

Міністерство освіти і науки України
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання практичних робіт
з дисципліни
“Мультиагентні системи”
для студентів спеціальності
7.080402 “Інформаційні технології проектування”
усіх форм навчання

2008

Методичні вказівки до виконання практичних робіт з дисципліни “Мультиагентні системи” для студентів спеціальності 7.080402 “Інформаційні технології проектування” / С.О. Субботін, О.О. Олійник, А.О. Олійник. – Запоріжжя: ЗНТУ, 2008. – 18 с.

Автори: Сергій Олександрович Субботін,
Олексій Олександрович Олійник,
Андрій Олександрович Олійник

Рецензент: Г.В. Табунщик, к.т.н., доцент

Відповідальний
за випуск: А.В. Притула, к.т.н., доцент

Затверджено
вченою радою інституту
інформатики та радіоелектроніки

Протокол № 3
від 17.09.2008 р.

Затверджено
на засіданні кафедри
“Програмні засоби”

Протокол № 1
від 28.08.2008 р.

ЗМІСТ

Вступ.....	4
1 Практична робота № 1 Формалізація мультиагентних систем ..	5
1.1 Мета роботи	5
1.2 Основні теоретичні відомості.....	5
1.3 Завдання до роботи	6
1.4 Зміст звіту	7
1.5 Контрольні питання	7
2 Практична робота № 2 Розробка Web-агента	8
2.1 Мета роботи	8
2.2 Основні теоретичні відомості.....	8
2.2.1 Основні поняття про програмні агенти	8
2.2.2 Властивості програмних агентів.....	8
2.2.3 Класифікація агентів	10
2.2.4 Розумні агенти.....	11
2.2.5 Web-агент	11
2.3 Завдання до роботи	14
2.4 Зміст звіту	16
2.5 Контрольні питання	17
Література	18

ВСТУП

Дане видання призначене для вивчення та практичного освоєння студентами усіх форм навчання основ побудови мультиагентних систем.

Відповідно до графіка студенти перед виконанням практичної роботи повинні ознайомитися з конспектом лекцій та рекомендованою літературою.

Для одержання заліку по кожній роботі студент здає викладачу цілком оформлений звіт з текстами розроблених програм (якщо треба за завданням), який супроводжується файлами програм, що виконуються (якщо треба за завданням), файлами даних (якщо треба за завданням) і текстом звіту на електронно-цифровому носії.

Звіт має містити:

- титульний аркуш;
- мету, варіант і завдання роботи;
- лаконічний опис теоретичних відомостей;
- текст програми, що обов'язково містить коментарі (якщо треба за завданням);
- вхідні та вихідні дані програми (якщо треба за завданням);
- змістовний аналіз отриманих результатів та висновки.

Звіт виконують на білому папері формату А4 (210 x 297 мм). Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші скріплюють за допомогою канцелярських скріпок. Для набору тексту звіту використовують редактор MS Word: шрифт Times New Roman, 12 пунктів. Міжрядковий інтервал: полуторний – для тексту звіту, одинарний – для листингів програм, таблиць і роздруківок даних.

Під час співбесіди студент повинний виявити знання про мету роботи, по теоретичному матеріалу, про методи виконання кожного етапу роботи, по змісту основних розділів розробленого звіту з демонстрацією результатів на конкретних прикладах. Студент повинний вміти правильно аналізувати отримані результати. Для самоперевірки при підготовці до виконання і здачі роботи студент повинний відповісти на контрольні питання, приведені наприкінці опису відповідної роботи. Загальний залік студент одержує після виконання і здачі останньої роботи.

1 ПРАКТИЧНА РОБОТА № 1

ФОРМАЛІЗАЦІЯ МУЛЬТИАГЕНТНИХ СИСТЕМ

1.1 Мета роботи

1.1.1 Вивчити принципи та правила формального подання мультиагентних систем.

1.1.2 Навчитися виконувати формальне подання мультиагентних систем.

1.2 Основні теоретичні відомості

Мультиагентну систему (MAC) в загальному випадку можна подати у вигляді множини з трьох елементів: Агенти, Середовище, Зв'язки між Середовищем та Агентами:

$$MAC = \langle \text{Агенти}, \text{Середовище}, \text{Зв'язки} \rangle.$$

Кожен Агент_i описується за допомогою множини чотирьох елементів: Стан_i , Вхід_i , Вихід_i , Процес_i :

$$\text{Агент}_i = \langle \text{Стан}_i, \text{Вхід}_i, \text{Вихід}_i, \text{Процес}_i \rangle,$$

де Стан_i – це множина змінних, що повністю визначають агента; Вхід_i та Вихід_i – підмножина Стану_i , елементи яких пов'язані із середовищем.

