Analiza sterowania serwerem mediów z wykorzystaniem protokołu SIP w sieciach typu V2OIP

Abstrakt

W sieciach telekomunikacyjnych typu Voice And Video Over IP można wyróżnić kategorię usług określanych jako usługi multimedialne. Zalicza się do nich m.in. odtwarzanie komunikatów głosowych, interakcję abonenta z systemem IVR oraz usługi konferencyjne. Ich realizacja coraz częściej opiera się na wykorzystaniu serwera mediów (Media Server), który udostępnia szeroki zbiór funkcjonalności przetwarzania strumieni audio/video na potrzeby różnorodnych scenariuszy usługowych. Serwer ten ma również kluczowe znaczenie w procesie migracji usług z tradycyjnych sieci telekomunikacyjnych do sieci NGN.

Celem niniejszej pracy jest dokonanie krytycznej analizy metod sterowania serwerem mediów z wykorzystaniem protokołu SIP w architekturze sieci typu V OIP, podpartej syntezą wiedzy w omawianej tematyce. Ponadto celem pracy jest również opracowanie szczegółowych wymagań na protokół sterowania serwerem mediów i dokonanie pod ich kątem analizy istniejących prototypów takiego protokołu.

W pierwszej części pracy przyjęto definicję usługi multimedialnej oraz przedstawiono krótko protokół SIP i scharakteryzowano architekturę sieciową, w odniesieniu do której prowadzone będą rozważania. W drugiej części pracy omówiono architekturę i funkcje serwera medialnego oraz przeanalizowano jego interakcje z innymi elementami sieci. W

części trzeciej sformułowano zbiór wymagań na protokół sterowania modułem MS i przeprowadzono analizę funkcjonalną kilku prototypów tego protokołu w odniesieniu do tego zbioru. W efekcie wskazano prototyp, który najpełniej umożliwia wykorzystanie serwera mediów w procesie realizacji usług multimedialnych.

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.

TCP/IP i Internet

Adresy IP (IPv4)

W sieciach TCP/IP adres komputera zwany jest adresem IP. Oryginalny adres IP jest czterobajtową (32 bitową) liczbą. Przyjęła się konwencja zapisu każdego bajtu w postaci dziesiętnej i oddzielania ich kropkami. Ten sposób zapisu zwany jest notacją kropkowo-dziesiętną.

Bity w adresie IP są interpretowane jako: <adres sieciowy, adres hosta>

Można jednak niekiedy spotkać inny zapis będący dziesiętnym wyrażeniem 32 bitowej liczby binarnej. Na przykład adres 148.81.78.1 w notacji kropkowo dziesiętnej, będzie w postaci binarnej wyglądał następująco:

10010100010100010100111000000001 zaś dziesiętnie będzie to liczba 2488356353. Określona liczba bitów 32-bitowego adresu IP jest adresem sieciowym, a reszta adresem hostowym. Adres sieciowy określa sieć LAN, zaś adres hosta konkretną stację

roboczą w tej sieci.

By dopasować sieci o różnych rozmiarach (różnej liczbie komputerów), adresy IP podzielono na kilka klas.

Istnieje pięć klas adresów IP: A, B, C, D oraz E, z czego tylko A, B i C są wykorzystywane do adresowania sieci i hostów, a D i E są zarezerwowane do zastosowań specjalnych.

Klasa A obsługuje 126 sieci, z których każda ma ponad 16 milionów hostów, (ponieważ pomimo tego, że jest to adres 7-bitowy, to wartości 0 i 127 mają specjalne znaczenie). Adresy klasy B są przeznaczone dla sieci o rozmiarach do 65534 hostów. Może być, co najwyżej 16384 sieci w klasie B.

Adresy klasy C przeznaczone są dla małych organizacji. Każda klasa C może mieć do 254 hostów, a klas może być ponad 2 miliony.

Klasę sieci można określić na podstawie pierwszej liczby w notacji kropkowo- dziesiętnej:

klasa A: 1.xxx.xxx.xxx do 126.xxx.xxx.xxx klasa B: 128.zzz.xxx.xxx do 191.zzz.xxx.xxx klasa C: 192.zzz.zzz.xxx do 223.zzz.zzz.xxx

Adres z samymi zerami wskazuje na lokalną sieć. Adres 0.0.0.150 wskazuje na host z numerem 150 w tej sieci klasy C.

Adres 127.xxx.xxx.xxx klasy A jest używany do testu zwrotnego (loopback) – komunikacji hosta z samym sobą. Zazwyczaj jest to adres 127.0.0.1.

Dokument RFC 1918 („Address Allocation for Private Internets”) określa, jakie adresy IP mogą być użyte wewnątrz prywatnej sieci. Zarezerwowane są dla nich trzy grupy adresów IP:

od 10.0.0.0 do 10.255.255.255 od 172.16.0.0 do 172.16.255.255 od 192.168.0.0 do 192.168.255.255 Maska sieciowa (IPv4)

Jest to adres IP, który ma jedynki na pozycjach bitów odpowiadających adresom sieciowym i zera na pozostałych (odpowiadających adresom hosta). Klasa adresów sieciowych wyznacza maskę sieciową.

Adresy klasy A mają maskę 11111111000000000000000000000000 czemu w zapisie kropkowo-dziesiętnym odpowiada 255.0.0.0, klasy B: 11111111111111110000000000000000 (255.255.0.0) klasy C zaś: 11111111111111111111111100000000 (255.255.255.0).

Dla wygody używany jest najczęściej zapis kropkowo dziesiętny.

Należy jednak pamiętać, że maska (jak również adres IP) zapisana jest stricte w postaci binarnej.

Należy również zauważyć, że zaczęto nadawać maski niebędące czysto maskami wg klas adresów IP, (czyli, takich, w których liczba jedynek jest wielokrotnością oktetów – liczby 8), lecz zwiększając liczbę jedynek przy takiej samej liczbie bitów (32). Umożliwiło to uzyskanie maski np. 11111111111111111111111111100000 (255.255.255.224) co pozwala na objęcie podsiecią 30 hostów.

Adres sieciowy jest bitowym iloczynem maski sieciowej z którymkolwiek z adresów IP sieci. Jeśli 206.197.168.200 jest adresem IP systemu, a 255.255.0.0 jest maską, to 206.197.0.0 jest adresem sieciowym.

Jeśli zaś 206.197.168.200 jest adresem IP (bitowo 11001110110001011010100011001000) zaś 255.255.192.0 jest maską (bitowo 11111111111111111100000000000000), to iloczyn bitowy daje 206.197.128.0 (bitowo 11001110110001011000000000000000).

Czasami można spotkać skrótowo zapisany adres sieciowy w postaci: 168.100.189.0/24 gdzie część stojąca przed znakiem „/” jest adresem IP zaś liczba 24 jest skrótowo zapisaną maską sieciową. Jest to liczba bitów ustawionych w masce sieciowej na 1, czyli przy standardowej 32 bitowej masce jest to 11111111111111111111111100000000 (255.255.255.0).

System nazw domen

Każdy hostowy komputer w sieci TCP/IP ma swój adres IP. Jednak, ponieważ trudno jest zapamiętać adresy IP nawet kilku hostów, więc powstały specjalne serwery tłumaczące adresy domenowe (postaci: it.pw.edu.pl) na adresy kropkowo-dziesiętne (148.81.78.1). Serwery te nazywane są serwerami DNS (Domain Name Server).

Serwery dokonujące translacji w drugą stronę, to serwery Rev-DNS. System ten nosi nazwę systemu nazw domenowych (Domain Name System).

Nazwa domenowa tworzona jest od strony prawej do lewej. Na początku występują nazwy domen najwyższego poziomu (Top-Level Domains), następnie domeny niższych poziomów, a na końcu znajduje się nazwa hosta.

