

РАЗДЕЛ «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»

УДК 004.4'242

ЧЕРНОМУРОВА Л.О., к.ф.-м.н., доцент
ІВАНІН Д.О., к.ф.-м.н., доцент
САВИЦЬКА К.В., магістр

Дніпродзержинський державний технічний університет

ПОБУДОВА КІНЦЕВИХ АВТОМАТІВ З ВИКОРИСТАННЯМ ЙМОВІРНІСНИХ АЛГОРИТМІВ

Вступ. В останній час все ширше застосовується автоматне програмування, в межах якого поведінка програм описується за допомогою скінчених детермінованих автоматів. Такі автомати знаходять широке застосування для розпізнавання регулярних мов, при побудові дискретних пристроїв керування. Зокрема, теорія побудови компіляторів базується на ієрархії моделей автоматів.

Для багатьох задач автомати вдається будувати евристично. Наочним прикладом, в якому автоматична побудова логіки сутності зі складною поведінкою має вирішальне значення, є одна з класичних задач із області спільного використання ймовірнісних алгоритмів і кінцевих автоматів – задача про «Розумну мурашу» [1] та модифікації цієї задачі.

Постановка задачі. Наведемо найпростішу постановку задачі про «Розумну мурашу». Задано клітинне поле, по якому може пересуватися мураха. Поле має розміри $n \times n$, воно перебуває на поверхні двовимірного тора і на ньому знаходиться N яблук. В деяких клітинах поля розміщена їжа (яблуко).

На рис.1 зображено приклад поля з їжею, що розміщена в заповнених клітинах.

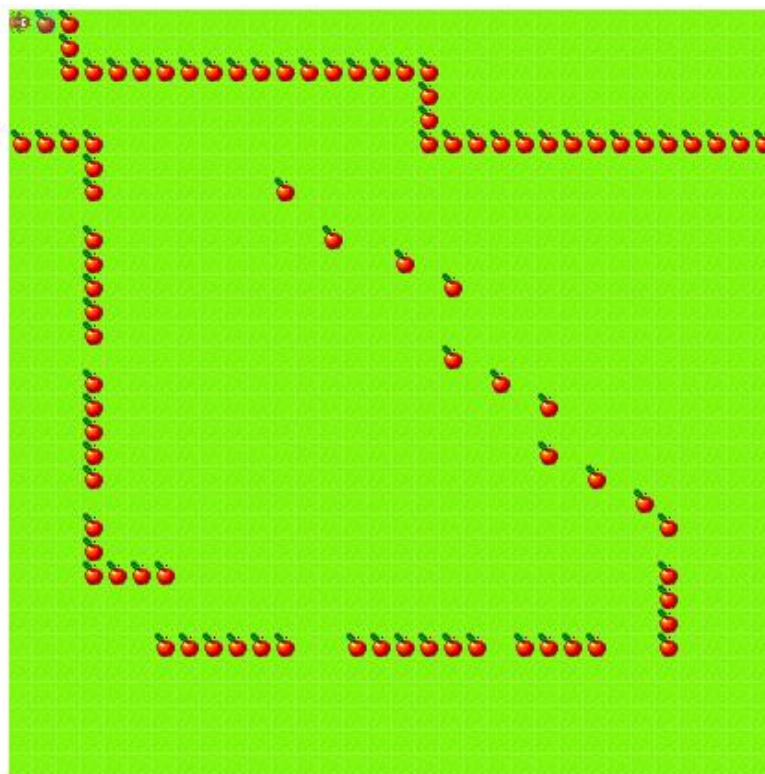


Рисунок 1 – Приклад поля для «Розумної мурахи»

З'їдена мурахою їжа не поповнюється, мураха живе на протязі всієї гри. Вона може пересуватися по будь-яким клітинам поля та обирати напрямок руху. За один хід вона здійснює одну з трьох дій:

- перейти в клітину, що знаходиться перед нею;
- повернути ліворуч;
- повернути праворуч.

