.

12. Model krajobrazowy oparty na metodzie automatów komórkowych – LANDSCAPE

Pojęcie automatu komórkowego zostało już omówione w rozdziale 4. Niniejszy rozdział ma na celu pokazanie na prostym przykładzie, jak można tworzyć modele krajobrazowe, wykorzystując aparat automatów komórkowych.

Zaczniemy od skonstruowania siatki lub mapy. Wybierzmy siatkę składającą się z 100 kwadratów (10×10), jak na rys. 12.1. Moglibyśmy, w zasadzie, wybrać siatkę heksagonalną, składającą się z regularnych sześciokątów, lub siatkę złożoną z komórek różnych rozmiarów, ale ograniczymy się do najprostszego przypadku.

Siatkę do prezentowania wyników modelowania, z komórkami 10×10, możemy utworzyć na ekranie komputera w bardzo różny sposób. Na przykład możemy utworzyć 100 standardowych obiektów typu Shape (prostokąt). Oczywiście nie musimy 100 razy powtarzać całej operacji tworzenia obiektu – wystarczy utworzyć jeden obiekt, a dalej operować poleceniami Copy i Paste, przy czym indeksacja obiektów będzie wykonywana automatycznie. Można też wybrać sposób graficzny – operować rysunkiem siatki, znajdując za każdym razem współrzędne komórki.

W załączonym wariancie modelu (LANDSCAPE) utworzono `n = 100` obiektów Cell z indeksem od 0 do 99. Przejście od indeksu komórki do współrzędnych prostokątnych dokonywane jest według wskazówek znajdujących się w części `Form_Load` programu:

```
For i = 0 To n
   X(i) = (i Mod 10) + 1
   Y(i) = Int(i / 10) + 1
Next i
```

Przekształcenie odwrotne – poszukiwanie indeksu komórki na podstawie jej koordynat – dokonuje się za pomocą funkcji `COOR`.

```
Private Function COOR(ByVal xx As Integer, ByVal yy As In-
teger) As Integer
   COOR = (xx-1) + (yy - 1) * 10
End Function
```

Tutaj `X(i)` i `Y(i)` to współrzędne komórki z indeksem **`i`**, wyrażenie `i Mod 10` oznacza dzielenie przez modulo 10, czyli uwzględniamy tylko resztę z dzielenia. Na przykład mamy komórkę `X(15)`. Wtedy dzielimy 15 przez 10 i dostajemy 1,5, teraz 5 + 1 = 6, czyli `X(15)= 6`. Dla `Y(15)` reszta z dzielenia danej liczby przez 10 jest odrzucana przez operator `Int`. W tym przypadku dzielimy 15 przez 10 i też dostajemy 1,5, ale odrzucając resztę mamy 1+1=2, czyli `Y(15) = 2`.

Kolejny etap polega na utworzeniu procedury, która będzie odnajdywała komórki sąsiadujące z daną komórką, gdyż pojęcie sąsiada należy do definicji automatu komórkowego. W naszym przypadku przyjmiemy najprostszy schemat czterech sąsiednich komórek, jak to pokazano na rys. 12.1. W procedurze znajdywania sąsiadów muszą być przewidziane takie przypadki, kiedy komórka znajduje się na granicy siatki. W naszym modelu zakładamy, że w takim przypadku sąsiada nie ma, o czym informuje nadanie sąsiedniej komórce indeksu `-1`, a taki indeks po prostu nie może istnieć. Nie przeszkadza to ekologom zajmującym się „czystą nauką”. W ich teoretycznych modelach automatów komórkowych granic w ogóle nie ma. Graniczne prawe komórki sąsiadują z granicznymi lewymi, skrajne prawe komórki sąsiadują z skrajnymi lewymi, a skrajne górne ze skrajnymi dolnymi. Wszystkie zdarzenia zachodzą więc nie na płaszczyźnie, a na powierzchni torusa (czyli „obwarzanka”), która – tak jak i powierzchnia kuli – granic nie ma. Wróćmy do naszej zgrzebnej rzeczywistości.

```
Private Sub NEIB1(ByVal i As Integer)
  If Y(i) > 1 Then
    NN(1) = COOR(X(i), Y(i)-1)
  Else
    NN(1)= -1
  End If
  If X(i) < 10 Then
    NN(2) = COOR(X(i)+1, Y(i))
  Else
    NN(2)= -1
  End If
  If Y(i) < 10 Then
    NN(3) = COOR(X(i), Y(i)+1)
  Else
    NN(3)= -1
  End If
  If X(i) > 1 Then
    NN(4) = COOR(X(i)-1, Y(i))
  Else
    NN(4)= -1
  End If
```

```
End Sub
```

W powyższej procedurze indeksy sąsiadujących komórek przypisywane są zmiennym NN(1), NN(2), NN(3) i NN(4).

Następnym etapem w tworzeniu modelu jest opis stanów, w jakich może znajdować się komórka. Załóżmy, że komórkę może zajmować las (State(i) = 1), krzaki (State(i) = 2), łąka (State(i) = 3), ziemia orna (State(i) = 4) lub osiedle typu wiejskiego (State(i) = 5). Odwzorowanie każdej komórki na mapie może się odbywać za pomocą koloru (wygodne na monitorze) lub w postaci mnemonicznego rysunku (wygodne do wydruku). Poniżej podajemy procedurę MAP, posługującą się kolorami:

```
Private Sub MAP ()
  Dim i As Integer
  For i = 0 To n – 1
    Select Case State (i)
    Case 1
       cell(i).FillColor = RGB(0, 255, 0)
      Case 2
        cell(i).FillColor = RGB(125, 255, 125)
      Case 3
        cell(i).FillColor = RGB(255, 255, 0)
      Case 4
        cell(i).FillColor = RGB(0, 255, 255)
      Case 5
        cell(i).FillColor = RGB(100, 100, 100)
    End Select
   Next i
End Sub
```

Przypomnijmy, że FillColor steruje zalewaniem całego obiektu typu Shape jednym określonym kolorem, a zdefiniowanie koloru w postaci RGB powoduje zmieszanie trzech kolorów podstawowych: czerwonego (ang. *Red*), zielonego (ang. *Green*) i niebieskiego (ang. *Blue*). Więcej informacji o posługiwaniu się kolorami można znaleźć w drukowanej instrukcji *VB* lub po kliknięciu na przycisku pomocy [Help].

Teraz jest odpowiedni moment, żeby sprawdzić działanie odzwierciedlania stanu modelu na mapie. W tym celu utworzymy przycisk menu Test Map:

```
Private Sub mnuTEST_MAP_Click()
  Dim i As Integer
 For i = 0 To 99
    Select Case Rnd
    Case 0 To 0.2
       State(i)=1
```

```
        Case 0.2 To 0.4
            State(i)=2
        Case 0.4 To 0.6
            State(i)=3
        Case 0.6 To 0.8
          State(i)=4
        Case Else
          State(i)=5
       End Select
    Call MAP
End Sub
```

Powyższa procedura nadaje stan komórkom metodą stochastyczną. Jeżeli wszystko działa w normie, otrzymujemy podobny obraz jak na rys. 12.2 (postać graficzna, a nie kolorowa).

Zajmiemy się teraz utworzeniem stanu początkowego. Niech to będzie na przykład niewielki kawałek lasu (w komórce 15) otoczony łąkami. Stosowna procedura będzie wyglądać tak:

```
Private Sub mnuINITIAL_Click()
  Dim i As Integer
     For i = 0 To n – 1
       State(i) = 3
     Next i
   State(15) = 1
   Call MAP
End Sub
```

Stan początkowy naszego modelu przedstawia rys.12.3.

Rozpatrzmy bardzo prosty scenariusz rozprzestrzenienia się lasu na obszarze zajętym przez łąki. W tym celu – oprócz bloku stanów układu w danym momencie (`State(i)`) – utworzymy jeszcze jeden taki blok, odpowiadający następnemu momentowi czasu (`State1(i)`). Algorytm modelujący powinien przenieść układ ze stanu `State(i)` do stanu `State1(i)`. W tym celu wykonuje się skanowanie wszystkich komórek i określa się stan wszystkich sąsiadów każdej kolejnej komórki. Jeżeli daną komórkę zajmuje las, a sąsiednią pole, to w następnym momencie czasowym w tej ostatniej pojawi się las. To, zdawałoby się, proste zadanie komplikuje się, ponieważ las nie może rozprzestrzeniać się poza granice rozpatrywanej powierzchni. Poza tym jeżeli w danej komórce nie zachodziły żadne zmiany, znaczy to, że zachowała ona swój poprzedni stan

```
Private Sub mnuSCENARIO1_Click()
   Dim i As Integer
   Dim j As Integer
   For i = 0 To n -1
```

```
      State1(i) = 0
    Next i
    For i = 0 To n -1
       If State(i) = 1 Then
         Call NEIB1(i)
         For j = 1 To 4
         If NN(j) <> -1 Then
              State1(NN(j)) = 1
            End If
         Next j
       End If
     Next i
     For i = 0 To n – 1
       If State1(i) = 0 Then
          State1(i) = State(i)
       End If
       State(i) = State1(i)
     Next i
    Call MAP
 End Sub
```

Stan układu po siódmym kroku czasowym przedstawia rys. 12.4.

Skomplikujmy nieco algorytm modelujący przez wprowadzenie stochastyczności i dodanie do zespołów roślinnych leśnego i łąkowego trzeciego zespołu – krzakowego. Jeżeli daną komórkę zajmuje las, z prawdopodobieństwem `Prob4` po następnym kroku czasowym w komórce będzie nadal las, natomiast z prawdopodobieństwem `Prob1` komórka sąsiadująca z komórką zalesioną może zostać zajętą przez krzaki.

Jeżeli daną komórkę zajmują krzaki, to z prawdopodobieństwem `Prob3` mogą zostać zastąpione przez las. Z prawdopodobieństwem `Prob2` krzaki mogą rozprzestrzenić się na sąsiednie komórki. Jeżeli w danej komórce nic się nie dzieje, to trwa w niej stan poprzedni.