Nazwy te są oddzielone od siebie kropkami.

Domeny najwyższego poziomu podzielone są na domeny geograficzne (Country Code Domains – dwuliterowe

identyfikatory przyznane poszczególnym krajom np. .uk, .de, .jp, .us, itp.) oraz organizacyjne (Generic Domains – przyznawane w zależności od prowadzonej działalności np. .com, .org, .net, .edu, .gov, .mil, .int).

NAT, PAT, IP-Masqarade i serwery Proxy

Są to technologie umożliwiająca współdzielenie jednego publicznego adresu IP w celu umożliwienia dostępu do Internetu przez wiele komputerów w sieci lokalnej. Stosowane są, dlatego, że liczba publicznych adresów IP (mowa tu cały czas o IPv4) jest dużo mniejsza, niż liczba komputerów podłączonych do Internetu.

Chcąc umożliwić dostęp wielu komputerom w sieci lokalnej do Internetu przy pomocy tylko jednego adresu IP należy

zastosować urządzenie (najczęściej jest to po prostu komputer) podłączone z Internetem pełniące funkcję tzw. bramy z przydzielonym publicznym adresem IP i połączonym z siecią lokalną. Komputerom w sieci lokalnej przydziela się adresy z prywatnej puli adresów IP (takie, które nie występują już w Internecie – określone odpowiednimi, wspomnianymi wcześniej normami i zwane adresami prywatnymi lub czasem nieroutowalnymi). Dzięki takiemu rozwiązaniu każdy komputer w danej sieci lokalnej ma możliwość dostępu do Internetu, zaś z zewnątrz cała sieć lokalna jest widziana jako jeden host.

Technologia NAT (Network Address Translation) polega na mapowaniu adresów zewnętrznych IP do jednego lub więcej adresów IP hostów wewnętrznych.

Technologie PAT (Port Address Translation) oraz IP-Masqarade polegają na tym, że komputer pełniący funkcj ę bramy zajmuje się takim modyfikowaniem ramek pakietów wchodzących i wychodzących z sieci lokalnej, aby możliwy był dostęp poprzez pojedynczy publiczny adres IP, a pakiety przychodzące docierały do właściwych komputerów w sieci lokalnej.

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.

Fizyczna realizacja MS

Fizyczna implementacja serwera mediów może zostać zrealizowana na dwa sposoby:

- Programowo – w postaci aplikacji (software) umiejscowionej na serwerze sprzętowym (np. firmy Hewlett-Packard lub Sun w przypadku serwera mediów firmy

Alcatel-Lucent).
- Sprzętowo – jako dedykowany moduł sprzętowy (np. MS firmy Radisys).

W ramach każdej z tych opcji istnieje szereg dalszych wariantów, np.

- implementacja programowa może być podzielona na osobne aplikacje odpowiadające MRFP i MRFC, zlokalizowane na oddzielnych serwerach
- MRFP i MRFC jako osobne aplikacje w ramach jednego serwera.
- MS jako aplikacja zlokalizowana na jednym serwerze. Kilka takich samych serwerów tworzy cluster w sieci celem zachowania redundancji[1] i zwiększenia wydajności MS.
- implementacja sprzętowa może mieć postać kilku fizycznych maszyn odpowiadających np. jednemu MRFC i kilku MRFP.
- MRFC w postaci aplikacji może kontrolować kilka sprzętowych realizacji MRFP
- fizyczna realizacja rozproszona pomiędzy różne serwery (MS, serwer TTS, serwer z plikami audio, interpreter VXML)

Sposób rzeczywistej implementacji MS zależy zatem bardzo silnie od producenta i operatora sieci telekomunikacyjnej. Szczegółowa analiza wariantów architektury fizycznej nie jest celem tej pracy. Dalsze rozważania dotyczą serwera mediów jako bloku funkcjonalnego w architekturze funkcjonalnej sieci.

MS a jakość usług (Quality of Service)

W sieciach typu V OIP wydajność pracy serwera mediów ma kluczowe znaczenie dla zapewnienia określonego poziomu jakości świadczonych usług[2] (QoS). Organizacje standaryzacyjne nie

definiują jednak wymagań wydajnościowych dla MS/MRF, określają natomiast kryteria dla konkretnych typów usług. W dokumencie [48] opisano kategorie usług sieciowych (obejmujące usługi związane w przetwarzaniem strumieni audio/video), wyróżnione ze względu na wymagany dla użytkownika końcowego sieci poziom QoS. Z kolei w dokumencie [49] zdefiniowane klasy QoS dla usług w sieciach IP oraz wytyczne dla różnych parametrów wydajnościowych sieci w ramach tych klas.

W związku z powyższym można jedynie rozpatrzeć kluczowe cechy fizycznej implementacji MS, które mają wpływ na jego wydajność i zapewniany przez sieć poziom QoS.

- Cechy implementacji MS wpływające na QoS

Do zbioru tych cech zaliczyć można:

- Aspekty logiczne MS

o Obsługa sygnalizacji – logiczna poprawność implementacji protokołów sygnalizacyjnych (wpływa na opóźnienia wynikające z przesyłania błędnych wiadomości i konieczności ponawiania ich wymiany) o Implementacja algorytmów przetwarzania strumieni a/v – rozpatrywana pod względem logicznej poprawności i optymalności (efektywności) – wpływa na przekłamanie danych i jakość sygnału oraz opóźnienia związane z przetwarzaniem sygnałów

- Aspekty fizyczne pracy MS

o Transport danych multimedialnych

- Szybkość transmisji strumieni a/v w ramach interfejsów serwera mediów do sieci transportowej IP
- Wydajność interfejsów transportowych MS – wpływa na ilość jednocześnie obsługiwanych sesji RTP

o Moc obliczeniowa maszyny na której pracuje MS

- Ilość pamięci

- Przestrzeń dyskowa
- Moc procesora o Stabilność systemu
- Odporność na zawieszanie się systemu
- Czas pracy bez konieczności ponownego uruchamiania MS

o Możliwość tworzenia clusterów – redundantnych struktur instancji serwera mediów – wpływa na zwiększenie mocy obliczeniowej MS oraz zabezpieczenie przed niedostępnością systemu

- Parametry opisujące wydajność pracy MS

Na podstawie wymienionych powyżej cech serwera mediów, które wpływają na poziom QoS jego pracy, można zdefiniować szereg parametrów, poprzez które opisuje się ten poziom.:

- Wydajność i pojemność MS

o Maksymalna ilość połączeń/sesji obsługiwanych jednocześnie przez MS

- Przez sesję RTP z terminalem końcowym użytkownika i wykonywanie wszelkich operacji na strumieniach a/v na potrzeby tego połączenia
- Można rozpatrywać sesje różnego typu (np. maksymalna ilość sesji z transkodowaniem pomiędzy kodekami określonych typów, bez transkodowania etc).