Процес_i – автономний метод, що виконує відповідні зміни над Станом . “Автономні метод” означають, що даний метод викликається без будь-якого зовнішнього впливу.

Середовищем являється множина з двох елементів:

$$\text{Середовище} = \langle \text{Стан}_c, \text{Процес}_c \rangle.$$

Важливою особливістю такого подання Середовища є те, що Середовище є саме по собі активним, оскільки воно містить свій власний Процес_c , котрий може змінювати Стан_c незалежно від агентів, що входять в це Середовище .

Вхід_i та Вихід_i різноманітних агентів пов'язані з елементами Стану_c , але середовище не відрізняє, які з елементів Стану_c знаходяться в залежності з ними. Відсутність Входу та Виходу у Середовища означає, що воно, на відміну від агентів, є необмеженим. Якби в Середовище були додані Вхід та Вихід , то це означало б, що середо-

вище обмежене та є фактично агентом високого рівня. Таким чином, може моделюватися взаємодія агентів високого та низького рівнів.

Наприклад, формальне подання одного з мультиагентних методів інтелектуальної оптимізації, методу бджолиної колонії (Bee Colony Optimization, BCO), буде наступним:

$BCO = \langle \text{Bees}, \text{Search Space}, \text{Waggle Dance} \rangle$.

$\text{Bee} = \langle \{ \text{Зайнятий Фуражир}, \text{Незайнятий Фуражир}, \text{Скаут} \}, \langle \{ \text{Solution}_{\text{parent}} \}, \{ \text{Solution}_{\text{bees}} \}, \{ \emptyset \} \rangle, \langle \{ \text{Solution}_{\text{parent}}' \}, \{ \text{Solution}_{\text{parent}}, \emptyset \}, \{ \text{RandSolution} \} \rangle, \{ \text{Розробка}, \text{Waggle Dance} \}, \{ \text{Вербування} \}, \{ \text{Пошук випадкових рішень}, \text{Waggle Dance} \} \rangle$.

$\text{Search Space} = \langle \emptyset \rangle$.

У відповідності з наведеною концепцією подання мультиагентних систем можуть створюватися їхні різноманітні типи, при цьому така концепція організації систем може бути використана не тільки в програмних (штучних) системах, а й в штучних системах. Така класифікація мультиагентних систем подана на рис. 1.1.

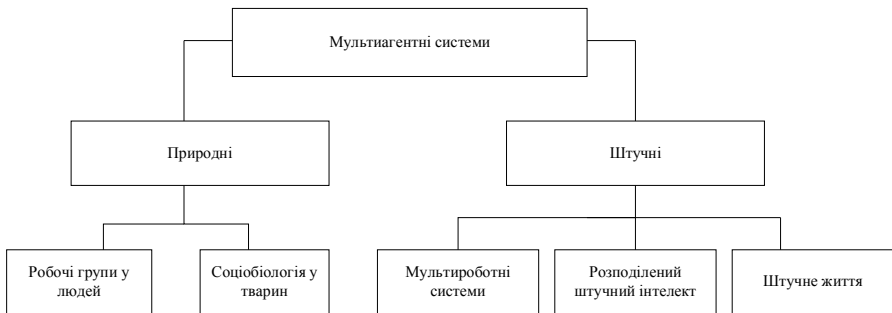


Рисунок 1.1 – Класифікація мультиагентних систем

1.3 Завдання до роботи

1.3.1 Ознайомитися з літературою та основними теоретичними відомостями за темою роботи.

1.3.2 Вручну на папері відповідно до варіанту виконати формальне подання мультиагентного методу інтелектуальної оптимізації:

– варіант 1: метод мурашиних колоній (Ant Colony Optimization, ACO);

– варіант 2: метод оптимізації з використанням рою частинок (Particle Swarm Optimization, PSO).

1.3.3 Оформити звіт з роботи.

1.3.4 Відповісти на контрольні питання.

1.4 Зміст звіту

1.4.1 Тема та мета роботи.

1.4.2 Короткі теоретичні відомості.