Jeżeli w stanie początkowym układ wygląda jak na rys. 12.2, to powyższy scenariusz po 20 krokach czasowych może doprowadzić do stanu jak na rys. 12.5. Zauważmy, że wobec stochastyczności algorytmu modelującego każda realizacja modelu będzie różnić się od poprzedniej. Rozpatrywany układ po procesie przejściowym, w którego wyniku zanikają wszystkie zespoły łąkowe, doznaje nieregularnych wahań: krzaki wchodzą na miejsce lasu albo odwrotnie, tworząc dziwaczną mozaikę.

```
 Private Sub mnu SCENARIO2_Click()
    Dim i As Integer
    Dim j As Integer
    For i = 0 To n -1
      State1(i) = 0
    Next i
```

```
    For i = 0 To n – 1
      If State(i) = 1 Then
        If Prob4 > Rnd Then
            State1(i) = 1
          End If
        Call NEIB1(i)
          For j = 1 To 4
            If NN(j) <> -1 Then
               If Prob1 > Rnd Then
                  State1(NN(j)) = 2
               End If
            End If
          Next j
        End If
        If State(i) = 2 Then
         If Prob3 > Rnd Then
             State1(i) = 1
           End If
      Call NEIB1(i)
      For j = 1 To 4
          If NN(j) <> -1 Then
               If Prob2 > Rnd Then
                State1(NN(j)) = 2
              End If
          End If
      Next j
      End If
    Next i
    For i = 0 To n – 1
      If State1(i) = 0 Then
      State1(i) = State(i)
     End If
     State(i) = State1(i)
    Next i
    Call MAP
  End Sub
```

Kontynuujemy komplikowanie naszego modelu. Wprowadźmy doń możliwość pojawienia się na modelowanym obszarze osiedla ludzkiego. Załóżmy, że osiedle może się pojawić tylko na obszernej polanie, i to w dodatku jedynie z pewnym prawdopodobieństwem – `ProbVillage`. Celem sformalizowania pojęcia „obszerna polana” wprowadzimy zmienną `Meadow`, w której będziemy sumować wszystkie łąki spotykane w okolicy danej komórki (`State(i) = 3`). Dopiero przybranie przez tę zmienną wartości 4, co oznacza „wokoło łąki”, pozwala na budowę pierwszego domu w wiosce. W pozostałych sprawach algorytm scenariusza trzeciego powtarza algorytm scenariusza drugiego. Jak już Czytelnicy z pewnością zauważyli, konsekwentnie, acz powoli, zwiększamy stopień komplikacji modelu. To bardzo ułatwia poszukiwanie ewentualnych błędów. Jeżeli Wasz program działał dobrze, a później dodaliście coś do tego programu i wtedy zaczęły się kłopoty, to z wielkim prawdopodobieństwem kłopoty te związane są z tą dodaną ostatnio częścią programu, tam więc należy szukać powodów owych kłopotów.

```
Private Sub SCENARIO3_Click()
   Dim i As Integer
   Dim j As Integer
   Dim Meadow As Integer
   For i = 0 To 99
     State1(i) = 0
   Next i
   For i = 0 To 99
      If State(i) = 5 Then
        State1(i) = 5
    End If
      If State(i) = 1 Then
         If Prob4 > Rnd Then
         State1(i) = 1
         End If
      Call NEIB1(i)
    For j = 1 To 4
      If NN(j) <> -1 Then
          If Prob1 > Rnd And State(NN(j)) <> 5 Then
            State1(NN(j)) = 2
          End If
        End If
    Next j
    End If
    If State(i) = 2 Then
       If Prob3 > Rnd Then
            State1(i) = 1
       End If
    Call NEIB1(i)
    For j = 1 To 4
       If NN(j) <> -1 Then
           If Prob2 > Rnd And State(NN(j)) <> 5 Then
             State1(NN(j)) = 2
           End If
         End If
    Next j
    End If
    If State(i) = 3 Then
      Meadow = 0
      Call NEIB1(i)
      For j = 1 To 4
```

```
          If NN(j) <> -1 Then
              If State(NN(j)) = 3 Then
             Meadow = Meadow + 1
           End If
       Else
          Meadow = Meadow – 5
       End If
    Next j
    If Meadow = 4 And ProbVillage > Rnd Then
       State1(i) = 5
    End If
   End If
  Next i
  For i = 0 To 99
    If State1(i) = 0 Then
       State1(i) = State(i)
    End If
    State(i) = State1(i)
  Next i
  Call MAP
 End Sub
```

Rys.12.6 przedstawia wyniki działania modelu według scenariusza trzeciego. Osady, zwłaszcza położone wśród łąk, zaczynają stopniowo otaczać się zaroślami krzaków i lasem. Nic w tym dziwnego, bo algorytm modelujący nie zawiera żadnych wskazówek dotyczących ludzkiej działalności gospodarczej, na przyklad usuwania krzaków wokół wsi.

W kolejnym scenariuszu wprowadzimy pewne prawdopodobieństwo tego, że w pobliżu osiedli ludzie zaczną usuwać zarośla i uprawiać ziemię (`ProbArableLand`). Wielkość tego prawdopodobieństwa jest proporcjonalna do aktywności rolniczej mieszkańców danej wioski. Zakłada się, że wyrębowi podlegają jedynie komórki zajęte przez krzaki i las. Pola muszą być, oczywiście, regularnie uprawiane, bo inaczej zarosną znów krzakami.

Wynik działania modelu według scenariusza czwartego przedstawia rys. 12.7. Modyfikowanie i uzupełnianie modelu można prowadzić dowolnie długo. Zatrzymamy się jednak na ostatnio omówionym scenariuszu i zajmiemy się ulepszaniem sposobów badania modelu.

Do oceny stanu całego układu w danym momencie użyteczne jest rozpoznanie, ile komórek znajduje się w określonych stanach. W tym celu utworzymy procedurę `LandSurface`, w której jednocześnie będzie obliczany współczynnik różnorodności Shannona (`Shannon(t)`).

```
Private Sub LandSurface(ByVal t As Integer)
Dim i As Integer
 Forest(t) = 0
```

```
    Bush(t) = 0
    Meadow(t) = 0
    Arable(t) = 0
    Village(t) = 0
    Dim p(5) As Single
    For i = 0 To 99
     Select Case State(i)
       Case 1
           Forest(t) = Forest(t) + 1
       Case 2
           Bush(t) = Bush(t) + 1
       Case 3
           Meadow(t) = Meadow(t) + 1
       Case 4
         Arable(t) = Arable(t) + 1
       Case 5
         Village(t) = Village(t) + 1
     End Select
    Next i
    p(1) = Forest(t) / 100
    p(2) = Bush(t) / 100
    p(3) = Meadow(t) / 100
    p(4) = Arable(t) / 100
    p(5) = Village(t) / 100
    Shannon(t) = 0
    For i = 1 To 5
      If p(i) > 0 Then
         Shannon(t) = Shannon(t) + p(i) * Log(p(i))
      End If
    Next i
    Shannon(t) = – Shannon(t)
  End Sub
```

Poza tym warto obejrzeć wykresy zmian w czasie liczby komórek zajętych przez las, krzaki, łąki, pola i wsie, jak również współczynnika różnorodności w całym układzie. W tym celu utworzymy pozycję menu GRAPHIC wraz z odpowiednią procedurą:

```
  Private Sub mnuGRAPHIC_Click()
    Dim t, i As Integer
    Dim scale_max as single
    scale_max = 100
    gra.Top = 0
    gra.Left = 0
    gra.Height = 5000
    gra.Width = 7000
    gra.ScaleTop = – 0.1 * scale_max
    gra.ScaleLeft = – 0.1 * tmax
```

```
    gra.ScaleHeight = 1.2 * scale_max
    gra.ScaleWidth = 1.2 * tmax
    gra.Visible = True
    gra.Cls
    gra.Line (0, 0) - (tmax, scale_max), , B
    gra.DrawWidth = 4
    gra.DrawWidth = 1
    Call CURVE(Forest(), RGB(0, 125, 0))
    Call CURVE(Bush(), RGB(125, 255, 125))
    Call CURVE(Meadow(), RGB(255, 255, 0))
    Call CURVE(Arable(), RGB(0, 0, 255))
    Call CURVE(Village(), RGB(0, 0, 0))
    gra.PSet (1, scale_max *(1 - 0.5 * Shannon(1))
    For t = 2 To tmax
      gra.Line  -  (t,  scale_max*(1  –  0.5  *  Shannon(t)),
    RGB(0, 0, 0)
    Next t
       gra.DrawWidth = 1
    For t = 0 To tmax Step 10
       gra.Line (t, tmax) - (t, 1.03*tmax)
       gra.CurrentX = t
       gra.CurrentY = 1/03 * tmax
       gra.Print t
    Next t
    For i = 0 To scale_max Step 10
       gra.Line (0, scale_max – i) - (-3, scale_max – i)
       gra.CurrentX = 0
       gra.CurrentY = scale_max – i
       gra.Print i
    Next i
  End Sub
```

Wygodne jest stworzenie osobnej procedury odpowiedzialnej za wykreślanie poszczególnych krzywych na wykresach; argumentami są nazwa bloku i kolor krzywej.

```
  Private Sub CURVE(ByRef ARR() As Integer, ByVal col As Long)
    Dim t As Integer
    gra.PSet (1, 100 – ARR(1))
    For t = 2 To 100
       gra.Line -(t, 100 – ARR(t)), col
    Next t
  End Sub
```

Radzamy Czytelnikom, by samodzielnie przeprowadzili eksperymenty na modelu przy różnych wielkościach prawdopodobieństw `Prob1`, `Prob2`, `Prob3`, `ProbVillage` i `ProbArableLand`. W tym celu można sporządzić paski przewijania (sposób ich tworzenia opisano w rozdziale 7). Później można zająć się udoskonalaniem modelu, co objawia szerokie możliwości. Liczbę stanów automatu komórkowego można powiększyć w sposób zasadniczy. Na przykład możemy wyróżnić klasy liściaste i iglaste lub nawet bardziej szczegółowe botaniczne zróżnicowania. Każda komórka może otrzymać informację o jej wysokości nad poziomem morza, jak również rzeźbę powierzchni gruntu w danej miejscowości, a algorytm modelujący można wzbogacić informacjami o nachyleniu zboczy i ekspozycji, co jest szczególnie istotne przy modelowaniu krajobrazów górskich. Istotnym uzupełnieniem będą także informacje geologiczne i gleboznawcze.

Do modelu można wprowadzić informacje o sieci hydrologicznej – w tym celu należy, na przykład, wskazać, przez który brzeg komórki wpływa ciek wodny, a przez który wypływa, jak też określić przepływ wody i wielkość spływu. To, oczywiście, skomplikuje model, lecz za to da możliwość odtworzenia w modelu takich zjawisk, jak: erozja gleby, spływ powierzchniowy, poziom wód gruntowych i jego wpływ na roślinność i rolnictwo. Niektóre komórki w modelu mogą reprezentować zbiorniki wodne (stawy, jeziora lub zbiorniki zaporowe), z ich specyficznymi i już dość dobrze poznanymi funkcjami przejść. Szczególnie ważne dla funkcjonowania całego układu mogą okazać się komórki rozmieszczone na granicy wody i lądu.

W końcu wprowadzenie do modelu krajobrazu sieci dróg i całej infrastruktury przemysłowej, rolniczej i innych dziedzin ludzkiej aktywności jeszcze bardziej zbliży model do rzeczywistości. Nie należy jednak się śpieszyć i usiłować uwzględnić w modelu od razu wszystko! Jeżeli nawet taki model będzie działał, to odróżnienie w nim rzeczywistości od błędów, jakie mogły się pojawić przy pisaniu algorytmu modelującego, będzie bardzo trudne.

*

Na zakończenie tego rozdziału podajemy kilka praktycznych porad dla tych, którzy rzeczywiście zamierzają wejść na kamienistą drogę programowania i modelowania imitacyjnego:

1. Wszystkie programy, które konstruujecie, wszystkie wyrażenia i terminy, których używacie, muszą być dla was absolutnie jednoznaczne. Pamiętajcie, że komputer, w odróżnieniu od człowieka, nie rozumie napomknień i niedomówień.
2. Tworząc model lub program, od samego początku myślcie o tym, jak będziecie później sprawdzać wyniki waszej twórczości. Im częściej będziecie przeprowadzać takie sprawdzania i im mniejsze będą sprawdzane fragmenty programu, tym większa będzie nadzieja, że cały model lub program będzie nadawał się do użytku.