o Maksymalna ilość CAPS (Call Atempts Per Second) – maksymalna ilość obsłużonych wywołań SIP przesyłanych do MS w ciągu 1 s. o Maksymalna ilość obsłużonych BHCA (Busy Hour Call Attempts) – wywołań SIP w godzinach największego ruchu w sieci o Maksymalny czas nawiązywania jednego połączenia (ustanawiania sesji SIP)

o Maksymalna ilość obsługiwanych jednocześnie sesji RTP o Maksymalna ilość obsługiwanych jednocześnie konferencji – tu konferencję należy rozumieć jako instancję miksera

o Maksymalna ilość użytkowników w ramach jednej konferencji –

tu ilość użytkowników należy rozumieć jako ilość źródeł strumieni a/v przesyłanych do/z miksera (z każdego źródła mogą pochodzić maksymalnie 2 strumienie a/v)

o Opóźnienie (w ms) pomiędzy odbieraniem a przesyłaniem strumienia (np. przy transkodowaniu strumieni pomiędzy terminalami A i B – strumień odbierany od A jest przetwarzany i przesyłany do B) o Procent wykorzystania zasobów fizycznych maszyny (mocy procesora, pamięci operacyjnej przestrzeni dyskowej) podczas różnych operacji (np. obsługi połączeń konferencyjnych, transkodowanie etc). o Przepustowość (w Mbps) interfejsów sieciowych serwera mediów służących do przesyłania strumieni multimedialnych • Dostępność MS

o High Availibility – parametr określający dostępność systemu, rozumianą jako dostępność dla użytkownika i/lub dla systemów z nim współpracujących. Serwer mediów jest zaliczany do systemów o wysokiej dostępności. Dla takich systemów wartość high availibility wyraża się zazwyczaj w postaci liczby procentów opisywanej liczbą wchodzących w jego skład cyfr 9[3]:

- 9% = niedostępność systemu przez 43.8 min/miesiąc lub 8.76 godzin/rok („trzy dziewiątki”)
- 99% = niedostępność systemu przez 4.38 min/miesiąc lub 52.6 min/rok („cztery dziewiątki „)
- 999% = niedostępność systemu przez 0.44 min/miesiąc lub 5.26 min/rok („pięć dziewiątek „)

o MTBF (Mean Time Between Failures) – maksymalny czas bezawaryjnej pracy systemu bez konieczności jego ponownego uruchamiania

o MTTR (Mean Time To Repair) – maksymalny czas potrzebny na naprawę systemu po wystąpieniu awarii

o Maksymalny czas trwania restartu systemu od wyłączenia go do chwili przywrócenia konfiguracji sprzed wyłączenia o Maksymalny czas na uaktualnienie (upgrade)

- Oprogramowania MS
- Konfiguracji sprzętowej MS

o Szacunkowa częstotliwość niezbędnych aktualizacji oprogramowania/ konfiguracji sprzętowej MS • Parametry sprzętowe MS

o Ilość pamięci operacyjnej, przestrzeni dyskowej, wydajność procesora o Ilość kart rozszerzeń

- Wspomagających przetwarzanie sygnałów
- Zawierających interfejsy sieciowe

o Miara wydajności jednostki centralnej (CPU) maszyny

- Liczba MIPS (Milion Instructions Per Seconds); określa liczbę milionów operacji stałoprzecinkowych wykonywanych przez CPU w ciągu 1 sekundy[4]
- Liczba FLOPS (Floating Point Operations Per Second) – liczba operacji zmiennoprzecinkowych wykonywanych przez CPU w ciągu 1 sekundy.

Należy podkreślić, iż praktyka rynkowa pokazuje, że znacznie bardziej efektywna jest implementacja MS w postaci odrębnej maszyny niż w postaci aplikacji umieszczanej na różnego typu serwerach sprzętowych. Przetwarzanie sygnałów, a zwłaszcza transkodowanie, pochłania bowiem dużo mocy obliczeniowej. Operacje te są przeprowadzane bardziej wydajnie i zużywają mniej zasobów w przypadku sprzętowej implementacji serwera mediów, zwykle przy wykorzystaniu kart sygnałowych z procesorami DSP. Wydajność spada dla programowych implementacji algorytmów przetwarzania sygnałów. Możliwe jest też podejście „mieszane” – tzn. programowa realizacja MS pracująca na serwerze i wykorzystująca zainstalowanie w nim karty z procesorami DSP.

---

[1] Tutaj: nadmiarowość w stosunku do tego co konieczne celem

zmniejszenia prawdopodobieństwa awarii Media Servera w sieci. Serwery z aplikacją MS lub maszyny, które są realizacją MS, są duplikowane i w razie awarii któregoś z nich (lub silnego obciążenia) zadania przejmuje następny. Taka redundantna struktura (cluster) jest przez sieć „widziana" jako jeden Media Server. Algorytmy współdzielenia/przejmowania pracy przez instancje MS w ramach clustera zależą od implementacji i nie są przedmiotem standaryzacji.

[2] Poziom jakości usług świadczonych dla klienta jest określany zwykle w dokumencie SLA – Service Level Agreement

[3] Określanie wartości High Availibility jest oparte o dokładne pomiary czasu dostępności danego systemu wykonane z wykorzystaniem niezawodnych narzędzi, jak i przy udziale osób oraz o szacunkowe obliczenia matematyczne. Ostateczna wartość parametru określa dostępność z punktu widzenia użytkownika (administratora systemu) jak i sieci (system może być bowiem dostępny dla administratora ale niedostępny w sieci i odwrotnie).

[4] Zamiennie wielkość ta określana jest terminem MOPS (Million Operations Per Second) – milion operacji na sekundę

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.

Konstruowanie populacji

genotypów reprezentowanych przez wektory zmiennopozycyjne

Prezentacja zaawansowanej metody kodowania chromosomów przy pomocy ciągów liczb rzeczywistych. Przybliżenie pojęcia równowagi pomiędzy naporem selekcyjnym i różnorodnością genetyczną. Opis zaawansowanych operatorów krzyżowania i mutacji, przystosowanych do oddziaływania na populację genotypów reprezentowanych przez wektory zmiennopozycyjne. Prezentacja sposobu wzbogacenia algorytmu genetycznego poprzez opracowanie metody dynamicznej zmiany rozkładu prawdopodobieństwa wyboru operatora mutowania.

Reprezentacja binarna używana tradycyjnie w algorytmach genetycznych ma pewne wady, gdy stosuje się ją do rozwiązywania wielowymiarowych zadań wymagających dużej dokładności. Na przykład przy 100 zmiennych i dziedzinie o zakresach [-500, 500] oraz żądanej dokładności 6 cyfr po przecinku długość binarnego wektora rozwiązań wynosi 3000. To z kolei prowadzi do przestrzeni o liczebności rzędu 10^{1000}. Dla takich zadań klasyczne algorytmy genetyczne działają słabo.

W miejsce reprezentacji przy pomocy wektorów binarnych najczęściej stosuje się reprezentację zmiennopozycyjną. Każdy chromosom jest ciągiem liczb rzeczywistych o długości takiej samej jak wektor oczekiwanego rozwiązania. Dokładność tego sposobu kodowania zależy od używanego komputera, ale ze względu na popularność mikroprocesorów wspomagających operacje na liczbach zmiennopozycyjnych jest z reguły znacznie wyższa niż przy reprezentacji binarnej. Zwiększanie dokładności reprezentacji binarnej poprzez wprowadzenie większej liczby bitów jest złym rozwiązaniem, gdyż zwalnia działanie algorytmu.

Populację chromosomów reprezentowanych przez ciągi liczb rzeczywistych o liczbie elementów równej liczbie zmiennych sterujących funkcji celu konstruuje się przy pomocy iteracyjnego wzoru 2.1. Pojedyncza zmienna składowa genotypu oznaczona jest literą g_i, a w dwukolumnowej macierzy D o liczbie wierszy równej liczbie zmiennych sterujących zapisane są granice dziedziny. Oznacza to, że długość przedziału dozwolonych wartości każdego argumentu funkcji celu może być inna.

$$g_i = real\left(D_{1,i}, D_{2,i}\right) \quad (2.1.)$$

gdzie g_i – i-ty gen w chromosomie,

real(x,y) – funkcja zwracająca losową liczbę rzeczywistą, należącą do przedziału [x , y].

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.

CSS i XML

Arkusz stylów CSS

Kaskadowe arkusze stylów (ang. Cascading Style Sheets – CSS) jest językiem służącym do opisu sposobu renderowania stron WWW. Arkusz taki jest listą reguł ustalających, jak ma zostać renderowana przez przeglądarkę internetową zawartość elementu HTML lub XML. CSS zostały stworzone aby oddzielić sposób formatowania stron WWW od struktury logicznej dokumentu.