1.4.3 Виконане формальне подання відповідного мультиагентного методу інтелектуальної оптимізації.

1.4.4 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

1.5 Контрольні питання

1.5.1 Назвіть елементи, які використовуються при описі мультиагентних систем?

1.5.2 Назвіть елементи, що використовуються для подання окремого агенту?

1.5.3 Яким чином забезпечується зв'язок *Агентів* з *Середовищем*?

1.5.4 Чим є *Середовище* відносно *Агентів*?

1.5.5 Порівняйте агентно-орієнтоване та об'єктно-орієнтоване програмування.

1.5.6 Назвіть класифікацію мультиагентних систем.

1.5.7 До якого розділу наведеної класифікації мультиагентних систем відносяться мультиагентні методи інтелектуальної оптимізації?

1.5.8 Наведіть переваги та недоліки мультиагентних систем.

1.5.9 Які основні галузі використання мультиагентних систем?

1.5.10 Перечисліть методи оптимізації, в яких використовується мультиагентний підхід.

2 ПРАКТИЧНА РОБОТА № 2 РОЗРОБКА WEB-АГЕНТА

2.1 Мета роботи

2.1.1 Вивчити основні принципи роботи програмних агентів.

2.1.2 Навчитися створювати Web-агента, який взаємодіє із певними службами в середовищі Internet.

2.2 Основні теоретичні відомості

2.2.1 Основні поняття про програмні агенти

Агент – це апаратна або програмна сутність, здатна діяти в інтересах досягнення цілей, поставлених перед нею користувачем.

В межах мультиагентної парадигми програмні агенти являють собою автономні компоненти, що діють від імені користувача.

Агентів також називають розумними агентами (Intelligent agent), тому що розумність є ключовим чинником при їхньому створенні. Хоча агенти можуть приймати різні форми, але вони всі реалізують якісь корисні функції за людину. Агент може одержати завдання, яке необхідно виконати, і приймати потрібні рішення в процесі комунікації з іншими агентами (можливо, навіть із агентами, що представляють інтереси інших користувачів).

2.2.2 Властивості програмних агентів

Агенти можуть мати одну або декілька властивостей, наведених в табл. 2.1.

Однією з основних властивостей, які асоціюються із програмними агентами, є автономність (Autonomy). Агент вважається автономним, якщо він здатний діяти без прямого управління людиною. Поняття автономності включає також наявність в агента декількох цілей, яких він може досягти. Наявність задач також сприяє тому, що агент одержує можливість планувати.

Таблиця 2.1 – Властивості програмних агентів

Властивість	Визначення
Автономність	Можливість діяти незалежно від користувача
Адаптивність	Здатність до навчання під час роботи
Комунікативність	Здатність до комунікації з користувачем або іншими агентами
Здатність до співробітництва	Працює з іншими агентами для досягнення мети
Персоніфікованість	Поводиться природно (проявляє емоції)
Мобільність	Можливість переміщатися по навколишньому середовищу

Агент є адаптивним (Adaptivity), якщо він може змінювати своє поведіння на підставі досвіду. Він повинен учитися й робити висновки залежно від стану навколишнього середовища й своїх знань.

Ще однією важливою характеристикою агентів є здатність до комунікації (Communicative). Агент повинен уміти спілкуватися з користувачем, щоб визначати свої завдання або ідентифікувати початкову інформацію. Крім того, агентові необхідно спілкуватися з навколишнім середовищем. Наприклад, пошуковий агент, що знаходить цікаві Web-сторінки, повинен уміти спілкуватися за допомогою протоколу HTTP, щоб підключатися до серверів і одержувати від них інформацію (шукані сторінки).

Агент здатний до співробітництва (Collaborative), якщо він може спілкуватися з іншими агентами для вирішення своїх завдань. Наприклад, агенти, які діють від імені користувачів на аукціонах, можуть брати участь в укладанні спільних угод, орієнтуючись на інформацію, отриману від користувачів. Подібні агенти звичайно входять у системи з більшою кількістю агентів, тому що для співробітництва потрібна безліч агентів.

Для деяких агентів найбільш істотною є здатність до персоналізації (Personality). Програми з їхнім застосуванням звичайно використовуються в розважальних цілях. Тому, якщо персоналізація агента буде змінювати його залежно від переданої інформації, користувач зможе зрозуміти те, що агент намагається повідомити. Наприклад, коли оповідач про що-небудь розповідає й вираз його особи при цьому

не змінюється, слухачеві може бути важко його зрозуміти. Інформація, закодована в людських почуттях, дозволяє одержати додаткові відомості, тому можливість персоналізації може бути важливою для агентів певного класу.