3. Nigdy nie nabierajcie przekonania, że jakaś wielkość nie może być równa zeru. Zadbajcie o to, aby nie dopuścić do dzielenia przez zero. Komputery bardzo tego nie lubią.
4. Nie zapominajcie, że wasz program będzie czytany nie tylko przez komputer, ale i przez ludzi, a zwłaszcza przez Was samych. Dlatego piszcie Wasze programy w taki sposób, abyście za rok czy dwa mogli go zrozumieć, a nie mówić: „Co za idiota to napisał!".
5. Komentarze do programu to rzecz bardzo potrzebna i pożyteczna; ale lepiej jest nie dawać komentarzy w ogóle, niż umieścić komentarze nieprawdziwe.
6. Pisząc program, trzeba być bardzo skupionym i uważnym. Błędnie napisany, a nie dostrzeżony w porę nawet jeden symbol może doprowadzić do katastrofalnych skutków.
7. Starajcie się nie pisać długich modułów i procedur. Wśród programistów popularne jest powiedzenie, że długość modułów jest odwrotnie proporcjonalna do kwalifikacji piszącego program.
8. Nadając nazwy zmiennym, starajcie się, aby te nazwy kojarzyły się z istotą owych zmiennych; to bardzo ułatwia rozumienie programu.
9. Starajcie się nie mieszać w tym samym fragmencie programu liczb i symboli zmiennych, nawet wtedy gdy jesteście przekonani, że tych liczb nie będziecie nigdy zmieniać. Ulokujcie wszystkie liczby w jakimś jednym miejscu (na przykład w procedurze Form Load); ułatwi to późniejszą weryfikację programu.
10. Współczesne języki programowania oferują liczne możliwości przechodzenia między rodzajami danych, jak również pracy z użyciem nie zdefiniowanych uprzednio zmiennych i innych „śliczności". Starajcie się unikać tych śliczności; oszczędzicie sobie w ten sposób straty czasu i nerwów.
11. Traktujcie „z ograniczonym zaufaniem" to, co napisano w książkach, czasopismach, a nawet podręcznikach programowania. Nowo poznaną konstrukcję sprawdźcie w praktycznym zastosowaniu, czy działa ona tak właśnie, jak tego oczekujecie, a dopiero wtedy zastosujcie w waszym programie lub modelu. To samo dotyczy danych o obiektach przyrodniczych. To, co wydrukowano, może być w pełni prawdziwe dla określonych warunków, ale czy pasuje do waszego modelu?
12. Gdy podczas badania modelu otrzymacie wynik nieoczekiwany i interesujący, pomyślcie natychmiast, że jest to wynikiem błędu w programie lub w sformułowaniu celu. Dopiero po usunięciu wszelkich możliwych artefaktów pomyślcie nieśmiało, że dowie-

dzieliście się czegoś nowego o obiekcie modelowania. Zdarza się to, niestety, niezbyt często!

Pragnąc, z jednej strony, ułatwić Czytelnikom pisanie własnych programów z zastosowaniem automatów komórkowych, a z drugiej dać pogląd o całości tej procedury, proponujemy do obejrzenia całość naszego programu *KRAJOBRAZ* w takiej postaci, jaka została uruchomiona w *Visual Basicu* (zob. Dodatek B).

Literatura

Acock B., Reynold J. 1997. Introduction: modularity in plant models. „Ecological Modelling", 94, s. 1-6.

Alimov A. F., Umnov A. A. 1997. Исследования разнообразия потоков биоты в толще воды с использованием математической модели (Studies of biotic flow diversity in water bodies using a mathematical model). In: Alimov A. F., Bul'on V. V. [Eds.]. Реакция озерных экосистем на изменения биотических и абиотических условий (The response of lake ecosystems to changes in biotic and abiotic conditions.). Труды Зоологического Института Росийской Академии Наук. 272.

Andrzejewski R. 2001. Kształtowanie systemu ekologicznego gminy. W : W trosce o Ziemię. Księga ku czci Profesora Stefana Kozłowskiego. Lublin: Wydawnictwo KUL, s. 157-171.

Armand D. L. 1980. Nauka o krajobrazie. PWN, Warszawa, 287 ss.

Bakier W. L. 1989. A review of models of landscape changes. „Landscape Ecology", 2(2), s. 111-133.

Balzter H., Braun P. W., Kohler W. 1998. Cellular automata models for vegetation dynamics. „Ecological Modelling", 107, s. 113-125.

Bernadzki E. 1993. Obecne problemy planowania hodowlanego. „Prace IBL". Ser. B. 15, s. 134-140.

Bernadzki E. 1997. Cele hodowli lasu wczoraj i dziś. „Sylwan", 4, s. 23-31.

Bernadzki E., Bolibok L., Brzeziecki B., Zajączkowski J., Zybura H. 1998. Compositional dynamics of natural forests in the Białowieża National Park, northeastern Poland. „Journal of Vegetation Science", 9, s. 229-238.

Bertalanffy L. von 1975. Perspectives on General Systems Theory. Braziller, NY, 183 ss.

Boehmer H. J. 1997. Zur Problematik des Mosaik-Zyklus-Begriffes. „Natur und Landschaft", 72, s. 333-338.

Bojarski W. W. 1984. Podstawy analizy i inżynierii systemów. Warszawa: PWN.

Bonan G. B., van Cleve K., 1992. Soil temperature, nitrogen mineralization, and carbon source-sink relationships in boreal forests. „Canadian Journal of Forest Research", 22, s. 629-639.

Bormann F. H. Likens G. E. 1979. Pattern and process in forested ecosystem: Disturbance, development and state on the Hubburd Brook ecosystem study. New York: Springer, 253 ss.

Bosatta E. 1982. Acidification and release of nutrients from organic matter – A model analysis. „Oecologia", 55, 1, ss. 30-33.

Bossel H. 1985. Umweltdynamik - 30 Programme für kybernetische Umwelterfahrungen auf jedem BASIC-Rechner. Te-Wi Verlag, München, 466 ss.

Botkin D. B. 1993. Forest Dynamics: An Ecological Model. Oxford, New York: Oxford University Press.

Botkin D. B., Janak F. J., Wallis J. R. 1972. Some ecological consequences of computer model of forest growth. „Journal of Ecology", 60, s. 649-873.

Braun-Blanquet J. 1951. Pflanzensoziologie. 2. Aufl. Wien.

Bruchwald A. 1986. Simulation growth model MDI-1 for Scots pine „Annals" Warsaw Agricultural University 34, s. 47-52.

Bruchwald A. 1988. Przyrodnicze podstaw budowy modeli wzrostu. „Sylwan", 11/12, s. 1-10.

Bruchwald A. 1991. MIS growth model for pine. „Annals". Warsaw Agricultural University. Forest and Wood Technology, 41, s. 3-7.

Brzeziecki B. 1991. Ecological growth Model of the Forest: some methodical and calibration problems. „Sylwan", 9, s. 5-15.

Brzeziecki B. 1999. Ekologiczny model drzewostanu. Zasady konstrukcji, parametryzacja, przykłady zastosowań. Warszawa, 115 ss.

Brzeziecki B., Zybura H. 1998. Naturalne zmiany składu gatunkowego i struktury pierśnic drzewostanu na siedlisku olsu jesionowego w okresie 47 lat: sukcesja czy regeneracja? „Sylwan", 4, s. 19-31.

Bugala W. 1991. Drzewa i krzewy dla terenów zieleni. Wyd. 2. Warszawa: PWRiL.

Bugmann H. K. M. 1994. On the ecology of mountainous forests in a changing climate: a simulation study. Diss. ETH No. 10638. Zurich.

Childress W. H., Rykiel E. J., Kohler W. 1996. Transition rule complexity in grid-based automata model. „Landscape Ecology ", 11 (5), s. 257-266.

Colasanti R. L, Grime J. P. 1993. Resource dynamics and vegetation processes: a deterministic model used two-dimensional cellular automata. „Functional Ecology", 7, s. 169-176.

Čumačenko S. I. 1992. Базовая модель динамики многовидового разновозрастного лесного ценоза.„Труды МЛТИ", 248, s. 147-179.

Dengler A. 1992. Waldbau auf ökologischer Grundlage. Bd. 1. Der Wald als Vegetationsform und seine Bedeutung für den Menschen (bearbeitet von E. Roehrig und N. Bartsch). Hamburg und Berlin: Verlag Paul Parey.

Droesen W. J. 1996. Formalization of ecohydrological expert knoweledge applying fuzzy techniques. „Ecological Modelling", 85(1), s. 75-81.

Dylis N. V. 1978. Основы биогеоценологии. Москва: Наука, 152 ss.

Ellenberg H. 1986. Vegetation Mitteleuropas mit den Alpen in ökologischer Sicht. 4. Aufl. Stuttgart: Verlag Eugen Ulmer.

Faliński J. B., Herezniak J. M. 1977. Zielone grądy i czarne bory Białowieży. Warszawa: Nasza Księgarnia.

Faliński J. B. 1986. Vegetation dynamics in temperate lowland primeval forests. Ecological studies in Białowieża forest. „Geobotany", 8, s. 1-537.

Faliński J. B. 1988. Succession, regeneration and fluctuation in the Bialowieza Forest (NE Poland). „Vegetatio", 77, s. 115-128.

Forman R. T. T., Gordon M. 1986. Landscape Ecology. New York: Wilej.

Forrester J. 1974. Динамика развития города. Москва: Прогресс.

Frelich L. E., Lorimer C. G. 1991. A simulation of landscape – level stand dynamics in the Northern hardwood region. „Journal of Ecology", 79, s. 223-233.

Friend A. D., Shugart H. H., Running S. W. 1993. A physiology-based gap model of forest dynamics. „Ecology", 74, s. 792-797.

Gardner M. 1971. On cellular automata, self reproduction the garden of Eden and game „Life". „Scientific American", 224, s. 112-117.

Green D. G. 1989. Simulated effects of fire, dispersal and spatial pattern on competition within forest mosaics. „Vegetatio", 82, s. 139-153.

Greniewski H., Kempista M. 1963. Cybernetyka z lotu ptaka. Warszawa: PWN 142 ss.

Greniewski H. 1969. Cybernetyka niematymatyczna. Warszawa: PWN, 520 ss.

Greniewski H. 1969. Cybernetyka niematymatyczna. Wykresy. Warszawa: PWN 53 ss.

Grodzynśkyj M. D. 1993. Основи ландшафтної екології. Київ: Наука, 224 ss.

Guttorp P. 1995. Stochastic Modelling of Scientific Data. London: Chapman and Hall.

Halle F., Oldeman R. A. A., Tomlinson P. B. 1978. Tropical trees and forests: An architectural analysis. Heidelberg: Springer, 441 ss.

Hansen S., Jensen M. E., Neilser N., Svendsen H 1990. DAISY – Soil, Plant, Atmosphere. System Modelling 10. Copenhagen.

Hassel M, P, Comins H. N, May R. M. 1991. Spatial structure and chaos in insect population dynamics. „Nature", 353, s. 255-258.

Hogeweg P. 1988. Cellular automata as a paradigm for ecological modelling. „Applied Mathematics and Computation", 27, s. 81-106.

Horn H. S. 1975. Forest succession. „Scientific American", 232, s. 90-98.

Jaworski A. 1994. Hodowla lasu. Wymagania siedliskowe ważniejszych gatunków drzew leśnych oraz zasady ich odnawiania. Kraków: Akademia Rolnicza.

Jørgensen S. E. 1994. Fundamentals of ecological modelling. Amsterdam, London, New York, Tokyo: Elsevier, 628 ss.

Jørgensen S. E., Nielsen S. N., 1994. Models of the structural dynamics in lakes and reservoirs. „Ecological Modelling", 74, s. 39-46.

Karafyllidis I., Thanailakis A. 1997. A model for predicting forest fire spreading using cellular automata. „Ecological Modelling", 99, s. 87-97.