Umożliwiło to znaczne uproszczenie kodu HTML.

XML

XML, czyli eXtensible Markup Language – rozszerzalny język znaczników, zaprojektowany przez World Wide Web Consortium. XML, w przeciwieństwie do np. HTML nie jest językiem służącym do przechowywania konkretnych danych. Jest natomiast językiem opisującym dane, czyli tzw. metajęzykiem. Przede wszystkim nie jest to język do tworzenia dokumentów, lecz język do tworzenia innych języków. Dopiero te języki służą do budowania dokumentów w Internecie. Języki takie nazywamy aplikacjami XML. Najbardziej znane z nich to m.in.:

- XHTML (następca HTML),
- MathML (tworzenie dokumentów zawierających formuły matematyczne),
- WAP (wyświetlanie internetowej informacji w telefonach komórkowych),
- GedML (obsługa danych genealogicznych)
- MusicML (notacja muzyczna)
- VoiceML (dane głosowe)
- ThML (teksty teologiczne)
- VML i SVG (grafika wektorowa)
- XMLNews (wymiana aktualnych wiadomości)
- RSS (kanały informacyjne).

Aplikacji takich powstało mnóstwo i są to języki służące biznesowi, naukowcom, wyspecjalizowanym mediom itd.

XML jest podzbiorem języka SGML (każdy dokument XML jest zarazem dokumentem SGML, oferuje 80% jego możliwości, przy dziesięciokrotnie łatwiejszym ich wykorzystaniu). Dzięki XML sami możemy tworzyć własne znaczniki, a przeglądarka będzie wyświetlała je w żądany sposób. Język ten nie ma ograniczonej liczby znaczników, gdyż pozwala przechowywać dowolne dane i to w jak najbardziej wygodny dla nas sposób, bo sami go określamy. Sami określamy strukturę danych, która może być

tabelaryczna, ale może także tworzyć drzewo. W ten sposób nie jesteśmy, jako twórcy zbiorów XML w żaden sposób ograniczeni. Na tym przede wszystkim polega wyższość XML nad innymi formatami zapisu danych.

Dzięki oddzieleniu treści od formy – czego nie umożliwia wspomniany HTML – łatwo skupić się na samych danych. Zwykłe programy mogą dzięki wspólnemu formatowi XML łatwiej wymieniać dane, a informacje publikowane np. w Sieci mogą być łatwiej przetwarzane.

Przykład zastosowania j ęzyka XML:

<?xml version=”1.0″?>

<news numer=”1″>

<tytul>Piractwo komputerowe zostanie ukrócone!</tytul>

<data>11 grudnia 2006</data>

<autor>Bartek Ziobrowski</autor>

<tresc><nazwisko>Mariusz Dutko</nazwisko> oświadczył, że powoła grupę do walki z piractwem. We współpracy z <firma>Microsoft</firma>, <firma>KMP</firma> i <firma>RIAA</firma> powoła grupę zwaną Agencją Ochrony Własności Intelektualnej (<skrot>AOWI</skrot>).</tresc>

</news>

Jak widać składnia dokumentu jest bardzo schludna. Każdy znacznik jest czytelny i jasno określa zawartość. Oczywiście nad interpretacją takich znaczników przez cały czas czuwa procesor XML, dzięki któremu przeglądarka wyświetla je w żądany sposób – jest to tzw. parser XSLT.

Jak każdy język, XML ma również swoje zasady, których należy przestrzegać. Nieobowiązkowa jest deklaracja XML, znajdująca się na początku dokumentu. Zalecane jest jednak, aby ją

umieścić. Jej zawartość wygląda następująco:

<?xml version=”1.0″ encoding=”ISO-8859-2″ standalone=”yes”?>

Jak widać, zdefiniowana jest wersja języka XML, zawiera również informację o kodowaniu, natomiast atrybut „standalone” określa, czy dokument może występować samodzielnie czy też będzie on potrzebował odwołania do zewnętrznego DTD.

Znaczniki konstruuje się identycznie jak w HTML, jednak w języku XML – w odróżnieniu od HTML – każdy znacznik musi być zamknięty.

Bardzo ważną sprawą jest wielkość liter. XML nie wymaga stosowania ujednoliconej, konkretnej wielkości liter, aczkolwiek konieczna jest konsystencja w wykorzystaniu jednego systemu. I tak <news> nie jest tym samym znacznikiem co <News> czy <NEWS>. Nie można go również zamknąć korzystając z </NEWS>.

Kolejnym ważną cechą, o której należy pamiętać to hierarchia elementów. Dla poprawienia spójności kodu w specyfikacji XML przyj ęto zasadę, że każdy znacznik musi zostać zamknięty zanim zostanie zamknięty każdy z jego rodziców. Biorąc pod uwagę powyższy przykład zastosowania XML, nie możemy napisać:

<tytuł><data>11 grudnia 2006</news></tytuł>

ale musimy to zanotować tak:

<tytuł><data>11 grudnia 2006</data></tytuł>

Sprawa atrybutów rozwiązana jest podobnie jak w przypadku HTML. Ich wartości muszą być zamknięte w cudzysłów, aczkolwiek nie ma znaczenia czy będzie on podwójny czy pojedynczy. Spacje wokół znaku równości również nie odgrywają roli, np.:

<autor imie=”Mariusz” nazwisko=”Dutko” />

oznacza to samo co:

<autor imie = 'Mariusz’ nazwisko = 'Dutko’ />

Nazwy dotyczące nazewnictwa są wspólne są wspólne zarówno dla nazw elementów, jak i atrybutów oraz pewnych rzadziej używanych konstrukcji. Nazwy w XML mogą zawierać dowolne znaki alfanumeryczne, w tym wszelkie znaki narodowe, cyfry i trzy znaki przestankowe: podkreślenie, myślnik i kropkę. Niedozwolone są inne znaki specjalne, jak apostrofy, znaki zapytania, wykrzykniki, procenty itp. Nazwy muszą znajdować się w całości w jednej linii bez żadnych spacji. Dwa ze znaków specjalnych nie mogą być umieszczane wewnątrz elementów, są to: „&”, który należy zastąpić „&” oraz „<”, zastępowany przez „<”.

Specjalne bloki CDATA służą do umieszczania zawartości, którą parser potraktuje jako czysty tekst i przepuści ją bez przetwarzania. Wszelkie elementy XML wewnątrz CDATA zostaną przez parser zignorowane.

Dokumenty XML, podobnie jak HTML, można tworzyć i zmieniać w dowolnym edytorze tekstu, jednak wygodniej jest korzystać z wyspecjalizowanych narzędzi, które znacznie ułatwią pracę przy tworzeniu dokumentów.[16][17]

---

[16]„Kurs XML”, webdiary.pl/technologie+list.cat+27.htm

[17]„Wprowadzenie do XML”, Michał Paluchowski, nethut.pl/artykul/120/2

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.

Struktura programu GenMachine i schemat blokowy

Program GenMachine składa się z dwóch części. Pierwsza część, umieszczona w pliku main.cpp jest sterująca, rekompilowana dla każdej funkcji celu, druga natomiast jest biblioteką statyczną o nazwie GenMachine.lib, dołączaną za każdym razem do części sterującej i stanowi niezmienny trzon systemu symulacji ewolucji.

Część sterująca ma za zadanie odwoływać się do funkcji biblioteki z pożądanymi w danej chwili argumentami. Wykonuje to w następujących kilkunastu krokach:

- Zainicjowanie zmiennych niezbędnych do pracy programu, takich jak:

logFileName – ścieżka do pliku w którym znajdzie się raport,

iniFileName – ścieżka do pliku tekstowego zawierającego parametry,

population_size – maksymalna liczba osobników w jednym pokoleniu,

vars_number – ilość zmiennych sterujących, czyli długość genotypu.