Мурасі відомо, чи знаходиться їжа в клітині перед нею. Задача мурахи – почати рух з краю поля, за k ходів з'їсти якомога більше одиниць їжі (яблук), тобто відвідати максимальне число клітин, які містять їжу. Після проходження k ходів підраховується кількість їжі, що з'їдена мурахою. Це значення і є результатом гри.

Для даної задачі цільову функцію можна представити у наступному вигляді:

$$F = eaten - \frac{steps}{k}, \quad (1)$$

де $eaten$ – кількість з'їденої їжі (яблук);

$steps$ – номер кроку, на якому було з'їдено останнє яблуко;

k – задана кількість кроків.

Один із способів опису поведінки мурахи – кінцевий автомат з діями на переходах (автомат Мілі), у якого є одна вхідна змінна логічного типу (чи знаходиться їжа перед мурахою), а множина вихідних впливів складається з перерахованих дій. Приклад такого автомату представлено у вигляді графа переходів на рис.2.

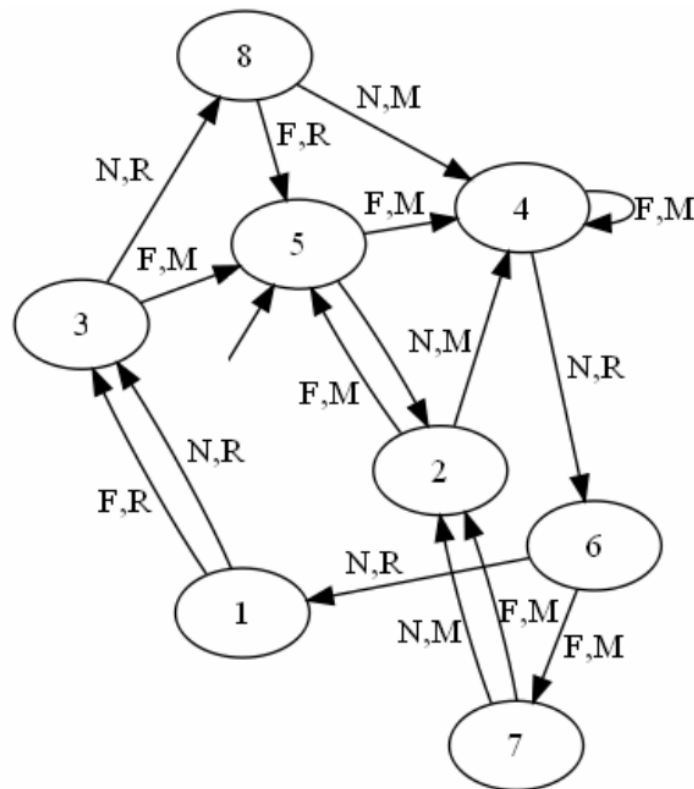


Рисунок 2 – Представлення автомата у вигляді графа переходів

Необхідно з використанням ймовірнісних алгоритмів побудувати автомат Мілі, близький до оптимального, що керує мурахою. Оптимальним вважається автомат, який має фіксоване число станів, а мураха, керована цим автоматом, з'їдає максимальне число яблук.

Результати роботи. В даній роботі для розв'язання поставленої задачі були застосовані такі ймовірнісні алгоритми: простий та клітинний генетичні алгоритми, алгоритм імітації відпалу [2]. Головну увагу було зосереджено на алгоритмі імітації відпалу та порівнянні його ефективності з генетичними алгоритмами.