Karpiński J. J. 1949. Materiały do bioekologii Puszczy Białowieskiej. „Rozprawy i Sprawozdania IBL". Ser. A.

Kellomaeki S., Pukkala T. 1989. Forest landscape. A method of amenity evaluation based on computer simulation. „Landscape and Urban Planning", 18(2), s. 117-125.

Kienast F. 1987. FORECE – A forest succession model for southern central Europe. Oak Ridge National Laboratory, Oak Ridge , TN, ORNL/TM 10575, 69 ss.

Kienast F. 1993. Analysis of historic landscape patterns with a Geographical Information methodological outline. „Landscape Ecology", 8, 2.

Klekowski R. Z. Mienszutkin W. W. 1996. Modelowanie matematyczne procesów ekologicznych. Warszawa: PAN. Wydział II Nauk Biologicznych, 249 ss.

Klekowski R. Z., Menshutkin V. V. 2002. Modelowanie komputerowe w ekologii. Lublin: Towarzystwo Naukowe KUL.

Korzukhin T.-M., Wagner R. 1996. Process versus empirical models: which approach for forest ecosystems management? „Canadian Journal of Forest Research", 26, s. 879-887.

Kowalski M. 1992. Ecological succession in Polish Forests. „Folia Forestalica Polonica". Ser. A, 34, s. 5-18.

Kozak I. 1997. Биогеоценотический покров Северного склона Украинских Карпат. Докторская диссертация (Praca habilitacyjna), 75 ss.

Kozak I., Menshutkin V. 1999. Computer simulation of forests Ecosystem Dynamics. „Biology Bulletin", 26, 6, s. 586-592.

Kozak I., Menshutkin V. 2000. An investigation of forest succession in Bieszczady Mountains using a computer model. „Folia Forestalica Polonica", 42, s. 67-81

Kozak I., Menshutkin V. 2001a Investigation of spruce forest dynamics in Bieszczady Mountains using a computer modelling. „Ecology" (Bratislava), 20, 4, s. 371-378.

Kozak I., Menshutkin V. 2001b. Prediction of beech forest succession in Bieszczady Mountains using a computer model. „Journal of Forest Science", 47 (8), s. 333-339.

Kozak I., Menshutkin V., Jóźwina M., Potaczała G., 2003. Modelling of beech forest dynamics in the Bieszczady Mountains in response to climate change. Ecology. (Bratislava). 22. 2. 152-161.

Kozak I., Menshutkin V., Ferchmin M., Potaczała G, Jóźwina M.,Kozak O., Seńko Z. Badania sukcesji sosnowego lasu w Kampinoskim Parku Narodowym z wykorzystaniem modelu FORKOME. (Parki Narodowe i Rezerwaty Przyrody, w druku).

Kozak I., Menshutkin V., Parpan V., Shparyk Yu., Parpan T., Viter R. Kozak O., Seńko Z. (in press). Computer simulations of natural beech forest dynamic in Boberka river basin in Ukrainian Beskids.

Kozłowski S. 1997. W drodze do ekorozwóju. Warszawa: Wydawnictwo Naukowe PWN.

Kozłowski S. [red.] 1999. Ekorozwój gminy Jozefów. Lublin: Redakcja Wydawnictw KUL.

Kozłowski S. [red.] 1999. Ekorozwój gminy Strzyżewice. Lublin: Redakcja Wydawnictw KUL

Kraeuchi N. 1994. Modelling forest succession as influenced by a changing environment. Mitteilungen der Eidgenoessischen Forschungsanstalt fuer Wald. „Schnee und Landschaft", 69, s. 145-271.

Krapiwin W. F., Swirjeżew J. A., Tarko A. N. 1988. Математическое моделирование глобальных екологических процессов. Москва: Наука.

Leemans R., Prentice I. C. 1987. Description and simulation of tree-layer composition and size distribution in a primeval *Picea-Pinus* forest. „Vegetatio", 69, s. 147-156.

Leemans R., Prentice I. C. 1989. FORSKA, a general forests succession model. Uppsala: Institute of Ecological Botany, 70 ss.

Leibundgut H. 1991. Unsere Waldbaeume: Eigenschaften und Leben. 2., ueberarb. und erg. Aufl. Bern- Stuttgart: Haupt.

Lembach M. 1994. Expert system-model coupling with the framework of an ecological advisory system. „Ecological Modelling", 75-76, s. 589-600.

Lindner M., Sievaenen R., Pretsch H. 1997. Improving the simulation of stand structure in a forest gap model. „Forest Ecology and Management", 95, s. 183-195.

Liu J., Ashton P. S. 1995. Individual-based simulation models for forest succession and management. „Forest Ecology and Management", 73, s. 157-175.

Logofet D. O. 1999. Сукцесионная динамика растительности: класические концепции и современные модели. In: Экология России на рубеже XXI века (наземные экосистемы), Москва: Наука s. 70-98.

Łukasiewicz J. 1961. Z zagadnień logiki i filozofii. Pisma wybrane. Wyboru dokonał, wstępem i przypisami opatrzył J. Słupecki. Warszawa: PWN.

Makosa K., Dzierzbicki J., Gromadzki A., Kliczkowska A., Krzyżanowski A. 1994. Zasady kartowania siedlisk leśnych. Warszawa: Wyd. IBL.

Martin Ph. 1992. EXE: A climatically sensitive model to study climate change and CO_2 enrichment effects on forests. „Australian Journal of Botany", 40, s. 717-735.

Martinez-Ramos M., Alvares-Buylla E., Sarukhan J. 1989. Tree demography and gap dynamics in a tropical forest. „Ecology", 70, 3, s. 555-558.

Martinez-Ramos M., Alvares-Buylla E., Sarukhan J., Pinero D. 1988. Tree fall age determination and gap dynamics in a tropical forest. „Journal of Ecology", 76, 3, s. 700-716.

Мелихов А. Н. 1971. Ориентированные графы и конечные автоматы. Москва. Наука.

Menshutkin, V. [ed.] 1997. Невская губа – опыт моделирования. Санкт Петербург, 375 ss.

Menshutkin V., Kozak I. 1997. An Investigation of a mixed beech forest dynamics in Ukrainian Carpathians using a computer model. In: Perzanowski K., Augustyn M. [eds]: Selected ecological problems of Polish-Ukrainian Carpathians. Bieszczady. Poland August 18-21,1997, Warsaw -Ustrzyki Dolne. ICE PAS s. 23-29.

Menshutkin W., Klekowski R. Z. 2001a Эколого-экономическая модель развития региона, основанная на экспертных оценках (на примере горного района Бещад, Польша). Известия АН. „Серия биологическая", 4, s. 507-512.

Menshutkin W., Klekowski R. Z. 2001b. Optimal management of the dam reservoir ecological system. „Ecohydrology and Hydrobiology" Vol.1, N4, s. 435-440.

Mienszutkin W., Fischer Z. 1995. Model prognostyczny ekologicznych układów górnej partii Karkonoszy. In: Fischer Z. [red.], Problemy ekologiczne wysokogórskiej części Karkonoszy. Dziekanów Leśny, s. 345-363.

Mitchell K. J. 1975. Stand description and growth simulation from low-level stereo photos of tree crowns. „Journal of Forestry". 73, s. 12-16.

Molofsky J. 1994. Population dynamics and pattern formation in theoretical population. „Ecology", 75, s. 30-39.

Mołčanow A. 1973. Влияние леса на окружающую среду. Москва: Haya.

Mozgawa J. 1985. Metodologiczne podstawy wykorzystania fotointerpretacji do modelowania syntetycznego w drzewostanach. Warszawa: SGGW-AR. Rozprawy naukowe i monografie.

Mynarski S., Szumilak J., Baścik K., Koczyński W. 1989. Elementy teorii systemów i informacji. PWN, 92 ss.

Naveh Z. 1984. Conceptual and theoretical basis of landscape ecology as a human ecosystem science. In: Naveh Z., Lieberman A. S. [eds], Landscape ecology: theory and application. Berlin, Heidelberg, New York: Springer, s. 26-105.

Newell A. 1973. Production systems: model of control structures. W: Visual Information Processing. New York: Academic Press, s. 463-526.

Nikolov N. T., Fox D. G. 1994. A coupled carbon-water-energy-vegetation model to *assess* responses of temperate forest ecosystems to changes in climate and atmospheric C02. Part l. Model concept. „Environmental Pollution", 83, s. 251-262.

Oja T. 1985. Модели развития древостоя. Препринт. Таллин: AN ЭССР, 60 ss.

Oktaba W. 1976. Matematyka i podstawy statystyki matematycznej. Warszawa: PWN.

Olinger S. V., Aber J. D., Fedores C. A. 1998. Estimating regional productivity and water yield using an ecosystem model linked to GIS. „Landscape and Urban Planning", 13(5), s. 223-234.

O'Neill R. V., Milne B. T., Tumer M. G., Gardner R. H. 1988. Resource utilization scales and landscape pattern. „Landscape Ecology", 2,1.

Pacyniak C. 1992. Najstarsze drzewa w Polsce. Warszawa: Wyd. PTK „Kraj".

Pastor J., Post W. M. 1985. Development of a linked forest productivity-soil process model. U.S. Department of Energy, ORNL/TM-9519.

Pawłowski W. J. 1996. Computer simulation of growth of a spruce stand using the PICEAT model. „Ekologia Polska", 44, 3-4, s. 333-349.

Pianka E. R. 1975. Niche relation of desert lizards. „Ecology and Evolution of Communities" (Cambridge), s. 292-314.

Polański Z. 1981. Metodyka badań doswiadczalnych. Kraków: Politechnika Krakowska.

Popadjuk R. V., Czistjakowa A. A., Czumaczenko S. I. 1991. Восточноевропейские широколиственные леса Москва: Наука, 335 ss.

Porté A., Bartelink H. H. 2002. Modelling mixed forest growth: a review of models for forest management. „Ecological Modelling", 150 (1-2), s 141-188.

Prentice I. C., Helmisaari H. 1991. Silvics of north European trees: Compilation, comparisons and implications for forest succession modelling. „Forest Ecology and Management”, 42, s.79-93.

Puchalski T. 1972. Rębnie w gospodarstwie leśnym. Warszawa: PWRiL.

Rapport D. J. 1991. Myths in the foundations of economics and ecology. „Biological Journal of the Linnean Society”, 44 (3), s. 185-202.

Recnagel F., Petzoldt T., Jaenke O., Krusche F. 1994. Hybrid expert system DE-LAQUA – a toolkit for water quality control of lakes and reservoirs. „Ecological Modelling”, 71(1-3), s. 17-36.

Richling A., Solon J. 1994. Ekologia krajobrazu. Warszawa, 315 ss.

Ritchie J. B. 1989. An expert system for a range land simulation model. „Ecological Monographs”, 6, s. 91-105.

Running S. W. 1994. Testing FOREST-BGC ecosystem process simulation across a climatic gradient in Oregon. „Ecological Applications”, 4, s. 238-267.

Satoh K 1990. Single and multi armed spatial patterns in cellular automaton model for an ecosystem. „Journal of Physical Society of Japan”, 59, s. 4204-4207.

Schreiber K. F. 1990. The History of Lanscape Ecology in Europe W: Zonneveld I. S., Forman R. T. T. [red.]. Changing Landscapes: An Ecological Perspective. New York: Springer.

Sequeire R. A., Olson R. L., McKinon J. M. 1997. Implementic generic, object-oriented models in biology. „Ecological Modelling”, 94, s. 17-31.

Shugart H. H., 1984. Theory of forests Dynamics. New York: Springer, 278 ss.