- Wywołanie konstruktora obiektu klasy GenMachine, wraz z przekazaniem mu nazwy pliku z parametrami, najlepiej ini oraz rozmiaru populacji. W przypadku nie podania drugiego argumentu, konstruktor przyjmuje domyślny rozmiar populacji równy 100 genotypów.
- Ustalenie przedziałów w jakich muszą się mieścić zmienne sterujące funkcji celu i zapisanie ich w zmiennej tablicowej minmax.
- Wywołanie funkcji Init z argumentami określającymi rozmiar genotypu oraz dziedzinę funkcji celu.

- Sprawdzenie czy Etap Inicjalizacji, czyli alokacja pamięci, utworzenie egzemplarza klasy GenMachine oraz utworzenie populacji początkowej przebiegło pomyślnie. Ewentualny numer błędu zostaje zwrócony przez funkcję Init i zachowany w zmiennej
- Uruchomienie iteracyjnego procesu symulacji poprzez wywołanie funkcji Start, zwracającej najlepsze napotkane dostosowanie, czyli notabene poszukiwane optimum funkcji.
- Pobranie wartości zmiennych sterujących z najlepiej dostosowanego genotypu.
- Sporządzenie raportu z pracy programu, zawierającego:

– czas trwania symulacji,

– najlepsze dopasowanie,

– wektor zmiennych sterujących najlepiej dopasowanego genotypu oraz

– następujące dane dotyczące każdego pokolenia: numer pokolenia, najlepsze dostosowanie, średnie dostosowanie, odchylenie standardowe dostosowań, liczba wystąpień krzyżowania każdego typu, liczba wystąpień operatora krzyżowania w ogóle, liczba wystąpień mutowania każdego typu, liczba wystąpień operatora mutowania w ogóle.

- Opcjonalnie, jeśli maksymalna liczba genotypów w pokoleniu nie jest zbyt duża, zachowanie w dodatkowym pliku tekstowym wszystkich zmiennych sterujących, czyli składników każdego osobnika.

Budowę drugiej części programu, czyli biblioteki GenMachine, można przybliżyć przy pomocy podziału ze względu na definicje używanych klas obiektów, lub też ze względu na etapy w jakich jej kod jest wykonywany. W programie używa się trzech klas obiektów:

- Genotype – niewielka klasa służąca do opisu pojedynczego

osobnika. Składa się zaledwie z trzech metod: konstruktora, destruktora, inicjalizatora oraz czterech zmiennych składowych, wystarczających do opisania genotypu:

Gene – jednowymiarowa tablica, czyli wektor wszystkich genów w postaci liczb rzeczywistych;

Fitness – dostosowanie osobnika, czyli wartość jaką zwróciła dla tego osobnika funkcja Formula;

Rfitness – względne dopasowanie osobnika, czyli jego wartość na tle całego aktualnego pokolenia; suma wszystkich względnych dopasowań wynosi jeden, a zatem dopasowanie względne jest prawdopodobieństwem wyboru genotypu do reprodukcji;

Cfitness – dopasowanie łączne, zawiera sumę dopasowania względnego bieżącego osobnika i sumy dopasowań względnych wszystkich osobników o mniejszym numerze porządkowym.

- IniFile – obiekt powiązany z plikiem ini, z zaimplementowanymi algorytmami szybkiego odnajdywania sekwencji znaków w plikach tekstowych, służy do pobrania szesnastu parametrów nie przekazywanych jako argumenty funkcji inicjujących. Oto przykładowa zawartość pliku GenMachine.ini wraz z komentarzem:

[main]

MAXGENS = 40000 – maksymalna liczba pokoleń

[selection]

rozmiar turnieju w turniejowej metodzie selekcji, lub dla wartości 0 i 1 selekcja proporcjonalna

(typu real) prawdopodobieństwo krzyżowania

prawdopodobieństwo wyboru krzyżowania prostego

prawdopodobieństwo wyboru krzyżowania arytmetycznego

prawdopodobieństwo wyboru krzyżowania heurystycznego

cierpliwość krzyżowania heurystycznego, czyli liczba prób spłodzenia potomka dopuszczalnego (wewnątrz dziedziny)

(typu real) prawdopodobieństwo mutacji

prawdopodobieństwo wyboru pierwszej metody mutowania w pierwszym pokoleniu

prawdopodobieństwo wyboru pierwszej metody mutowania w ostatnim pokoleniu

prawdopodobieństwo wyboru drugiej metody mutowania w pierwszym pokoleniu

prawdopodobieństwo wyboru drugiej metody mutowania w ostatnim pokoleniu

prawdopodobieństwo wyboru trzeciej metody mutowania w pierwszym pokoleniu

prawdopodobieństwo wyboru trzeciej metody mutowania w ostatnim pokoleniu

liniowa zmiana prawdopodobieństwa wyboru 1 – tak, 0 – nie

DEGREE = 4 – stopień niejednorodności (3. typ mutacji);

parametr mutacji nierównomiernej (patrz metoda GenMachine::delta())

- GenMachine – główna klasa aplikacji, składająca się z trzydziestu pięciu metod i trzydziestu czterech zmiennych składowych; jeden egzemplarz klasy GenMachine, deklarowany w części sterującej, zawiaduje całym procesem symulacji, deklaruje obiekty pozostałych klas i rezerwuje dla nich zasoby; jej przejrzysty interfejs oraz obsługa błędów umożliwia kalibrację parametrów.

Kod programu GenMachine wykonywany jest w dwóch etapach. W

Etapie Inicjalizacji wywołuje się konstruktora klasy GenMachine, następnie jedną z metod tej klasy o nazwie Init. Etap Symulacji polega na przekazaniu sterowania metodzie Start.

Etap Inicjalizacji

Do czynności jakie wykonywane są po wywołaniu konstruktora klasy GenMachine należą:

- zainicjowanie generatora liczb losowych,
- zapamiętanie przekazanego jako argument wywołania funkcji rozmiaru populacji w zmiennej popsize,
- alokacja pamięci dla zmiennej tablicowej pop, czyli dla podstawowej (aktualnej) populacji obiektów klasy Genotype w liczbie równej popsize,
- alokacja pamięci dla zmiennej tablicowej newpop, czyli dla dodatkowej (nowej) populacji genotypów,
- pobranie parametrów metody z pliku ini, przy pomocy klasy IniFile,
- unormowanie prawdopodobieństw wyboru metod krzyżowania i mutowania,
- rezerwacja pamięci dla tablic raportu oraz pomocniczej tablicy index wspierającej selekcję turniejową.