Деякі агенти мають мобільність (Mobility), або можливість переміщення по навколишньому середовищу. Ця властивість є дуже корисною при створенні певних програм. Властивість мобільності використовується у програмних агентах, що одержують інформацію з віддаленої бази даних. При звичайних умовах хост приймає інформацію від бази даних, збирає потрібні відомості, а потім фільтрує їх за допомогою критеріїв, заданих користувачем. При одержанні мобільності агент може бути спрямований у віддалену базу даних, щоб автоматично відтфільтрувати результати, а потім повернути тільки ті, які необхідні користувачеві. Це дозволяє як заощадити на пропускну здатності каналу, так і спростити архітектуру, особливо в тому випадку, якщо два хости можуть бути відз'єднані один від одного.

Агент необов'язково повинен мати всі властивості. У більшості випадків використовуються всього дві.

2.2.3 Класифікація агентів

Незалежні агенти можна класифікувати наступним чином.

1. Агенти штучного життя.
2. Програмні агенти.
 - 2.1. Агенти для завдань.
 - 2.2. Агенти для проведення дозвілля.
 - 2.3. Віруси.
3. Інші агенти.

Агент, використання якого залежить від завдання, застосовується для вирішення певної проблеми. Прикладом може служити пошуковий агент у мережі Internet. Крім пошуку за заданими критеріями, агент також стежить за тим, які зі знайдених посилань відкриє користувач. Це дозволяє створити шкалу цінностей пошуку, що сприяє кращому сортуванню результатів, а також прискорює одержання користувачем потрібного результату. Пошуковий агент може працювати у фоновому режимі, повторюючи пошук, щоб знайти й надати користувачеві нову інформацію.

Пошукові агенти володіють не тільки автономністю, але й адаптивністю, оскільки вони вивчають найцікавіші для користувача теми, ґрунтуючись на його реакції (тобто посиланнях, які відкриває користувач).

Іншим прикладом може служити агент, що використовується для роботи на аукціоні. Він здатний діяти від імені користувача, купуючи товари в мережі Internet по найнижчій ціні. Він може робити це в співробітництві з іншими агентами. Знайшовши агентів, які намагаються купити схожі товари, можна зробити спільну оптову покупку групи товарів за зниженою ціною.

Розважальні агенти корисні для взаємодії у віртуальному світі або подання персонажа в якості інтерфейсу користувача.

Вірус по-іншому можна назвати небезпечним мобільним програмним агентом. Хоча вірус не обов'язково повинен бути адаптивним, він є автономним і мобільним. Замість того щоб використовувати спеціальний протокол для переміщення по мережі, вірус застосовує стандартні протоколи й найчастіше протокол передачі пошти.

2.2.4 Розумні агенти

Хоча зробити агента розумним досить складно, існує безліч методів, які можна використовувати для того, щоб додати агентові здатність до прийняття розумних рішень. До таких методів належать:

- нейронні мережі;
- системи, засновані на знаннях;
- дерева рішачючих правил;
- еволюційні методи;
- нечітка логіка;
- алгоритми кластеризації.

2.2.5 Web-агент

Завдання Web-агенту полягає в тому, щоб взаємодіяти з певними службами у мережі Internet (наприклад, із службами новин) і збирати інформацію на підставі критеріїв, заданих користувачем. Після збирання інформації Web-агент надає її користувачеві в тому порядку, який найбільш відповідає запиту (починаючи з даних, що мають найвищий рейтинг за умовами пошуку).

Web-агент використовує стандартні протоколи й передає користувачеві інформацію через звичайний браузер. Для спілкування з користувачем агент містить HTTP-сервер, для зв'язку із зовнішніми серверами є HTTP-клієнтом, а для одержання новин функціонує як клієнт протоколу NTTP (рис. 2.1).