Shugart H. H., McLaughlin S. B., West D. C. 1980. Forest models: their development and potential applications for air pollution effects research. Proceedings of Symposium on: Effects of air pollutants on Mediterranean and temperate forest ecosystems. June 22-27, Riverside, California.

Shugart H. H., Smith T. M. 1996. A review of forest patch models and their application to global change research. „Climatic Change”, 34, s. 131-153.

Smith T. M., Urban D. L. 1988. Scale and resolution of forest structure pattern. „Vegetatio”, 74, 2-3, s. 143-150.

Sočawa V. B. 1978. Введение в учение о геосистемах. Новосибирск, 319 ss.

Solomon D. S. 1974. A growth model of natural and silviculturally treated stands of even-aged northern hard woods. U.S.D.A. Forest Servise Tech. Report 36, 30 ss.

Sprugel D. G. 1991. Disturbance, equilibrium, and environmental variability: what is “natural” vegetation in a changing environment? „Biological Conservation”, 58, s. 1-18.

Starfield A. M., Bleloch A. L. 1983. Expert systems. An approach to problems in ecological management that are difficult to quantify. „Journal of Environmental Management”, 16 (3), s. 261-268.

Sterbacek Z., Pomije J., Skopek V., Vokoun J. 1990. A composite landscape ecology prognostic expert system-COLEPS. Part. 11. Application of the heuristic model

HLEPES to prognoses of forest structures more resistant to emissions and climate changes. „Ecological Modelling", 52(3-4), s. 225-237.

Sterbacek Z., Skopek V., Zavazal V. 1990. A composite landscape ecology prognostic expert system-COLEPS. Part. 1. „Ecological Modelling", 50(1-3), s. 145-156.

Straškraba M. 1994. Ecotechnological models for reservoir water quality management. „Ecological Modelling", 74(1-2), s. 1-38.

Straškraba M., Gnauck A. 1985. Freshwater ecosystems; modelling and simulation. (Developments in environmental modelling, 8.). Amsterdam: Elsevier, 310 ss.

Suhanow W. W., Pietropawlowskij B. S., Czawtur N. A. 1994. Структура растительных сообществ Сихоте-Алинского заповедника. Владивосток, 219 ss.

Sukačov N. 1964. Основные понятия лесной биогеоценологии. Основы лесной биогеоценологии (Fundamentals of forest biogeocoenology). Москва: Наука, 545 ss.

Sullivan A. D., Clutter J. L. 1972. A simultaneous growth and yield model for loblolly pine. „Forest Science", 18, s. 76-86.

Szwagrzyk J. 1988. Struktura i dynamika lasu: teoria, metody badania, kontrowersje. „Wiadomości Ekologiczne", 34, s. 355-373.

Szwagrzyk J. 1994. Symulacyjne modele dynamiki lasu oparte na koncepcji odnawiania drzewostanu w lukach. „Wiadomości Ekologiczne", 40, s. 57-75.

Szymański J. M. 1991. Życie systemów. PWN, 238 ss.

Tenhunen J. D., Lenz R, Hantschel R. 2001. Ecosystem Approaches to Landscape Management in Central Europe. Ecological Studies, 147. Springer, 636 ss.

Terano T., Asaji K., Sugenko M. [red.] 1993. Прикладные нечеткие системы. Москва: Мир.

Timlin D. J, Pachepsky Y. A. 1997. A modular soil and root simulator. „Ecological Modelling", 94, s. 67-80.

Tomanek J. 1994. Botanika leśna. Wyd. 5. Warszawa: PWRiL.

Tuma A., Haasis H.-D., Rentz O. 1996. A comparison of fuzzy expert systems, neural networks and neuro-fuzzy approaches. Controlling energy and water flows. „Ecological Modelling", 85(1), s. 93-98.

Turkow W. G. 1985. Пространственно-временная структура ценопопуляции едификаторных климаксовых пихтово-еловых лесов Среднего Урала. In.: Структура и динамика биогеоценозов Урала. Свердловск, s. 3-11.

Ulrich B. 1992. Forest ecosystem theory based on material balance. „Ecological Modelling", 63, s. 163-183.

Van Der Ploeg S. W. F., Braat L. C., Van Lieporg W. F. J. 1987. Integration of resource economics and ecology. „Ecological Modelling", 38(1-2), s. 171-190.

Voit E. O., Sands J. 1996. Modelling forest growth. I. Canonical approach. „Ecological Modelling", 86, s. 51-71.

Von Neuman J. 1966. Theory of Self-Reproduction Automata. Urbania, Ill.

Waggoner P. E., Stephens G. R. 1970. Transition probabilities for a forest. „Nature” 225, s. 1160-1161.

Watt A. S. 1947. Pattern and process in the plant community. „Journal of Ecology”, 35, s. 1-22.

Wierzbicki A. 1999. System pomiarowy ZMŚP na terenie Stacji Bazowej „Pożary” w Kampinoskim Parku Narodowym. W: Zintegrowany monitoring środowiska przyrodniczego. Stacja bazowa „Pożary” w Kampinoskim Parku Narodowym. Warszawa, s. 85-121.

Wilkie D., S., Finn J. T. 1988. A spatial model of land use and forest regeneration in Ituri forest of northeastern Zaire. „Ecological Modelling”, 41, s. 307-323.

Williams M. 1996. A three-dimensional model of forest development and competition. „Ecological Modelling”, 89, s. 73-98.

Zadeh L. A. 1975. Fuzzy logies and Approximate Reasoning „Synthese”, 30, s. 407-428.

Zadeh L. A. 1999. Fuzzy sets as a basis for a theory of possibility „Fuzzy sets and systems”, 100, s. 9-34.

Zarzycki K. 1963. Lasy Bieszczadów Zachodnich. „Acta Agraria et Silvatica”. Seria leśna, 3, s. 3-132.

Zięba S. 1998. Natura i człowiek w ekologii humanistycznej. Lublin, 288 ss.

Zięba S. 2002. Ekosystem leśny wartością człowieka. Lublin, 268 ss.

Zonneveld J. I. S. 1984. A collection of graphical models used in landscape ecology. Utrecht.

RYSUNKI

Dodatek B

Instrukcja uruchamiania programów na CD

CELAUTO

Model jest deterministyczny; każde uruchomienie daje identyczne wyniki!

Radzimy, aby zapoznając się z tym programem czytelnik trzymał ksiażkę otwartą na rozdziale 4.2. (Modelowanie krajobrazów za pomocą aparatu automatów komórkowych).

Model uruchamiamy klikając dwa razy nazwę CELAUTO – otwiera się okno główne z menu i dwoma oknami. W menu u góry klikamy [STATES] – otwiera się plansza informacyjna, ukazująca 35 komórek, z których każda składa sie z czterech części, zajętych przez jeden z czterech rodzajów pokrycia (przypomnijmy: F – las, A – łąki lub pola uprawne, V – osady typu wiejskiego, T – osady typu miejskiego). Usuwamy planszę kliknięciem w jej polu.

Klikamy [INITIAL STATE] – otwiera się mapa Bieszczad z 1996 r. z podziałem na komórki i z liczbami ich „ćwiartek” o różnym pokryciu.

Klikamy [SCENARIO]. Mamy do dyspozycji cztery scenariusze: zalesienie, wzrost upraw polowych, urbanizacja i deurbanizacja (spadek urbanizacji). Klikamy, na przykład, [FORESTATION]. Ukazuje się matryca przejść; zawiera ona – skompilowane na podstawie opinii ekspertów – zmiany pokrycia komórek, spowodowane przez zmiany demograficzne i polityki gospodarczej. Na przykład komórki FFAA (połowa lasu i połowa pola) zmieniają się na FFFA (tylko ćwiartka pola, reszta – las). Kliknięcie usuwa macierz.

Klikamy [RUN MODEL]. W lewym oknie ukazuje się mapa stanu pokrycia Bieszczad po 100 latach „zalesiania”, a w prawym – przebieg zmian w czasie. Osiedla typu wiejskiego i miejskiego nie uległy istotnym zmianom, powierzchnia pól zmalała, a lasów wzrosła. U góry wykresu jest informacja, że dotyczy to przejścia od stanu wyjściowego do zalesiania. U dołu wykresu klikamy [SAVE GRAPHIC]; wykres przetrwa w czasie dalszych działań.

Klikamy [SCENARIO], a następnie na przykład [URBANISATION]. Ukazuje się odpowiednia matryca przejść. Po jej usunięciu klikamy [RUN MODEL]. Na mapie widać wyraźny wzrost pokrycia V i T (zwróćmy uwagę na nie podlegający zmianom obszar Parku Narodowego!). Na wykresie w prawym oknie napis u góry informuje, że jest to pokaz zmian w okresie od 100 do 200 roku, a więc urbanizacji po poprzednich 100 latach zalesiania. Klikamy [SAVE GRAPHIC2].

Znów klikamy [SCENARIO] i na przykład [AGRICULTURE]. Mapa ukazuje wzrost liczby żółtych kwadracików, a na wykresie z prawej widać wzrost upraw, spadek zalesienia, bez zmian natomiast ilość osiedli wiejskich i miejskich; napis przypomina, że jest to przejście od polityki urbanizacji do forowania upraw rolnych (w okresie od 200 do 300 lat). U dołu z lewej dla przypomnienia nadal widać wykres dla poprzedniego etapu: zalesianie → urbanizacja. Możemy kliknąć [SCENARIO] i wybrać następne przejście lub kliknąć [INITIAL STATE] i powrócić do stanu wyjściowego. Możemy wybrać kolejne przejście od stanu wyjściowego lub kliknąć [END] i wyjść z programu.

TERRECO

Kliknij nazwę TERRECO. Otwiera się okno ustawiania warunków początkowych. Dwanaście pasków przewijania ma ustawione warunki „domyślne", które można zaakceptować klikając w menu [RUN MODEL]. Bardziej prawidłowe jest dobranie jedenastu parametrów, a na dwunastym pasku – pauzy między krokami programu (co 10 lat). Pauza ok. 5-10 sek. pozwala odczytać wskazania.

Po zakończeniu pracy modelu zostaje wyświetlony stan końcowy: T – biomasa drzew, B – biomasa krzaków, G – biomasa traw, O – materia organiczna, N – azot w glebie. Prawe okno prezentuje wykres zmian wymienionych wyżej zmiennych w czasie. Kliknięcie w obrębie obu okien usuwa ich zawartość i pozwala na nowe ustawienie zmiennych i uruchomienie modelu. Kliknięcie [END] zamyka program.

FORKOME

Prosimy Czytelników o zaglądanie do tekstu rozdziału 3: „Model FORKOME jako przykład modelu płatowego". Główną cechą tego modelu jest to, że obiektami są poszczególne osobniki drzew i ich losy od pojawienia się do śmierci. Model ten jest zawsze w postaci stochastycznej i dlatego do jego badania szczególnie przydatna jest metoda Monte-Carlo.

Klikamy dwa razy nazwę FORKOME. Otwiera się okno, w którego menu klikamy [SCENARIO]. Dalej klikając [Temperature], mamy do wyboru osiem scenariuszy zmiany temperatury powietrza; klikając [Precipitation], mamy do wyboru osiem scenariuszy zmiany opadów atmosferycznych; klikając [Animal press], mamy do wyboru osiem scenariuszy zmiany wpływu zwierzyny na zbiorowisko leśne; klikając [Felling], można wybrać odpowiedni scenariusz gospodarki leśnej w postaci selektywnego wycinania różnych gatunków drzew w odpowiednim wieku.