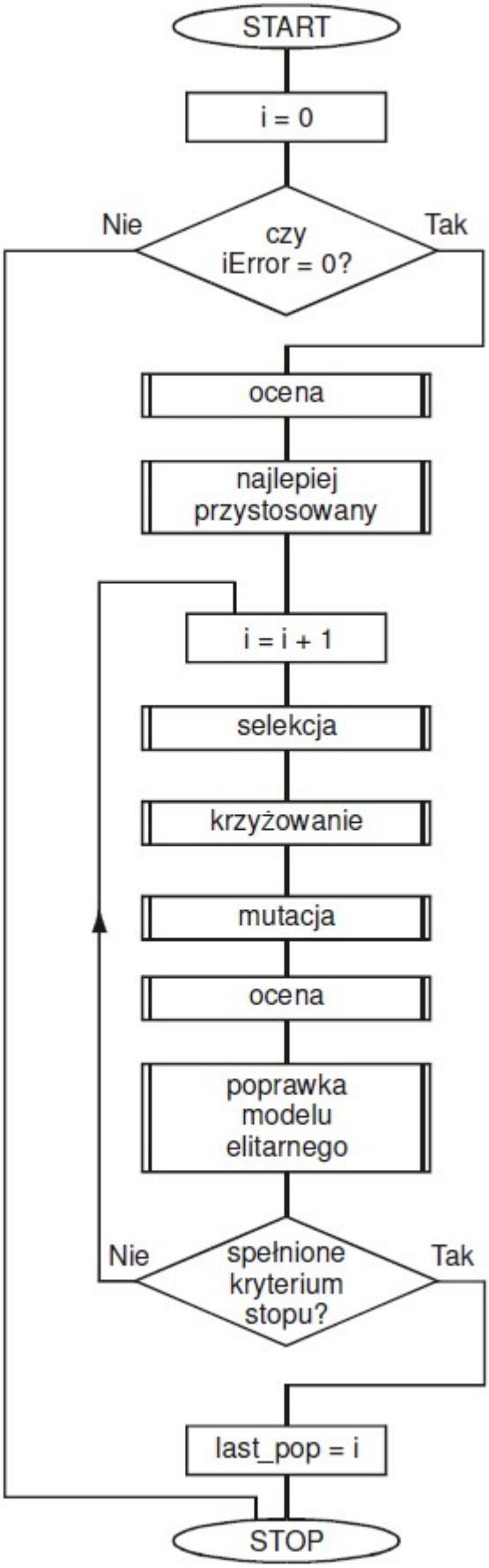
Pozostałe czynności Etapu Inicjalizacji wykonuje metoda Init. Są to:

- sprawdzenie poprawności wykonania konstruktora, czyli kontrola wartości zmiennej iError,
- zapamiętanie podanej jako argument wywołania funkcji Init długości genotypu w zmiennej składowej nvars,
- przepisanie podanych jako argument granic przedziałów do których należeć muszą wartości kolejnych zmiennych decyzyjnych do zmiennej tablicowej minmax,
- utworzenie populacji początkowej przy pomocy inicjalizatora klasy Genotype, który rezerwuje pamięć oraz metody klasy GenMachine o nazwie First – funkcja

First inicjuje wszystkie osobniki wartościami losowymi należącymi do dziedziny funkcji celu,

- wsparcie modelu elitarnego przez rezerwację pamięci dla najlepszego dotychczas osobnika,
- ustawienie flagi iError na zero – **Etap Inicjalizacji** zakończony pomyślnie.

Rys. 4.1. Schemat blokowy Etapu Symulacji


START
i = 0
Nie
czy
iError = 0?
Tak
ocena
najlepiej
przystosowany
i = i + 1
selekcja
krzyżowanie
mutacja
ocena
poprawka
modelu
elitarnego
Nie
spełnione
kryterium
stopu?
Tak
last_pop = i
STOP

Etap Symulacji.

Wszystkie bloki instrukcji wykonywane podczas Etapu Symulacji, czyli po przekazaniu sterowania do funkcji Start oraz kolejność ich wykonywania przedstawione są na Rys. 4.1. Znaczenie poszczególnych klatek schematu blokowego podane jest w poniższym zestawieniu:

- i = 0 – zainicjowanie licznika iteracyjnej pętli głównej zerem. W funkcji Start pełni on zarazem rolę licznika pokoleń.
- Czy iError = 0? – klatka decyzyjna sprawdzająca status błędów aplikacji. Zerowa wartość zmiennej iError oznacza bezbłędne zakończenie Etapu Inicjalizacji. Wartość różna od zera powoduje awaryjne przerwanie działania aplikacji.
- Ocena – wywołanie metody evaluate. Ocena wszystkich osobników aktualnej populacji, poprzez wielokrotne wywołanie funkcji Formula, zawierającej funkcję dostosowania, z wektorem zmiennopozycyjnym jako argumentem wywołania oraz zapisanie wyniku w zmiennej składowej Fitness klasy Genotype.
- Najlepiej przystosowany – wywołanie metody keep_the_best. Implementacja modelu elitarnego selekcji. Sprawdzenie czy po ostatniej ocenie pojawił się osobnik lepiej dostosowany od najlepiej ocenionego dotychczas. Jeśli tak, zapamiętanie najlepszego genotypu.
- i = i + 1 – zwiększenie licznika pokoleń na początku pętli głównej o jeden. Efektem tej zmiany jest zachowanie numerów pokoleń w przedziale [1 , maxgens].
- Selekcja – wywołanie metody select. Wybór genotypów do prokreacji, w zależności od wartości zmiennej składowej tourn, albo przy pomocy selekcji proporcjonalnej po uprzedniej korekcie ujemnego dopasowania, polegającej na przesunięciu wszystkich dopasowań do zbioru liczb nieujemnych, albo przy pomocy selekcji turniejowej, o rozmiarze turnieju ustalonym w ini.

- Krzyżowanie – wywołanie metody crossover(int pop_num). Losowy wybór dwóch osobników zgodnie z podanym przez parametr PXOVER prawdopodobieństwem krzyżowania, następnie utworzenie na ich miejsce dwóch nowych, przy pomocy wybranej losowo metody krzyżowania, zgodnie z przyjętym rozkładem prawdopodobieństwa. Możliwy jest wybór jednej z trzech zaimplementowanych metod:

krzyżowanie proste (jednopunktowe), krzyżowanie arytmetyczne,

krzyżowanie heurystyczne wspierane krzyżowaniem dwupunktowym. Opcjonalny parametr pop_num, dzięki któremu przekazać można metodzie crossover aktualny numer pokolenia, pozwala w razie potrzeby na dynamiczną zmianę częstotliwości krzyżowania (zmienna pxover), lub też zmianę rozkładu prawdopodobieństwa wyboru metody krzyżowania, już podczas trwania symulacji.

- Mutacja – wywołanie metody mut(int pop_num). Sprawdzenie przy pomocy dwóch zagnieżdżonych pętli każdego elementu składowego, wszystkich wektorów zmiennopozycyjnych, będących genotypami aktualnego pokoleniu, czy zgodnie z przyj ętym prawdopodobieństwem wystąpienia mutacji, podanym jako parametr PMUTATION, osobnik zmutował. Jeśli tak, zgodnie z przyjętym statycznym lub dynamicznym rozkładem prawdopodobieństwa, wybór jednego z trzech zaimplementowanych typów mutacji:

mutacji równomiernej, mutacji brzegowej, mutacji nierównomiernej.

Argument wywołania metody mut umożliwia nie tylko zastosowanie dynamicznego, zmieniającego się w czasie rozkładu prawdopodobieństwa wyboru typu mutacji, ale również opcjonalnie zmiany częstotliwości występowania mutacji w zależności od czasu trwania symulacji – czasu rozumianego jako numer populacji.

- Poprawka modelu elitarnego – wywołanie metody elitist. Sprawdzenie całej aktualnej populacji w poszukiwaniu

najlepiej dostosowanego osobnika, porównanie go z najlepszym napotkanym dotychczas genotypem i w zależności od wyniku porównania, albo zapamiętanie go jako najlepszego dotychczas i oznacza to postęp w poszukiwaniu optimum globalnego, albo zastąpienie najgorszego osobnika w bieżącym pokoleniu genotypem jak dotąd najlepszym.

- Spełnione kryterium stopu? – klatka decyzyjna sprawdzająca, czy nie zostało spełnione którekolwiek określonych kryteriów stopu. Niepomijalnym kryterium stopu jest wartość parametru MAXGENS, czyli największego numeru pokolenia. Poza nim na potrzeby eksperymentu, można w metodzie Stop(int pop_num) zdefiniować dodatkowe kryterium zatrzymania symulacji. Może nim być podobnie jak w innych technikach optymalizacji numerycznej różnica między aktualnym wynikiem, a obliczonym inną metodą wynikiem optymalnym, lub prędkość poprawiania się wyniku, lub odchylenie standardowe wartości dostosowań.
- Last_pop = i – zapamiętanie numeru ostatniego pokolenia, na potrzeby części sterującej aplikacji, która generuje raporty. Numer ostatniego pokolenia może zostać pobrany przez część sterującą, przy pomocy metody GetBest.