Метод відпалу – техніка оптимізації, що використовує упорядкований випадковий пошук по аналогії з процесом утворення в речовині кристалічної структури при її охолодженні. Наведемо опис алгоритму імітації відпалу для поставленої задачі. На самому початку задається температура T (параметр, що визначає межу області пошуку рішення), початкове значення якої досить велике, але поступово знижується. Процес імітації починається з вибору деякого рішення задачі, для якого генерується випадковим чином автомат з діями на переходах та наперед заданою кількістю станів. Це рішення називається поточним. Для цього рішення (отриманого автомату) підраховується значення цільової функції (1). Наступним кроком є створення робочого рішення шляхом зміни поточного рішення. Зміна рішення здійснюється в такий спосіб: випадково змінюються початковий стан автомату і перехід автомату в інший стан з діями на переходах. Вибір початкового стану і переходу розігрується за допомогою випадкової величини, що має нормальний розподіл, як показано на рис.3.

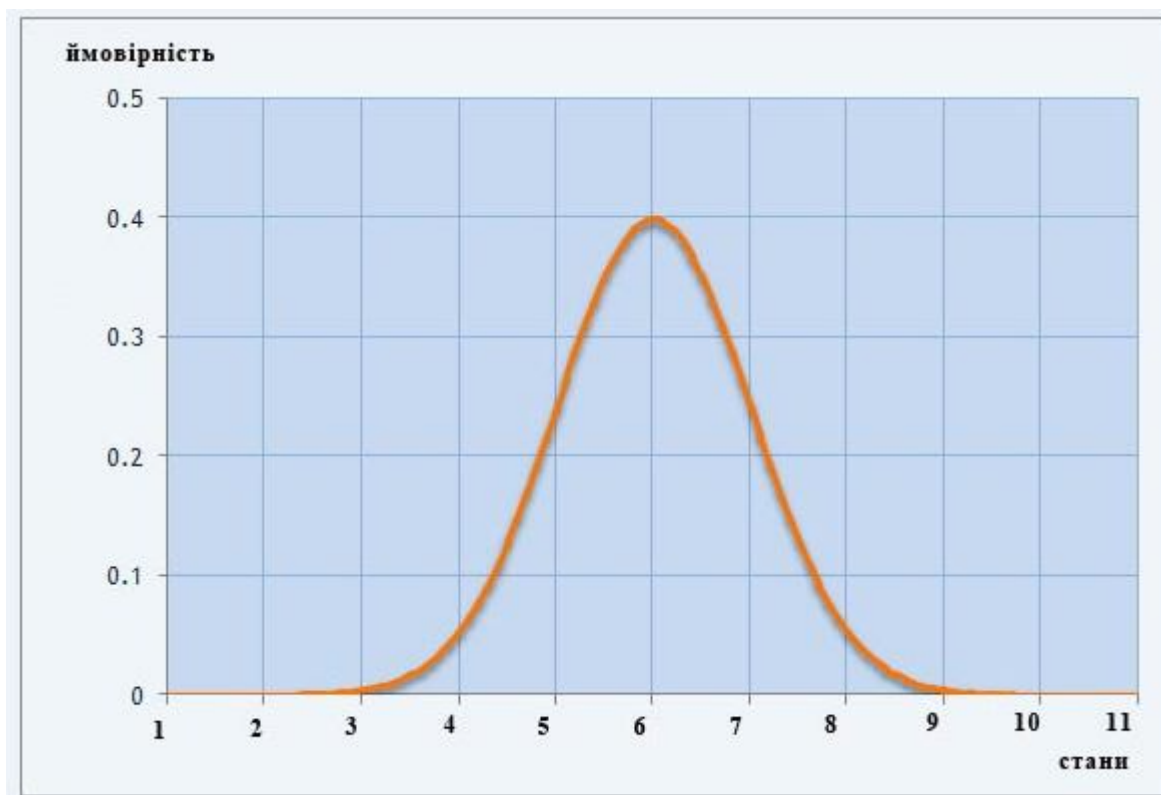


Рисунок 3 –Графік нормального розподілу

На осі x позначені стани автомата. Математичне очікування – це поточний стан автомата, а дисперсія розраховується за наступною формулою:

$$D = \frac{n}{2} T_{\text{пот}},$$

де $T_{\text{пот}}$ – значення поточної температури;

n – кількість станів автомата.