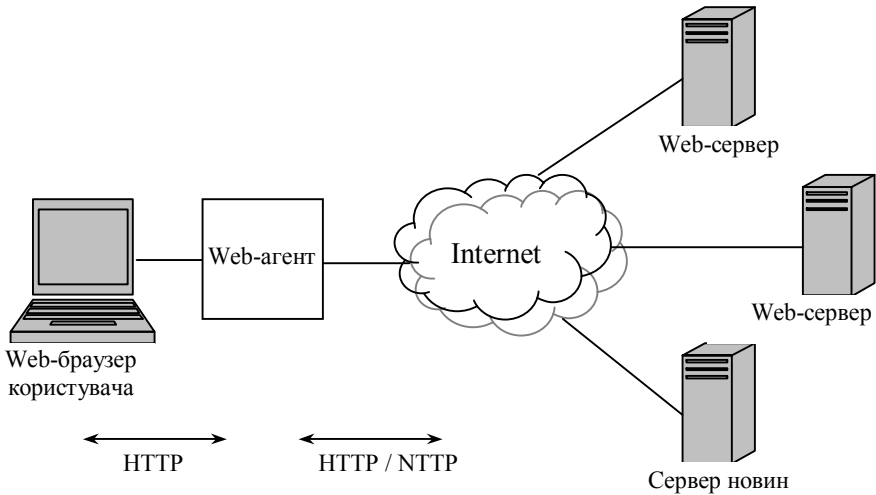


Рисунок 2.1 – Архітектура Web-агента

Web-агент буде відігравати роль додатка, що здійснює взаємодію між користувачем і мережею Internet при читанні новин. Користувач змінює параметри агента за допомогою простого файлу конфігурації. У ньому користувач указує Web-сторінки для моніторингу (для видачі повідомлення у випадку їхньої зміни) і групи новин для перегляду й визначає специфічні критерії пошуку. Коли агент знаходить об'єкти, які відповідають заданому критерію, він зберігає їх і видає користувачеві у вигляді Web-сторінки, посилання на якій відсортовані в потрібному порядку. Користувач може клацнути по посиланню на певний Web-сайт або рядку новин і перейти до потрібної статті.

Завдання агента полягає в тому, щоб спростити читання новин, відфільтрувати непотрібну інформацію й подати новини в порядку, при якому найцікавіші дані розташовані на першому місці.

Web-агент є дуже простим і не має здатності до навчання. Він володіє двома властивостями: автономністю (здатний працювати у

фоновому режимі без прямого керування з боку користувача) і здатністю до комунікації із зовнішніми серверами при зборі інформації.

Сенсорами агента є стандартні протоколи, які дозволяють йому збирати потрібну інформацію за заданими критеріями пошуку. Для передачі результатів пошуку користувачеві використовується протокол HTTP. Потім користувач підключається до Web-агента, як до будь-якого іншого Web-сервера.

Для отримання інформації з середовища Internet Web-агент виступає в якості HTTP-клієнта, який реалізує набір інтерфейсів для взаємодії з Internet серверами, наприклад, із серверами новин. Ці інтерфейси дозволяють додатку з'єднуватись із сервером новин, задавати групи новин за певними критеріями, переходити до заголовку статті, зчитувати всю статтю, аналізувати повідомлення, пропускати поточну новину та переривати зв'язок із сервером новин.

Протокол HTTP є інтерактивним протоколом, який повністю базується на ASCII-командах. Нижче наведено приклад роботи із HTTP-сервером новин yellow.geeks.org за допомогою Telnet. Дані, що вводить користувач, виділені жирним текстом.

```
root@plato /root]# telnet yellow.geeks.org 119
S: 201 plato.mtjones.com DNEWS Version 5.5dl, SO, posting OK
C: list
S: 215 list.of newsgroups follows
S: control 2 3 y
S: control.cancel 2 3 y
S: my.group 10 3 y
S: new.group 6 3 y
S: .
C: group my.group
S: 211 8 3 10 my.group selected
C: article 3
S: 220 3 <3C36AF8E. 1BD3047E@mtjones.com> article retrieved
S: Message-ID: <3C36AF8E. 1BD3047E@mtjones.com>
S: Date: Sat, 05 Jan 2002 00:47:27 -0700
S: From: "M. Tim Jones" <mtj@mtjones.com>
S: X-Mailer: Mozilla 4.74 [en] (Win98; U)
S: X-Accept-Language: en
S: MIME-Version: 1.0
S: Newsgroups: my.group
S: Subject: this is my post
S: Content-Type: text/plain; charset=us-ascii
S: Content-Transfer-Encoding: 7bit
S: NNTP-Posting-Host: sartre.mtjones.com
```

```

S: X-Trace: plato.mtjones.com 1010328764 sartre.mtjones.com
(6 Jan 2002 07:52:44 -0700)
S: Lines: 6
S: Path: plato.mtjones.com
S: Xref: plato.mtjones.com my.group:3
S:
S:
S: Hello
S:
S: This is my post.
S:
S:
S: .
C: date
S: 111 20020112122419
C: quit
S: 205 closing connection - goodbye!