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.

Systemy zarządzania treścią

(CMS): kluczowe narzędzie współczesnego zarządzania informacją

Systemy zarządzania treścią (ang. Content Management System) umożliwiają tworzenie i modernizację serwisów WWW bez znajomości technologii wykorzystanej do budowy serwisu.

Tworzenie, formatowanie i wprowadzanie treści w systemie CMS odbywa się za pomocą prostych w obsłudze interfejsów użytkownika, najczęściej prostych w obsłudze formularzy, przypominających zazwyczaj pracę z edytorem tekstu. Tak prosta obsługa pozwala na modyfikowanie serwisu niemal każdemu.

Zadanie systemów CMS polega na wyodrębnieniu zawartości menu głównego (treści informacyjnej serwisu) od formularza (drogi). I wypełnia ją bazą danych poprzez system przenoszenia wszystkich danych wprowadzonych w odpowiednie miejsca i jest kwalifikowana u Autora. Projekt strony modyfikuje sposób wykorzystania graficznego całej strony, przygotowanie strony i sformułowanie systemu spisywania późniejszych środków narzuca.

Takie rozwiązanie pozwala nam na swobodną zmianę wyglądu serwisu i aby wprowadzać nowe zmiany w serwisie redaktorzy nie muszą poznawać technologii wykorzystywanych przy tworzeniu serwisów internetowych.

Oczywiście bardzo wygodne i wskazane jest zastosowanie systemów CMS w aplikacjach wspomagających pracę agencji reklamowych, gdyż można dzięki temu całą aplikację wdrożyć jako kompletny serwis WWW, oferując dostęp klientom do zasobów. Dzięki takiemu rozwiązaniu pracownicy agencji mogą w dowolnym momencie modyfikować treści zamieszczane na łamach serwisu WWW.

Systemy zarządzania treścią, znane powszechnie pod skrótem CMS

(ang. Content Management System), odgrywają niezwykle ważną rolę we współczesnym zarządzaniu informacją i treściami cyfrowymi. W dobie dynamicznie rozwijającego się internetu, rosnącej ilości danych oraz coraz większych potrzeb w zakresie publikacji treści online, systemy CMS stały się nieodzownym narzędziem umożliwiającym firmom, organizacjom i osobom indywidualnym zarządzanie treściami w prosty, szybki i efektywny sposób.

Definicja i podstawowe funkcje systemów CMS

System zarządzania treścią (CMS) to oprogramowanie, które umożliwia tworzenie, edytowanie, organizowanie i publikowanie treści na stronach internetowych bez potrzeby zaawansowanej wiedzy technicznej. Dzięki CMS, użytkownicy mogą zarządzać zawartością strony internetowej, taką jak teksty, obrazy, filmy czy inne multimedia, w intuicyjny sposób za pomocą interfejsu graficznego, często zbliżonego do edytorów tekstu. Podstawową ideą CMS jest oddzielenie treści od struktury i wyglądu strony internetowej, co oznacza, że użytkownicy mogą dodawać i edytować treści, nie ingerując w techniczne aspekty działania strony, takie jak kodowanie czy projektowanie.

Najważniejsze funkcje, które oferują systemy CMS, to:

- **Tworzenie i edytowanie treści**: CMS pozwala użytkownikom na tworzenie, edycję i publikację treści w czasie rzeczywistym. Użytkownicy mogą dodawać teksty, obrazy, filmy, dokumenty i inne pliki multimedialne w prosty sposób, często przy użyciu edytorów typu „przeciągnij i upuść” (drag-and-drop).
- **Zarządzanie mediami**: Systemy CMS oferują zaawansowane funkcje zarządzania multimediami, takie jak organizowanie obrazów, filmów czy dokumentów w biblioteki, co ułatwia późniejsze ich wyszukiwanie i wykorzystanie.

- **Personalizacja treści**: CMS pozwala na personalizację treści w zależności od profilu użytkownika, co oznacza, że różne grupy odbiorców mogą zobaczyć różne wersje tej samej strony w zależności od swoich preferencji, lokalizacji geograficznej czy historii przeglądania.
- **Zarządzanie użytkownikami i uprawnieniami**: W systemach CMS możliwe jest tworzenie różnych poziomów dostępu dla użytkowników. Administratorzy mogą nadawać uprawnienia do edycji, publikacji czy przeglądania treści, co jest szczególnie istotne w dużych organizacjach.
- **SEO i optymalizacja wydajności**: Współczesne systemy CMS oferują narzędzia wspierające optymalizację treści pod kątem wyszukiwarek internetowych (SEO), co zwiększa widoczność stron internetowych w wynikach wyszukiwania.

Typy systemów zarządzania treścią

Wyróżnia się kilka typów systemów zarządzania treścią, które różnią się funkcjonalnością, zakresem zastosowań oraz stopniem zaawansowania technologicznego.

1. Open Source CMS

Systemy CMS typu open source są dostępne publicznie i mogą być dowolnie modyfikowane przez użytkowników. Najbardziej popularnymi systemami tego typu są WordPress, Joomla! i Drupal.

- **WordPress**: To najpopularniejszy na świecie system zarządzania treścią, który początkowo był zaprojektowany z myślą o blogach, ale obecnie jest używany do tworzenia różnorodnych stron internetowych, w tym sklepów internetowych, portali informacyjnych, a nawet stron korporacyjnych. Jego popularność wynika z prostoty obsługi, ogromnej liczby dostępnych wtyczek oraz szerokiej społeczności użytkowników.
- **Joomla!**: Jest to bardziej zaawansowany system CMS, który pozwala na tworzenie bardziej złożonych struktur stron

internetowych. Joomla! oferuje wiele zaawansowanych funkcji, takich jak system zarządzania użytkownikami, co czyni go idealnym wyborem dla stron społecznościowych i portali internetowych.

- **Drupal**: To CMS skierowany do bardziej zaawansowanych użytkowników, oferujący bardzo wysoką elastyczność i skalowalność. Drupal jest popularny wśród dużych organizacji i instytucji, które potrzebują rozbudowanych funkcji, dużego bezpieczeństwa i zaawansowanych możliwości zarządzania.

2. Proprietary CMS

Proprietary CMS to systemy, które są opracowywane i dystrybuowane przez prywatne firmy. Przykładem takich systemów są Adobe Experience Manager czy Sitecore. Systemy te są zazwyczaj bardziej kosztowne niż ich open source'owe odpowiedniki, ale oferują szeroką gamę zaawansowanych funkcji, w tym zaawansowane narzędzia do analizy danych, personalizacji treści i integracji z innymi systemami biznesowymi.

3. Headless CMS

Headless CMS to nowoczesne rozwiązanie, które oddziela warstwę zarządzania treścią od warstwy prezentacyjnej (front-end). W przeciwieństwie do tradycyjnych systemów CMS, gdzie warstwa zarządzania treścią i warstwa prezentacyjna są ściśle ze sobą powiązane, w headless CMS treści są dostarczane za pomocą interfejsu API do dowolnych aplikacji, takich jak strony internetowe, aplikacje mobilne czy platformy e-commerce. Takie podejście umożliwia większą elastyczność, zwłaszcza w kontekście dostarczania treści na różne urządzenia i kanały.

Zastosowania CMS w różnych branżach

Systemy CMS są szeroko stosowane w różnych branżach i sektorach gospodarki. Dzięki swojej elastyczności i łatwości użytkowania, mogą być wykorzystywane zarówno przez małe firmy, jak i międzynarodowe korporacje.