При високій температурі далекі від оптимальних рішення отримуються частіше, але здійснюється більш повний пошук рішень. Тобто при високій температурі існує більша ймовірність переходу в будь-який стан. При зниженні температури отримуємо рішення з кращим значенням цільової функції $F_{\text{пот}}$. Тобто діапазон станів, в який може перейти автомат, значно зменшується за рахунок зменшення значення дисперсії.

Для створеного робочого рішення підраховується значення цільової функції $F_{\text{роб}}$. Якщо значення цільової функції робочого рішення більше, ніж у поточного рішення, тоді робоче рішення стає поточним, а температура знижується. Якщо навпаки, тоді застосовується критерій допуску, згідно з яким випадковим чином розігрується величина в діапазоні $[0;1]$, і якщо ця величина менша, ніж $\exp\left(\frac{F_{\text{роб}} - F_{\text{пот}}}{T_{\text{пот}}}\right)$, тоді робоче рішення стає поточним.

Ймовірність прийняття робочого рішення у вигляді умови записується наступним чином:

$$P = \begin{cases} 1, & F_{\text{роб}} \geq F_{\text{пот}}, \\ \exp\left(\frac{F_{\text{роб}} - F_{\text{пот}}}{T_{\text{пот}}}\right), & F_{\text{роб}} < F_{\text{пот}}. \end{cases}$$

Робота алгоритму продовжується доти, доки температура не стане рівною нулю, або буде отримано прийнятне рішення.

Для розв'язання поставленої задачі була розроблена програма у середовищі Micro-soft Visual C# 2008 Express Edition на мові програмування C#. Проведені комп'ютерні дослідження (рис.4) показали, що генетичні алгоритми дозволяють отримати прийнятні результати досить швидко.