W tym programie też możemy ustawić parametry dla każdego z 14 gatunków drzew. Usuwamy obraz i ponownie klikamy [PARAMETERS], a następnie nazwę wybranego gatunku drzewa, na przykład [Abies alba]. Otwierają się paski przewijania z cechami tego gatunku; wartości te domyślnie są ustawione tak, że odpowiadają temu gatunkowi drzew w Puszczy Kampinoskiej; wielkości te możemy zmienić. Powtarzamy powyższe czynności dla 13 pozostałych gatunków.

Przechodzimy w menu do [INITIAL STATE]. Klikając na [STANDART] mamy do wyboru jedną konkretną powierzchnią doświadczalną: las sosnowy z Puszczy Kampinoskiej wielkości 1/12 ha. Możemy również tworzyć swoją własną powierzchnię ze swoimi drzewami. Klikamy [RUN_1] i model rusza na jeden rok do przodu. Celem symulacji na jeszcze jeden rok musimy ponownie nacisnąć [RUN_1] – i tak dalej. Wracamy do standardowej pozycji. Klikamy [RUN_600] i model rusza od razu do 600 lat. Na dole możemy obejrzeć zmianę biomasy w ciągu tej symulacji. Klikając w menu [GRAPHICS], a w submenu pozycję [GRAPHIC1], możemy obejrzeć zmiany w czasie liczby i biomasy drzew w procentach. Jeżeli w submenu klikniemy pozycję [GRAPHIC2], możemy obejrzeć kolorowe wykresy zmiany biomasy i liczebności drzew w ciągu symulacji do 600 lat. Gdy natomiast klikniemy pozycję submenu [GRAPHIC3], możemy obejrzeć zmiany średniego wieku zbiorowiska leśnego.

Możemy zastosować metodę Monte-Carlo, klikając odpowiednią pozycję w menu, z 20 realizacji. Model przedstawia wyniki w postaci kolorowej tabeli i kolorowego wykresu zmian biomasy w t/ha i w procentach.

Naciskając na końcu menu pozycję [DIAG], mamy możliwość obejrzenia samego algorytmu modelu.

DIAHEIGHT

Program omówiono w rozdziale 7. Oblicza on, na podstawie wzoru, wysokość drzewa (bez uwzględniania różnic gatunkowych) w zależności od jego pierśnicy (średnicy V na wysokości 130 cm).

Kliknąć nazwę DIAHEIGHT. Po otwarciu okna kliknąć przycisk [START]. Na pasku przewijania z nazwą DIAMETER ustawić żądaną średnicę(cm), kliknąć przycisk [START] i odczytać wysokość (dcm). Celem przeprowadzenia nowego obliczenia ponownie kliknąć [START] i dalej postępować jak wyżej.

TREEPAR

Program omówiono w rozdziale 8. Różni się on od poprzedniego tym, że zawiera parametry (B2, B3) charakterystyczne dla poszczególnych gatunków drzew.

Kliknąć nazwę TREEPAR. Po otwarciu okna kliknąć przycisk przy nazwie wybranego gatunku drzewa (w okienkach ustawiły się parametry B2, B3 charakterystyczne dla tego gaunku). Na pasku przewijania ustawiamy średnicę. Natomiast wysokość wyliczana jest automatycznie. Możemy kliknąć przycisk przy nazwie innego gatunku drzewa i postępować tak, jak opisano wyzej, albo przycisk [STOP] i zamknąć program.

FOREST

Program, omówiony w rozdziale 9, pokazuje sposoby prezentacji graficznej.

Kliknąć nazwę FOREST. Po otwarciu okna kliknąć [Initial State]. Kliknąć [Run] i powtarzając tę czynność obserwować zmiany na rysunku.

STOCHASTIC

O tym programie była mowa w rozdziale 10. Program pokazuje stochastyczne rozkłady – równomierny (proporcjonalny) i normalny – na przykładzie wzrostu źdźbeł trawy.

Kliknąć nazwę STOCHASTIC. Po otwarciu okna kliknięcie na [BEGIN] ustawia domyślnie na 1000 źdźbeł trawy. Po każdym nowym wyborze typu rozkładu i liczebności klikamy [BEGIN].

GRASS

Program omówiono w rozdziale 11.

Kliknąć nazwę GRASS. Po otwarciu okna kliknąć w menu pozycję [Initial State], a następnie [Run] i powtarzając klikanie obserwować, jak trawa pojawia się, rośnie i ginie.

LANDSCAPE

Program, przedstawiony w rozdziale 12, demonstruje działanie (stochastyczne) prostych automatów komórkowych.

Kliknąć nazwę LANDSCAPE. Kiedy otworzy się okno programu, kliknąć w menu pozycję [TEST MAP]. Pokazuje ona typy pokrycia (las, pole uprawne itd. – zob. rozdział 12).

Kliknąć w menu pozycję [Initial State] – powiedzmy, że jest to „lasek pośrodku łąki". Klikamy jedną z czterech pozycji [SCENARIO], proponujących różne sekwencje zmian. Powtarzamy klikanie wybranej pozycji [SCENARIO], powodując zmiany. [SCENARIO 4-100 steps] powtarza zmiany według scenariusza czwartego automatycznie w ciągu 100 kroków (lat). Pozycja [GRAPHIC] ilustruje przebieg w czasie tego ostatniego scenariusza oraz wskaźnik różnorodności Shannona.