1. Media i publikacje

W branży medialnej, systemy CMS są wykorzystywane do zarządzania treściami w portalach informacyjnych, blogach oraz stronach z wiadomościami. Firmy medialne, takie jak gazety czy telewizje, wykorzystują CMS do szybkiego publikowania nowych artykułów, zarządzania archiwami oraz personalizacji treści dla różnych grup odbiorców.

2. E-commerce

W sektorze e-commerce systemy CMS są podstawowym narzędziem do zarządzania sklepami internetowymi. Pozwalają na łatwe zarządzanie produktami, tworzenie i edycję opisów produktów, a także integrację z systemami płatności oraz logistycznymi. Popularne platformy e-commerce, takie jak Shopify czy Magento, oferują szeroką gamę funkcji CMS.

3. Instytucje publiczne

Rządy oraz instytucje publiczne coraz częściej wykorzystują CMS do zarządzania informacjami i komunikacji z obywatelami. Systemy te umożliwiają tworzenie portali, które są łatwe w nawigacji, a jednocześnie zawierają dużą ilość treści związanych z prawem, usługami publicznymi czy edukacją.

4. Edukacja

Uczelnie oraz inne instytucje edukacyjne używają systemów CMS do zarządzania swoimi stronami internetowymi, portalami dla studentów oraz platformami e-learningowymi. Systemy te umożliwiają łatwe publikowanie materiałów dydaktycznych, zarządzanie kursami oraz komunikację ze studentami.

Wyzwania związane z wdrożeniem CMS

Mimo licznych zalet, wdrożenie systemu CMS wiąże się także z pewnymi wyzwaniami, które mogą wpłynąć na skuteczność zarządzania treścią.

1. Bezpieczeństwo

Systemy CMS, zwłaszcza te typu open source, są narażone na różnorodne zagrożenia bezpieczeństwa, takie jak ataki hakerskie czy malware. Regularne aktualizacje systemów oraz stosowanie odpowiednich zabezpieczeń są kluczowe dla minimalizacji ryzyka.

2. Skalowalność

W miarę wzrostu organizacji i ilości treści, systemy CMS mogą wymagać większych zasobów serwerowych oraz lepszej optymalizacji, aby sprostać rosnącym wymaganiom.

3. Dostosowanie do potrzeb użytkownika

Choć wiele systemów CMS oferuje szeroki zakres funkcji, często konieczne jest ich dostosowanie do specyficznych potrzeb organizacji. Obejmuje to integrację z innymi systemami, dostosowanie interfejsu użytkownika czy personalizację funkcji.

Systemy zarządzania treścią (CMS) są niezbędnym narzędziem w dzisiejszym cyfrowym świecie, umożliwiającym firmom i organizacjom efektywne zarządzanie treściami i ich publikowanie w internecie. Niezależnie od branży, CMS oferują elastyczność, skalowalność oraz narzędzia umożliwiające personalizację treści i zarządzanie użytkownikami. Jednakże, skuteczne wdrożenie i zarządzanie systemami CMS wymaga odpowiedniego podejścia do kwestii bezpieczeństwa, skalowalności oraz integracji z innymi narzędziami biznesowymi.

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.

Określenie funkcji celu oraz zbioru jej argumentów

Program GenMachine został napisany w zorientowanym obiektowo języku programowania C++. Jego głównym i niezmiennym elementem jest obiekt klasy GenMachine zdefiniowany w dołączanej statycznie bibliotece GenMachine.lib. Kod źródłowy obiektu klasy GenMachine podany jest w załączniku, w plikach machine.h i machine.cpp. Większość parametrów potrzebnych do pracy systemu, takich jak maksymalna liczba pokoleń, czy też prawdopodobieństwo mutowania, przekazywana jest za pomocą obiektu klasy IniFile (pliki: IniFile.h, IniFile.cpp) i za pośrednictwem pliku tekstowego GenMachine.ini, a zatem zmiany parametrów na potrzeby przeprowadzanych w tym rozdziale badań nie wymagają rekompilacji programu.

Musi jednak istnieć osobny egzemplarz programu dla każdej funkcji celu i każdego zbioru jej argumentów. Określenie nowej funkcji celu jest równoznaczne ze zdefiniowaniem funkcji wirtualnej o nazwie real Formula(real* x) , która jako argument otrzymuje wektor liczb rzeczywistych oraz zwraca liczbę rzeczywistą. Na przykład określenie jednej z funkcji testowych De Jonga:

$$f(x_1,x_2) = 100(x^2 - x_2)^2 + (1 - xi)^2$$

(4.1.)

jako funkcji celu do maksymalizacji dla GenMachine wymaga włączenia do programu następującego fragmentu kodu:

```
real MyGenMachine::Formula( real* x )
```

```
{

double dopasowanie;

dopasowanie = ( x[0] * x[0] – x[1] ) * ( x[0] * x[0] – x[1] );

dopasowanie *= 100;

dopasowanie += ( 1 – x[0] ) * ( 1 – x[0] );

dopasowanie *= -1;

return (real) dopasowanie;

};
```

Kolejnym krokiem jest określenie liczby zmiennych decyzyjnych oraz dziedziny funkcji. Dokonuje się tego przez zdefiniowanie stałych VARS i GM_DOMAIN. Dla zdefiniowanej powyżej funkcji będzie to następujący fragment kodu:

```
#define VARS 2

#define
GM_DOMAIN
                              \

minmax[0] = (real) -2.048      ;    minmax[1]     = (real)
2.048  ;                          \

minmax[2] = (real) -2.048      ;    minmax[3]     = (real)
2.048  ;
```

Tablica minmax zawiera dokładnie dwa razy więcej elementów niż wynosi liczba zmiennych decyzyjnych, ponieważ jej elementy określają kolejno dolną granicę pierwszej zmiennej, górną granicę pierwszej zmiennej, następnie dolną i górną granicę drugiej zmiennej, itd.

W zależności od potrzeb program GenMachine może operować na zmiennych rzeczywistych siedmiocyfrowych z zakresu od 3,4 •

10^{-38} do 3,4 • 10^{38} , wtedy typ zmiennej real odpowiada standardowemu typowi C++ float, lub też piętnastocyfrowych z zakresu od 1,7 • 10^{-308} do 1,7 • 10^{308} , wtedy typ real jest zdefiniowany jako double.

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.

Implementacja mechanizmu dynamicznej zmiany rozkładu prawdopodobieństwa wyboru metody mutowania

Działanie idealnego algorytmu genetycznego cechuje szerokie spektrum poszukiwań w początkowych fazach procesu ewolucji oraz dokładne dostrajanie lokalne w fazach końcowych, kiedy znane jest już otoczenie optimum.

Ze względu na różne role jakie spełniają poszczególne operatory mutowania, korzystną wydaje się być operacja zastąpienia stałego rozkładu prawdopodobieństwa wyboru jednego z operatorów mutowania wersją dynamiczną tegoż rozkładu. Dla oszczędności czasu procesora wraz ze wzrostem wartości numeru pokolenia t, dokonuje się liniowej zmiany rozkładu prawdopodobieństwa. Początkowe oraz końcowe wartości prawdopodobieństwa wyboru wszystkich używanych operatorów mutacji podawane są jako parametry pracy systemu.

Domyślne wartości tych parametrów dla mutacji równomiernej,

brzegowej i nierównomiernej w przedstawionym dalej programie GenMachine wynoszą odpowiednio:

PMTYPE1ST ART = 0.35

PMTYPE1END = 0.15

PMTYPE2START = 0.35

PMTYPE2END = 0.15

PMTYPE3 ST ART = 0.3

PMTYPE3END = 0.7

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.