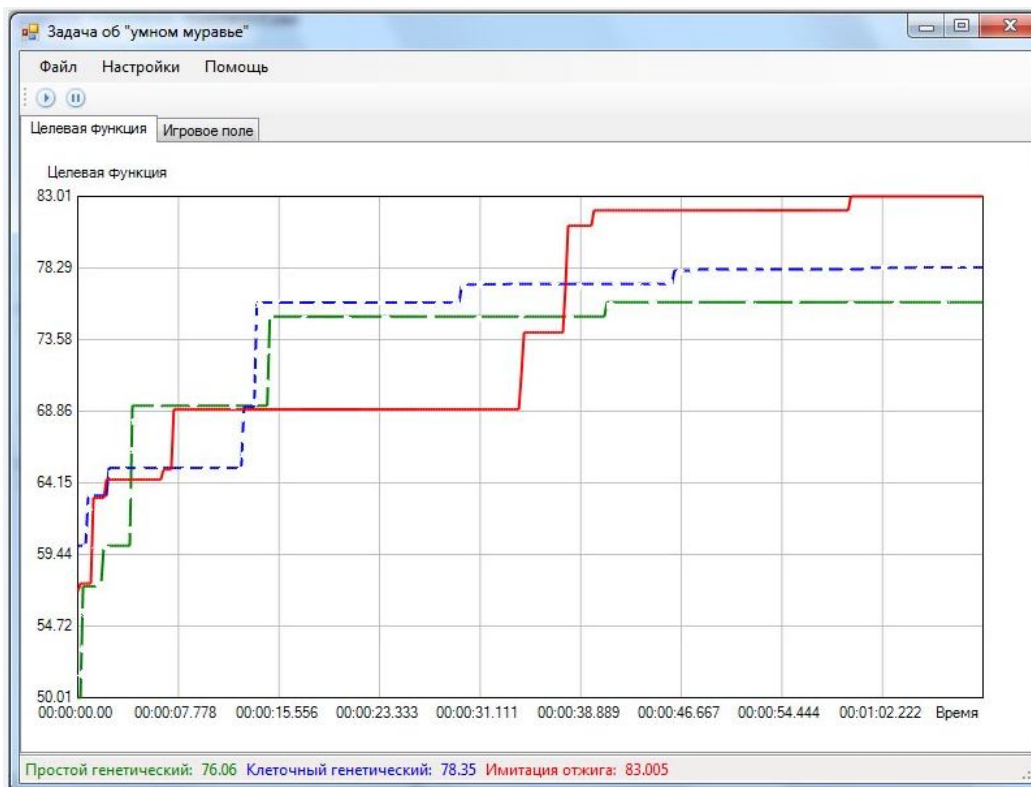


Рисунок 4 –Графіки порівнянь ймовірнісних алгоритмів

Зокрема, клітинний генетичний алгоритм на самому початку роботи, частіше за все, дає краще значення цільової функції у порівнянні з простим, але у подальшому процесі роботи в цілому генетичні алгоритми дають майже однаковий результат. Алгоритм імітації відпалу на початку процесу поступається генетичним алгоритмам, але під час подальшої роботи цей алгоритм в більшості випадків показує краще значення цільової функції, ніж генетичні алгоритми.

Висновки. Чисельні дослідження, проведені в даній роботі, показали, що алгоритм імітації відпалу приблизно на 5% ефективніший, ніж генетичні алгоритми для задачі про «Розумну мурашу». В даній роботі метод відпалу був застосований та показав також свою ефективність для розв'язання деяких модифікацій задачі про «Розумну мурашу», зокрема, «Мураха – 2» та «Мураха – 3».

Перевагою методу відпалу є властивість уникати "пасток" в локальних максимумах (мінімумах) функції, що оптимізується, і продовжувати пошук глобального максимуму (мінімуму). Ще однією перевагою є те, що навіть в умовах браку обчислювальних ресурсів для знаходження глобального максимуму (мінімуму) метод відпалу, як правило, дозволяє отримати досить непогане рішення. Метод відпалу і його модифікації є одним з найбільш ефективних методів випадкового пошуку оптимального рішення для великого класу задач. Результати роботи алгоритмів показали, що метод відпалу не програє генетичним алгоритмам, а для багатьох задач і виграє.

ЛІТЕРАТУРА

1. Бедный Ю.Д. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей» / Бедный Ю.Д., Шалыто А.А. // СПбГУ: ИТМО. – 2007. – [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/works/ant>
2. А.С.Лопатин. Метод отжига в задачах оптимизации / А.С.Лопатин. – 2004. – [Электронный ресурс] – Режим доступа: <http://www.math.spbu.ru/user/gran/students/cothesis.pdf>

Надійшла до редколегії 06.06.2012.

УДК 519.8, 004.4

КАДОЧНИКОВА Я.Е., к.ф.-м.н, доцент
ЛЕОНОВ А.В., студент

Днепропетровский государственный технический университет

ИСПОЛЬЗОВАНИЕ ВОЗМОЖНОСТЕЙ ASP.NET ДЛЯ ВИЗУАЛИЗАЦИИ РЕШЕНИЯ ОДНОЙ ЗАДАЧИ ОПТИМАЛЬНОГО РАЗБИЕНИЯ МНОЖЕСТВ

Введение. Интерес к непрерывным задачам оптимального разбиения множества (ОРМ) на непересекающиеся подмножества с целью минимизации некоторого критерия качества разбиения вызван тем, что к ним сводится достаточно широкий класс как теоретически, так и практически важных задач оптимизации, а также задач из других разделов прикладных наук. Отметим, что на протяжении последних тридцати лет Днепропетровской научной школой под руководством Е.М.Киселёвой получены существенные результаты в направлении развития теории ОРМ. Широкий обзор задач теории и практики, сводящихся к математическим моделям оптимального разбиения множеств, а также алгоритмы их решения приведены в монографии [1].