Dodatek B

Pełny wydruk programu KRAJOBRAZ

```
Const n = 100
Const tmax = 100
Dim State(n) As Integer
Dim State1(n) As Integer
Dim X(n) As Integer
Dim Y(n) As Integer
Dim xn1, yn1 As Integer
Dim xn2, yn2 As Integer
Dim xn3, yn3 As Integer
Dim xn4, yn4 As Integer
Dim N1, N2, N3, N4 As Integer
Dim NN(4) As Integer
Dim Prob1, Prob2, Prob3, Prob4 As Single

Dim ProbVillage As Single
Dim ProbArableLand As Single

Dim Forest(tmax) As Integer
Dim Bush(tmax) As Integer
Dim Meadow(tmax) As Integer
Dim Arable(tmax) As Integer
Dim Village(tmax) As Integer
Dim Shannon(tmax) As Single
Dim t As Integer

Private Sub Form_Load()
Dim i As Integer
Dim j As Integer

Prob1 = 0.1
Prob2 = 0.2
Prob3 = 0.2
Prob4 = 0.9
ProbVillage = 0.01
ProbArableLand = 0.5
'------------------------------------------------------------------

 hscPROB1.Value = 100 * Prob1

 lblPROB1.Caption = Format(Prob1, "#.##")
 hscPROB2.Value = 100 * Prob2
```

```
 lblPROB2.Caption = Format(Prob2, "#.##")
 hscPROB3.Value = 100 * Prob3
 lblPROB3.Caption = Format(Prob3, "#.##")
 hscPROB4.Value = 100 * Prob4
 lblPROB4.Caption = Format(Prob4, "#.##")
 hscPROB_VILLAGE.Value = 100 * ProbVillage
 lblPROB_VILLAGE.Caption = Format(ProbVillage, "#.##")
 hscPROB_ARABLE_LAND.Value = 100 * ProbArableLand
 lblPROB_ARABLE_LAND.Caption = Format(ProbArableLand, "#.##")
'------------------------------

For i = 0 To n
  X(i) = (i Mod 10) + 1
Next i
For i = 0 To n
  Y(i) = Int((i) / (10)) + 1
Next i
'----------------------------------
For i = 0 To 99
  cell(i).BackStyle = 1
  cell(i).BackColor = RGB(160, 255, 160)
  cell(i).FillStyle = 1
Next i
End Sub

Private Sub gra_Click()
  gra.Visible = False
End Sub

Private Sub hscPROB_ARABLE_LAND_Change()

   ProbArableLand = 0.01 * hscPROB_ARABLE_LAND.Value
   lblPROB_ARABLE_LAND.Caption = Format(ProbArableLand, "#.##")
End Sub

Private Sub hscPROB_VILLAGE_Change()

   ProbVillage = 0.01 * hscPROB_VILLAGE.Value
  lblPROB_VILLAGE.Caption = Format(ProbVillage, "#.##")
End Sub

Private Sub hscPROB1_Change()

  Prob1 = 0.01 * hscPROB1.Value
   lblPROB1.Caption = Format(Prob1, "#.##")
End Sub

Private Sub hscPROB2_Change()

   Prob2 = 0.01 * hscPROB2.Value
   lblPROB2.Caption = Format(Prob2, "#.##")
End Sub
```

```
Private Sub hscPROB3_Change()

   Prob3 = 0.01 * hscPROB3.Value
   lblPROB3.Caption = Format(Prob3, "#.##")
End Sub

Private Sub hscPROB4_Change()

   Prob4 = 0.01 * hscPROB4.Value
   lblPROB4.Caption = Format(Prob4, "#.##")
End Sub

Private Sub mnuBegin_Click()

Dim i As Integer
Dim r As Single
Cls

 For i = 0 To 99
   r = Rnd
   Select Case r
     Case 0 To 0.2
         State(i) = 1
     Case 0.2 To 0.4
         State(i) = 2
     Case 0.4 To 0.6
         State(i) = 3
     Case 0.6 To 0.8
         State(i) = 4
     Case Else
         State(i) = 5
    End Select
 Next i
 Call MAP
 Call MAP

End Sub

Private Sub mnuEnd_Click()

   End
End Sub

Private Sub mnuGRAPHIC_Click()
Dim t, i As Integer
gra.Top = 0
gra.Left = 0
gra.Height = 5000
gra.Width = 7000
gra.ScaleTop = -10
gra.ScaleLeft = -10
gra.ScaleHeight = 140
gra.ScaleWidth = 120
```

```
gra.Visible = True
gra.Cls
gra.Line (0, 0)-(100, 100), , B

gra.DrawWidth = 4

Call CURVE(Forest(), RGB(0, 125, 0))
Call CURVE(Bush(), RGB(125, 255, 125))
Call CURVE(Meadow(), RGB(255, 255, 0))
Call CURVE(Arable(), RGB(0, 0, 255))
Call CURVE(Village(), RGB(0, 0, 0))
gra.DrawWidth = 1
gra.PSet (1, 100 – 50 * Shannon(1))

For t = 2 To 100
  gra.Line -(t, 100 – 50 * Shannon(t)), RGB(0, 0, 0)
Next t
gra.DrawWidth = 1

For t = 0 To 100 Step 10
   gra.Line (t, 100)-(t, 103)
   gra.CurrentX = t
   gra.CurrentY = 103
   gra.Print t



Next t

For i = 0 To 100 Step 10
    gra.Line (0, 100 – i)-(-3, 100 – i)
    gra.CurrentX = -9
    gra.CurrentY = 100 – i
    gra.Print i
Next i
For i = 0 To 100 Step 10
    gra.Line (100, 100 – i)-(103, 100 – i)
    gra.CurrentX = 104
    gra.CurrentY = 100 – i
    gra.Print Format(i / 50, "#.#")
Next i
    gra.DrawWidth = 4
    gra.Line (0, 120)-(10, 120), RGB(0, 125, 0)
    gra.Line (20, 120)-(30, 120), RGB(125, 255, 125)
    gra.Line (40, 120)-(50, 120), RGB(255, 255, 0)
    gra.Line (60, 120)-(70, 120), RGB(0, 0, 255)
    gra.Line (80, 120)-(90, 120), RGB(0, 0, 0)
    gra.DrawWidth = 1
    gra.Line (0, 110)-(10, 110)
    gra.CurrentX = 12
    gra.CurrentY = 108
    gra.Print "Shannon Index"
    gra.CurrentX = 12
```

```
    gra.CurrentY = 118
    gra.Print "F"
    gra.CurrentX = 32
    gra.CurrentY = 118
    gra.Print "B"
    gra.CurrentX = 52
    gra.CurrentY = 118
    gra.Print "M"
    gra.CurrentX = 72
    gra.CurrentY = 118
    gra.Print "A"
    gra.CurrentX = 92
    gra.CurrentY = 118
    gra.Print "V"
End Sub

Private Sub mnuINITIAL_Click()

Dim i As Integer
t = 1
For i = 0 To 99
   State(i) = 3
Next i
State(15) = 1
Call MAP
Call MAP
End Sub

Private Sub MAP()
Dim i As Integer
Cls
For i = 0 To 99
  Select Case State(i)
    Case 1
      cell(i).BackColor = RGB(0, 255, 0)
      Call PIC_FOREST(i)
    Case 2
      cell(i).BackColor = RGB(180, 255, 180)
      Call PIC_BUSH(i)
    Case 3
      cell(i).BackColor = RGB(255, 255, 128)
      Call PIC_MEADOW(i)
    Case 4
      cell(i).BackColor = RGB(250, 160, 130)
    Case 5
      cell(i).BackColor = RGB(255, 255, 255)
      Call PIC_VILLAGE(i)
  End Select
Next i
End Sub

Private Function COOR(ByVal xx As Integer, ByVal yy As Integer) As
Integer
```

```
Dim x1, y1, C As Integer
For C = 0 To 99
   If X(C) = xx And Y(C) = yy Then

     COOR = C
       Exit Function
   End If
Next C
COOR = -1
End Function

Private Sub NEIB(ByVal i As Integer)

xn1 = X(i)
yn1 = Y(i) – 1

xn2 = X(i) + 1
yn2 = Y(i)

xn3 = X(i)
yn3 = Y(i) + 1

xn4 = X(i) – 1
yn4 = Y(i)

N1 = COOR(xn1, yn1)
N2 = COOR(xn2, yn2)
N3 = COOR(xn3, yn3)
N4 = COOR(xn4, yn4)

End Sub

Private Sub mnuRUN2_Click()

Dim i As Integer
Dim j As Integer

For i = 0 To 99
    State1(i) = 0
Next i

For i = 0 To 99
  If State(i) = 1 Then

   Call NEIB1(i)

   For j = 1 To 4
      If NN(j) <> -1 Then
         State1(NN(j)) = 1
      End If
   Next j
   End If
Next i
```

```
For i = 0 To 99
   If State1(i) = 0 Then

      State1(i) = State(i)

   End If
   State(i) = State1(i)
Next i

Call MAP
Call MAP

End Sub

Private Sub NEIB1(ByVal i As Integer)
xn1 = X(i)

yn1 = Y(i) – 1



xn2 = X(i) + 1
yn2 = Y(i)

xn3 = X(i)
yn3 = Y(i) + 1

xn4 = X(i) – 1
yn4 = Y(i)

NN(1) = COOR(xn1, yn1)
NN(2) = COOR(xn2, yn2)
NN(3) = COOR(xn3, yn3)
NN(4) = COOR(xn4, yn4)

End Sub

Private Sub mnuRUN3_Click()

Dim i As Integer
Dim j As Integer

For i = 0 To 99
    State1(i) = 0
Next i

For i = 0 To 99
  If State(i) = 1 Then
  If Prob4 > Rnd Then
     State1(i) = 1
  End If
   Call NEIB1(i)
```

```
   For j = 1 To 4
      If NN(j) <> -1 Then
        If Prob1 > Rnd Then
           State1(NN(j)) = 2
        End If
      End If
   Next j
   End If
'-------------------------------------------------------------------
   If State(i) = 2 Then
   If Prob3 > Rnd Then
      State1(i) = 1
   End If
   Call NEIB1(i)
   For j = 1 To 4
      If NN(j) <> -1 Then
        If Prob2 > Rnd Then
           State1(NN(j)) = 2
        End If
      End If
   Next j
   End If
Next i

For i = 0 To 99
   If State1(i) = 0 Then
      State1(i) = State(i)
   End If
   State(i) = State1(i)
Next i

Call MAP
Call MAP

End Sub

Private Sub mnuRUN4_Click()

Dim i As Integer
Dim j As Integer
Dim Meadow As Integer

For i = 0 To 99
    State1(i) = 0
Next i

For i = 0 To 99
   If State(i) = 5 Then
      State1(i) = 5
   End If
'-------------------------------------------------------------------
  If State(i) = 1 Then
  If Prob4 > Rnd Then
```

```
      State1(i) = 1
    End If
     Call NEIB1(i)
     For j = 1 To 4
        If NN(j) <> -1 Then
          If Prob1 > Rnd And State(NN(j)) <> 5 Then
             State1(NN(j)) = 2
          End If
        End If
     Next j
     End If
'------------------------------------------------------------------
     If State(i) = 2 Then
     If Prob3 > Rnd Then
        State1(i) = 1
     End If
     Call NEIB1(i)
     For j = 1 To 4
        If NN(j) <> -1 Then
          If Prob2 > Rnd And State(NN(j)) <> 5 Then
             State1(NN(j)) = 2
          End If
        End If
     Next j
     End If
'------------------------------------------------------------------
     If State(i) = 3 Then
     Meadow = 0
     Call NEIB1(i)
     For j = 1 To 4
        If NN(j) <> -1 Then
            If State(NN(j)) = 3 Then
              Meadow = Meadow + 1
            End If
        Else
          Meadow = Meadow – 5
        End If
     Next j
     If Meadow = 4 And ProbVillage > Rnd Then
        State1(i) = 5
     End If
     End If

Next i

For i = 0 To 99
   If State1(i) = 0 Then
      State1(i) = State(i)
   End If
   State(i) = State1(i)
Next i

Call MAP
```

```
Call MAP

End Sub

Private Sub mnuRUN5_100_Click()

Dim tt As Integer
For tt = 1 To 100
   lblTIME.Caption = Format(tt, "###")
   Call mnuRUN5_Click
   Call LandSurface(tt)
Next tt
End Sub

Private Sub mnuRUN5_Click()

Dim i As Integer
Dim j As Integer
Dim Meadow As Integer
t = t + 1
 lblTIME.Caption = Format(t, "###")
For i = 0 To 99
    State1(i) = 0