```

У наведеному прикладі показано, як з'єднання з NNTP-сервером створюється за допомогою Telnet-клієнта. Сервер новин відповідає вітанням, у якому вказується тип сервера й інша подібна інформація. Тепер через з'єднання можна давати команди. Існує два базових типи відповідей, які можна чекати від сервера – з одного рядка й з декількох рядків. Щоб ідентифікувати кінець відповіді, протокол NNTP, як і протокол SMTP, застосовує символ “.” у рядку, (наприклад, відповідь на команди list і article).

2.3 Завдання до роботи

2.3.1 Ознайомитися з основними теоретичними відомостями за темою роботи.

2.3.2 Розробити програмного Web-агента, який взаємодіє із певними службами в Internet та збирає інформацію на основі критеріїв, що задані користувачем.

Програмний Web-агент має відповідати вимогам, наведеним в п. 2.3.2.1 – 2.3.2.5.

2.3.2.1 За допомогою протоколу HTTP Web-агент має перевіряти зміни заданих web-сайтів, на яких відображається потрібна інформація (наприклад, новини). При цьому зв'язок з сайтами організується за допомогою інтерфейсу сокетів. Протокол мережного рівня – IP, протокол транспортного рівня – TCP, протокол приклад-

ного рівня – HTTP (порт 80). Перевірку на зміну сайтів з новинами рекомендовано виконувати шляхом перевірки розміру файлу, що пропонується скачати.

2.3.2.2 За допомогою протоколу HTTP Web-агент має відбирати на вказаних серверах потрібну інформацію (наприклад, на сайті новин, – статті), яка відповідає критеріям, що задані користувачем. Критеріями можуть бути: група новин та окремі слова, що мають зустрічатися в новині. Зв'язок з сервером новин має бути організований за допомогою інтерфейсу сокетів. Протокол мережного рівня – IP, протокол транспортного рівня – TCP, протокол прикладного рівня – HTTP (порт 119). При цьому врахувати наступні особливості серверів новин:

- після підключення до серверу має бути повернуте повідомлення, що вміщує код 201 у випадку вдалого підключення;

- для встановлення заданої групи новин необхідно відправити команду: `group назва_групи`. При цьому в результаті успішності має бути повернуте повідомлення, що вміщує наступну послідовність: код 211, загальну кількість повідомлень даної групи, номер першого та останнього повідомлення;

- для завантаження заголовку повідомлення необхідно відправити команду: `head номер_повідомлення`. При цьому, якщо все було виконано успішно, то повертається код 221 та заголовок повідомлення;

- для завантаження усього повідомлення необхідно відправити серверу команду: `article номер_повідомлення`. Якщо повідомлення успішно передано, то повертається код 220 та саме повідомлення;

- для ідентифікації кінця відповіді сервера використовується рядок з символом “.”

2.3.2.3 Web-агент має надавати користувачу Web-інтерфейс. При цьому мають бути передбачені такі можливості:

- можливість перегляду поточної конфігурації (тобто конфігурацію можна переглянути, прописавши в web-браузері посилання наступного типу: `http://192.168.115.2:8080/conf.html` – в випадку, коли web-агент працює на комп'ютері з ip-адресою 192.168.115.2 на порті 8080);

- можливість перегляду списку http-сайтів та http-серверів новин, на яких відбулися зміни (наприклад, `http://192.168.115.2:8080/index.html`). Тобто має бути відображена ін-

формація в вигляді двох таблиць: таблиця http-сайтів, що змінилися, та таблиця повідомлень на серверах новин. При цьому для цих повідомлень має бути відображена група, до якої вони відносяться, а також – задані слова, що були знайдені в цих повідомленнях. Послідовність відображення новин має відповідати їхній відповідності критеріям користувача;

- можливість перегляду повідомлення з серверу новин у web-браузері користувача;

- надати можливість користувачу вказати http-сервери, що було переглянуто, тобто які вже не треба виводити у таблиці до їх наступної зміни.

2.3.2.4 Робота зі збору інформації про стан серверів та робота із користувачем мають не заважати одна одній.