За все время исследований, связанных с решением задач ОРМ, было создано большое количество программных приложений, которые оставались недоступными широкому кругу пользователей. Поскольку задачи ОРМ имеют практическое значение, необходимым шагом является разработка web-сервиса, который смогут использовать сторонние клиенты для решения конкретных прикладных задач.

Постановка задачи. Необходимо разработать web-сервис и клиентское приложение, демонстрирующее работу web-сервиса, для решения непрерывной задачи оптимального разбиения множества на подмножества без ограничений с заданными координатами "центров" подмножеств. Математическая постановка данной задачи приведена в [1] и имеет следующий вид.

Найти

$$\min_{\{\Omega_1, \dots, \Omega_N\} \in \Sigma_{\Omega}^N} F(\{\Omega_1, \dots, \Omega_N\}),$$

где

$$F(\{\Omega_1, \dots, \Omega_N\}) = \sum_{i=1}^N \int_{\Omega_i} [c(x, \tau_i) + a_i] \rho(x) dx,$$

$$\Sigma_{\Omega}^N = \{ \{\Omega_1, \dots, \Omega_N\} : \bigcup_{i=1}^N \Omega_i = \Omega, \text{mes}(\Omega_i \cap \Omega_k) = 0, i \neq k, i, k = 1, \dots, N \},$$

функции $c(x, \tau_i)$ – действительные, ограниченные, измеримые по аргументу x , $x = (x^{(1)}, \dots, x^{(n)}) \in \Omega$, при любом фиксированном τ_i из Ω для всех $i = 1, \dots, N$; функция $\rho(x)$ – действительная, ограниченная, измеримая и неотрицательная на Ω ; a_1, \dots, a_N – заданные действительные неотрицательные числа.

Результаты работы. Для решения поставленной задачи было решено использовать популярную платформу .NET Framework [2]. Разработан проект WCF-сервиса Partitioning_Sets_Service и развернут на консольном приложении. На рис.1 приведена структура проекта:

- app.config – файл конфигураций веб-сервиса в формате XML;
- point.cs – класс для описания центра;
- area.cs – класс, который реализует решение задачи ОРМ;
- IPartitioningSetsService.cs – интерфейс, содержащий контракты для организации взаимодействия клиент-сервис;
- PartitioningSetsService.cs – класс, реализующий интерфейс IPartitioningSetsService;
- Program.cs – класс, который отвечает за создание экземпляра веб-сервиса, его запуск и остановку.

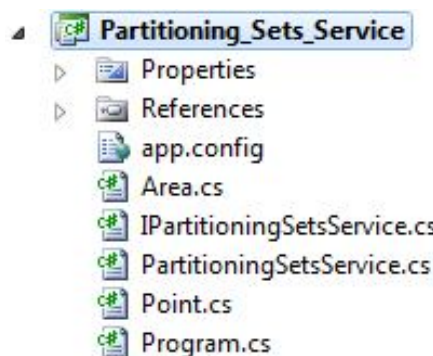


Рисунок 1 – Структура проекта WCF-сервиса

Для создания экземпляра web-сервиса выполняется запуск консольного приложения. Когда служба готова к обработке запросов, появляется сообщение «Service is running».

Когда с клиентского приложения приходит запрос, web-сервис его обрабатывает, вызывает нужные методы и отправляет ответ клиенту. Результаты по обработке каждого запроса отображаются в консольном приложении. Так на рис.2 можно увидеть время получения запроса, входные параметры и результирующую матрицу.