Next i

For i = 0 To 99
   If State(i) = 5 Then
      State1(i) = 5
      Call NEIB1(i)
      For j = 1 To 4
        If NN(j) <> -1 Then
          If  ProbArableLand  >  Rnd  And  State(NN(j))  <>  5  And
State(NN(j)) <> 1 Then
            State1(NN(j)) = 4
          End If
        End If
      Next j
   End If
'-------------------------------------------------------------------
  If State(i) = 1 Then
  If Prob4 > Rnd Then
     State1(i) = 1
  End If
   Call NEIB1(i)
   For j = 1 To 4
      If NN(j) <> -1 Then
        If Prob1 > Rnd And State(NN(j)) <> 5 Then
           State1(NN(j)) = 2
        End If
      End If
   Next j
   End If
'-------------------------------------------------------------------
```

```
   If State(i) = 2 Then
   If Prob3 > Rnd Then
      State1(i) = 1
   End If
   Call NEIB1(i)
   For j = 1 To 4
      If NN(j) <> -1 Then
        If Prob2 > Rnd And State(NN(j)) <> 5 Then
           State1(NN(j)) = 2
        End If
      End If
   Next j
   End If
'------------------------------------------------------------------
   If State(i) = 3 Then
   Meadow = 0
   Call NEIB1(i)
   For j = 1 To 4
      If NN(j) <> -1 Then
          If State(NN(j)) = 3 Then
            Meadow = Meadow + 1
          End If
      Else
        Meadow = Meadow – 5
      End If
   Next j
   If Meadow = 4 And ProbVillage > Rnd Then
      State1(i) = 5
   End If
   End If

Next i

For i = 0 To 99
   If State1(i) = 0 Then
      State1(i) = State(i)
   End If
   State(i) = State1(i)
Next i

Call MAP
Call MAP
End Sub

Private Sub LandSurface(ByVal t As Integer)

Dim i As Integer
 Forest(t) = 0
 Bush(t) = 0
 Meadow(t) = 0
 Arable(t) = 0
 Village(t) = 0
 Dim p(5) As Single
```

```
 For i = 0 To 99
 Select Case State(i)
 Case 1
    Forest(t) = Forest(t) + 1
 Case 2
    Bush(t) = Bush(t) + 1
 Case 3
    Meadow(t) = Meadow(t) + 1
 Case 4
     Arable(t) = Arable(t) + 1
 Case 5
     Village(t) = Village(t) + 1
 End Select
 Next i
p(1) = Forest(t) / 100
p(2) = Bush(t) / 100
p(3) = Meadow(t) / 100
p(4) = Arable(t) / 100
p(5) = Village(t) / 100
Shannon(t) = 0
For i = 1 To 5
    If p(i) > 0 Then
        Shannon(t) = Shannon(t) + p(i) * Log(p(i))
    End If
Next i
Shannon(t) = -Shannon(t)
End Sub

Private Sub PIC_MEADOW(ByVal i As Integer)

Dim x0, y0, xx, yy As Single
x0 = cell(i).Left
y0 = cell(i).Top
xx = cell(i).Width
yy = cell(i).Height
Line (x0 + 0.1 * xx, y0 + 0.2 * yy)-(x0 + 0.4 * xx, y0 + 0.2 * yy)
Line (x0 + 0.5 * xx, y0 + 0.1 * yy)-(x0 + 0.5 * xx, y0 + 0.3 * yy)
Line (x0 + 0.6 * xx, y0 + 0.1 * yy)-(x0 + 0.6 * xx, y0 + 0.3 * yy)
Line (x0 + 0.7 * xx, y0 + 0.2 * yy)-(x0 + 0.9 * xx, y0 + 0.2 * yy)
Line (x0 + 0.1 * xx, y0 + 0.4 * yy)-(x0 + 0.2 * xx, y0 + 0.4 * yy)
Line (x0 + 0.3 * xx, y0 + 0.3 * yy)-(x0 + 0.3 * xx, y0 + 0.5 * yy)
Line (x0 + 0.4 * xx, y0 + 0.3 * yy)-(x0 + 0.4 * xx, y0 + 0.5 * yy)
Line (x0 + 0.5 * xx, y0 + 0.4 * yy)-(x0 + 0.7 * xx, y0 + 0.4 * yy)
Line (x0 + 0.8 * xx, y0 + 0.3 * yy)-(x0 + 0.8 * xx, y0 + 0.5 * yy)
Line (x0 + 0.9 * xx, y0 + 0.3 * yy)-(x0 + 0.9 * xx, y0 + 0.5 * yy)
Line (x0 + 0.1 * xx, y0 + 0.7 * yy)-(x0 + 0.4 * xx, y0 + 0.7 * yy)
Line (x0 + 0.5 * xx, y0 + 0.6 * yy)-(x0 + 0.5 * xx, y0 + 0.8 * yy)
Line (x0 + 0.6 * xx, y0 + 0.6 * yy)-(x0 + 0.6 * xx, y0 + 0.8 * yy)
Line (x0 + 0.7 * xx, y0 + 0.7 * yy)-(x0 + 0.9 * xx, y0 + 0.7 * yy)

End Sub

Public Sub PIC_FOREST(ByVal i As Integer)
```

```
Dim x0, y0, xx, yy As Single
x0 = cell(i).Left
y0 = cell(i).Top
xx = cell(i).Width
yy = cell(i).Height
Line (x0 + 0.2 * xx, y0 + 0.1 * yy)-(x0 + 0.2 * xx, y0 + 0.7 * yy)
Line (x0 + 0.1 * xx, y0 + 0.7 * yy)-(x0 + 0.3 * xx, y0 + 0.7 * yy)
Line (x0 + 0.1 * xx, y0 + 0.2 * yy)-(x0 + 0.2 * xx, y0 + 0.1 * yy)
Line (x0 + 0.2 * xx, y0 + 0.1 * yy)-(x0 + 0.3 * xx, y0 + 0.2 * yy)
Line (x0 + 0.1 * xx, y0 + 0.3 * yy)-(x0 + 0.2 * xx, y0 + 0.2 * yy)
Line (x0 + 0.2 * xx, y0 + 0.2 * yy)-(x0 + 0.3 * xx, y0 + 0.3 * yy)
Line (x0 + 0.1 * xx, y0 + 0.4 * yy)-(x0 + 0.2 * xx, y0 + 0.3 * yy)
Line (x0 + 0.2 * xx, y0 + 0.3 * yy)-(x0 + 0.3 * xx, y0 + 0.4 * yy)
Line (x0 + 0.1 * xx, y0 + 0.5 * yy)-(x0 + 0.2 * xx, y0 + 0.4 * yy)
Line (x0 + 0.2 * xx, y0 + 0.4 * yy)-(x0 + 0.3 * xx, y0 + 0.5 * yy)
Line (x0 + 0.1 * xx, y0 + 0.6 * yy)-(x0 + 0.2 * xx, y0 + 0.5 * yy)
Line (x0 + 0.2 * xx, y0 + 0.5 * yy)-(x0 + 0.3 * xx, y0 + 0.6 * yy)
'-----------------------------------------------------------------
Line (x0 + 0.8 * xx, y0 + 0.1 * yy)-(x0 + 0.8 * xx, y0 + 0.7 * yy)
Line (x0 + 0.7 * xx, y0 + 0.7 * yy)-(x0 + 0.9 * xx, y0 + 0.7 * yy)
Line (x0 + 0.7 * xx, y0 + 0.2 * yy)-(x0 + 0.8 * xx, y0 + 0.1 * yy)
Line (x0 + 0.8 * xx, y0 + 0.1 * yy)-(x0 + 0.9 * xx, y0 + 0.2 * yy)
Line (x0 + 0.7 * xx, y0 + 0.3 * yy)-(x0 + 0.8 * xx, y0 + 0.2 * yy)
Line (x0 + 0.8 * xx, y0 + 0.2 * yy)-(x0 + 0.9 * xx, y0 + 0.3 * yy)
Line (x0 + 0.7 * xx, y0 + 0.4 * yy)-(x0 + 0.8 * xx, y0 + 0.3 * yy)
Line (x0 + 0.8 * xx, y0 + 0.3 * yy)-(x0 + 0.9 * xx, y0 + 0.4 * yy)
Line (x0 + 0.7 * xx, y0 + 0.5 * yy)-(x0 + 0.8 * xx, y0 + 0.4 * yy)
Line (x0 + 0.8 * xx, y0 + 0.4 * yy)-(x0 + 0.9 * xx, y0 + 0.5 * yy)
Line (x0 + 0.7 * xx, y0 + 0.6 * yy)-(x0 + 0.8 * xx, y0 + 0.5 * yy)
Line (x0 + 0.8 * xx, y0 + 0.5 * yy)-(x0 + 0.9 * xx, y0 + 0.6 * yy)
End Sub

Private Sub PIC_BUSH(ByVal i As Integer)

Dim x0, y0, xx, yy As Single
x0 = cell(i).Left
y0 = cell(i).Top
xx = cell(i).Width
yy = cell(i).Height
Line (x0 + 0.1 * xx, y0 + 0.2 * yy)-(x0 + 0.2 * xx, y0 + 0.3 * yy)
Line (x0 + 0.2 * xx, y0 + 0.1 * yy)-(x0 + 0.2 * xx, y0 + 0.3 * yy)
Line (x0 + 0.3 * xx, y0 + 0.2 * yy)-(x0 + 0.2 * xx, y0 + 0.3 * yy)
Line (x0 + 0.4 * xx, y0 + 0.2 * yy)-(x0 + 0.5 * xx, y0 + 0.3 * yy)
Line (x0 + 0.5 * xx, y0 + 0.1 * yy)-(x0 + 0.5 * xx, y0 + 0.3 * yy)
Line (x0 + 0.6 * xx, y0 + 0.2 * yy)-(x0 + 0.5 * xx, y0 + 0.3 * yy)
Line (x0 + 0.7 * xx, y0 + 0.2 * yy)-(x0 + 0.8 * xx, y0 + 0.3 * yy)
Line (x0 + 0.8 * xx, y0 + 0.1 * yy)-(x0 + 0.8 * xx, y0 + 0.3 * yy)
Line (x0 + 0.9 * xx, y0 + 0.2 * yy)-(x0 + 0.8 * xx, y0 + 0.3 * yy)
'------------------------------------------------------------------
Line (x0 + 0.2 * xx, y0 + 0.5 * yy)-(x0 + 0.3 * xx, y0 + 0.6 * yy)
Line (x0 + 0.3 * xx, y0 + 0.4 * yy)-(x0 + 0.3 * xx, y0 + 0.6 * yy)
Line (x0 + 0.4 * xx, y0 + 0.5 * yy)-(x0 + 0.3 * xx, y0 + 0.6 * yy)
```

```
Line (x0 + 0.5 * xx, y0 + 0.5 * yy)-(x0 + 0.6 * xx, y0 + 0.6 * yy)
Line (x0 + 0.6 * xx, y0 + 0.4 * yy)-(x0 + 0.6 * xx, y0 + 0.6 * yy)
Line (x0 + 0.7 * xx, y0 + 0.5 * yy)-(x0 + 0.6 * xx, y0 + 0.6 * yy)
'-----------------------------------------------------------------
Line (x0 + 0.1 * xx, y0 + 0.8 * yy)-(x0 + 0.2 * xx, y0 + 0.9 * yy)
Line (x0 + 0.2 * xx, y0 + 0.7 * yy)-(x0 + 0.2 * xx, y0 + 0.9 * yy)
Line (x0 + 0.3 * xx, y0 + 0.8 * yy)-(x0 + 0.2 * xx, y0 + 0.9 * yy)
Line (x0 + 0.4 * xx, y0 + 0.8 * yy)-(x0 + 0.5 * xx, y0 + 0.9 * yy)
Line (x0 + 0.5 * xx, y0 + 0.7 * yy)-(x0 + 0.5 * xx, y0 + 0.9 * yy)
Line (x0 + 0.6 * xx, y0 + 0.8 * yy)-(x0 + 0.5 * xx, y0 + 0.9 * yy)
Line (x0 + 0.7 * xx, y0 + 0.8 * yy)-(x0 + 0.8 * xx, y0 + 0.9 * yy)
Line (x0 + 0.8 * xx, y0 + 0.7 * yy)-(x0 + 0.8 * xx, y0 + 0.9 * yy)
Line (x0 + 0.9 * xx, y0 + 0.8 * yy)-(x0 + 0.8 * xx, y0 + 0.9 * yy)

End Sub

Private Sub PIC_VILLAGE(ByVal i As Integer)

Dim x0, y0, xx, yy As Single
x0 = cell(i).Left
y0 = cell(i).Top
xx = cell(i).Width
yy = cell(i).Height
Line (x0 + 0.1 * xx, y0 + 0.9 * yy)-(x0 + 0.1 * xx, y0 + 0.6 * yy)
Line (x0 + 0.1 * xx, y0 + 0.6 * yy)-(x0 + 0.3 * xx, y0 + 0.4 * yy)
Line (x0 + 0.3 * xx, y0 + 0.4 * yy)-(x0 + 0.5 * xx, y0 + 0.6 * yy)
Line (x0 + 0.5 * xx, y0 + 0.6 * yy)-(x0 + 0.5 * xx, y0 + 0.9 * yy)
Line (x0 + 0.5 * xx, y0 + 0.9 * yy)-(x0 + 0.1 * xx, y0 + 0.9 * yy)
Line (x0 + 0.3 * xx, y0 + 0.4 * yy)-(x0 + 0.6 * xx, y0 + 0.1 * yy)
Line (x0 + 0.5 * xx, y0 + 0.6 * yy)-(x0 + 0.8 * xx, y0 + 0.3 * yy)
Line (x0 + 0.5 * xx, y0 + 0.9 * yy)-(x0 + 0.8 * xx, y0 + 0.6 * yy)
Line (x0 + 0.6 * xx, y0 + 0.1 * yy)-(x0 + 0.8 * xx, y0 + 0.3 * yy)
Line (x0 + 0.8 * xx, y0 + 0.3 * yy)-(x0 + 0.8 * xx, y0 + 0.6 * yy)
Line (x0 + 0.4 * xx, y0 + 0.3 * yy)-(x0 + 0.6 * xx, y0 + 0.5 * yy)
Line (x0 + 0.5 * xx, y0 + 0.2 * yy)-(x0 + 0.7 * xx, y0 + 0.4 * yy)
Line (x0 + 0.2 * xx, y0 + 0.9 * yy)-(x0 + 0.2 * xx, y0 + 0.6 * yy)
Line (x0 + 0.2 * xx, y0 + 0.6 * yy)-(x0 + 0.3 * xx, y0 + 0.6 * yy)
Line (x0 + 0.3 * xx, y0 + 0.6 * yy)-(x0 + 0.3 * xx, y0 + 0.9 * yy)
Line (x0 + 0.35 * xx, y0 + 0.7 * yy)-(x0 + 0.35 * xx, y0 + 0.6 * yy)
Line (x0 + 0.35 * xx, y0 + 0.6 * yy)-(x0 + 0.45 * xx, y0 + 0.6 * yy)
Line (x0 + 0.45 * xx, y0 + 0.6 * yy)-(x0 + 0.45 * xx, y0 + 0.7 * yy)
Line (x0 + 0.45 * xx, y0 + 0.7 * yy)-(x0 + 0.35 * xx, y0 + 0.7 * yy)
Line (x0 + 0.35 * xx, y0 + 0.35 * yy)-(x0 + 0.35 * xx, y0 + 0.15 * yy)
Line (x0 + 0.3 * xx, y0 + 0.15 * yy)-(x0 + 0.48 * xx, y0 + 0.15 * yy)
Line (x0 + 0.45 * xx, y0 + 0.25 * yy)-(x0 + 0.45 * xx, y0 + 0.15 * yy)
Line (x0 + 0.7 * xx, y0 + 0.8 * yy)-(x0 + 0.8 * xx, y0 + 0.9 * yy)
Line (x0 + 0.8 * xx, y0 + 0.7 * yy)-(x0 + 0.8 * xx, y0 + 0.9 * yy)
Line (x0 + 0.9 * xx, y0 + 0.8 * yy)-(x0 + 0.8 * xx, y0 + 0.9 * yy)

End Sub

Private Sub (ByRef ARR() As Integer, ByVal col As Long)
Dim t As Integer
```

```
gra.PSet (1, 100 – ARR(1))
For t = 2 To 100
   gra.Line -(t, 100 – ARR(t)), col
Next t
End Sub
```