2.3.2.5 Конфігурація роботи Web-агента має розміщуватися в окремому конфігураційному файлі, в якому можна вказати наступні параметри:

- http-сайти з новинами;
- сервери новин;
- групи новин та відповідні їм слова, що мають знаходитися в середині повідомлень.

2.3.3 Виконати тестування розробленого програмного агента.

2.3.4 Проаналізувати роботу створеного програмного агента. Виявити його переваги та недоліки.

2.3.5 Оформити звіт з роботи.

2.3.6 Відповісти на контрольні питання.

2.4 Зміст звіту

2.4.1 Тема та мета роботи.

2.4.2 Короткі теоретичні відомості.

2.4.3 Текст програми (із коментарями), що реалізує роботу Web-агенту.

2.4.4 Результати тестування Web-агента.

2.4.5 Копії екранних форм, що реалізують взаємодію користувача із програмним агентом як складової частини інтелектуальної системи підтримки прийняття рішень.

2.4.6 Результати тестування Web-агенту як складової частини інтелектуальної системи підтримки прийняття рішень (отримані із Internet дані, поповнені факти бази знань тощо).

2.4.7 Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

2.5 Контрольні питання

2.5.1 Що являє собою програмний агент?

2.5.2 Поясніть та проаналізуйте властивості агентів: автономність, адаптивність, комунікативність, здатність до співробітництва, персоніфікованість, мобільність.

2.5.3 З якою метою застосовуються агенти, використання яких залежить від завдань?

2.5.4 Що таке розважальні агенти?

2.5.5 Проаналізуйте комп'ютерні віруси як програмні агенти.

2.5.6 Які методи використовуються для надання інтелектуальності програмним агентам?

2.5.7 Які протоколи використовує Web-агент?

2.5.8 Виконайте аналіз архітектури Web-агенту?

2.5.9 Якими властивостями володіє Web-агент?

2.5.10 Проаналізуйте протокол, який використовується для передачі користувачу результатів пошуку за допомогою Web-агенту.

2.5.11 Наведіть і поясніть схему потоків усередині Web-агенту?

2.5.12 Проаналізуйте основні функції обраного інструментарію розробки Web-агенту: внутрішня структура, параметри, основні змінні, їх призначення та використання.

ЛІТЕРАТУРА

1. Ferber J., Müller J.P. Influences and reactions: A model of situated multiagent systems // Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS-96). – Menlo Park: IEEE Computer Society Press, 1996. – P. 72-79.
2. Müller J.P. The Design of Intelligent Agents: A Layered Approach (Lecture Notes in Computer Science). – Berlin: Springer, 1996. – 227 p.
3. Gasser L., Hill R.W. Coordinated problem solvers // Annual Review of Computer Science, – 1990. – №4. – P. 203–253.
4. Hackman J.R. Groups That Work (and Those That Don't). – San Francisco, California: Jossey-Bass, 1990. – 544 p.
5. Parunak H.V.D. “Go to the ant”: Engineering principles from natural multi-agent systems // Annals of Operation Research – 1997. – №75. – P. 69–101.
6. Majchrzak A., Gasser L. Toward a conceptual framework for specifying manufacturing workgroups congruent with technological change // International Journal of Computer Integrated Manufacturing. – 1992. – №5. – P. 118–131.
7. Karaboga D. An Idea Based on Honey Bee Swarm for Numerical Optimization. – Technical report TR06. – Erciyes: Erciyes University Press, 2005. – 10 p.
8. Griffin D.R. Animal Minds. – Chicago: The University of Chicago Press, 2001. – 376 p.
9. Wilson E.O. The Insect Societies. – Cambridge, Massachusetts: The Belknap Press, 1974. – 562 p.
10. Agah A., Bekey G.A. Phylogenetic and Ontogenetic Learning in a Colony of Interacting Robots // Autonomous Robots. – 1997. – №4. – P. 85–100.
11. Джонс М.Т. Программирование искусственного интеллекта в приложениях. – М.: ДМК Пресс, 2004. – 312 с.
12. Олейник Ал.А. Сравнительный анализ методов оптимизации на основе метода муравьиных колоний // Комп'ютерне моделювання та інтелектуальні системи: Збірник наукових праць / За ред. Д.М. Пізи, С.О. Субботіна. – Запоріжжя: ЗНТУ, 2007. – С. 147–159.