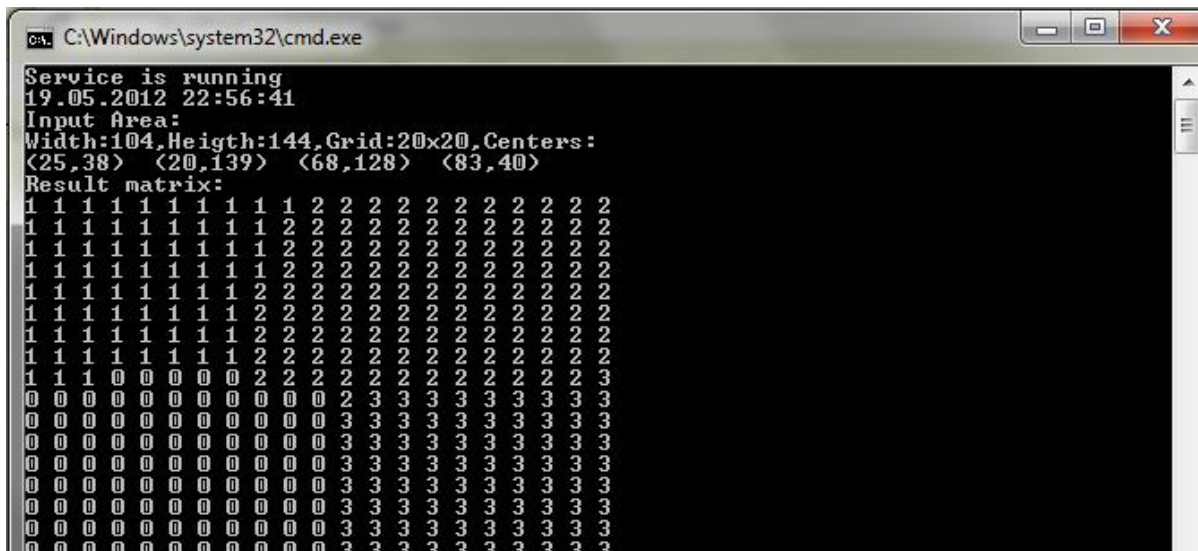


Рисунок 2 – Пример результата выполнения запроса

Проект `Service_Client` – клиентское приложение, которое разработано для наглядной демонстрации результатов работы web-сервиса. Для разработки клиентского приложения была выбрана технология ASP.NET Web Forms, которая позволила достичь следующих целей:

- интуитивно-понятный интерфейс;
- удобный ввод входных данных решаемой задачи;
- формирование результатов в понятном для пользователей виде;
- корректная обработка исключительных ситуаций, вызванных ошибочными действиями пользователя.

На рис.3 представлена структура проекта `Service_Client`.

Рассмотрим структуру проекта более детально:

- `Partitioning_Sets_Service` – ссылка на web-сервис. При добавлении ссылки к проекту автоматически формируются классы для технической реализации взаимодействия клиент-сервис;
- `Resources` – каталог для хранения дополнительных файлов проекта (изображений, анимации и т.п.);
- `Scripts` – каталог для хранения JScript библиотек;
- `Styles` – каталог для хранения CSS-стилей;
- `DrawImageClass.cs` – класс, реализующий работу с изображениями;
- `Global.asax` – содержит глобальные события приложения;
- `Web.config` – файл конфигураций приложения в формате XML;

- Task.aspx – страница ASP.NET Web Forms, которая содержит HTML разметку и классы с реализацией серверной логики;
- Task.aspx.cs – часть страницы ASP.NET Web Forms, которая реализует обработку событий страницы и ее элементов управления.

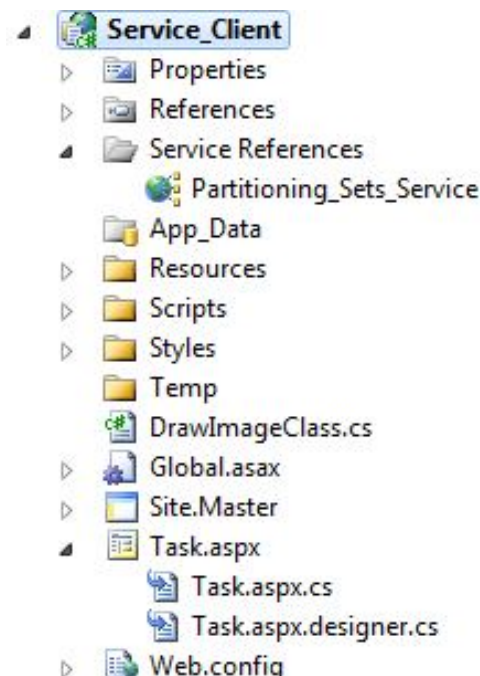


Рисунок 3 – Структура проекта клиентского приложения

Стартовая страница клиентского приложения в web-браузере выглядит, как показано на рис.4. Здесь пользователю предлагается загрузить карту, на которой будет проводиться решение задачи ОРМ. Если выбранный файл имеет некорректный формат, то пользователь увидит сообщение об ошибке.



Рисунок 4 – Стартовая страница консольного приложения

После удачной загрузки карты пользователю необходимо указать допустимую область на ней. Выбор области осуществляется мышью (рис.5), после чего на сервере выполняется обработка изображения и генерируется web-страница, содержащая выбранную область с поддержкой ввода центров двумя способами (рис.6):

- кликом мыши в нужной точке изображения;
- нажатием «Add Center» с последующим вводом координат центра.

После ввода всех параметров нажатием кнопки «Next» иницируется отправка данных разработанному web-сервису, который получает решение сформулированной

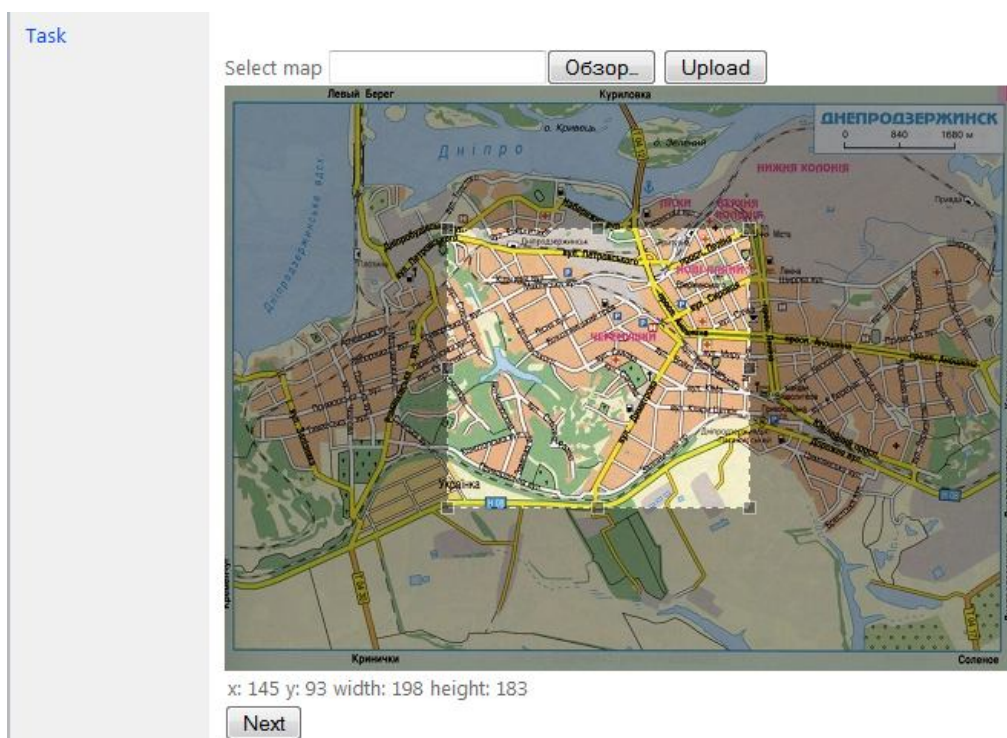


Рисунок 5 – Вид web-страницы при выборе области

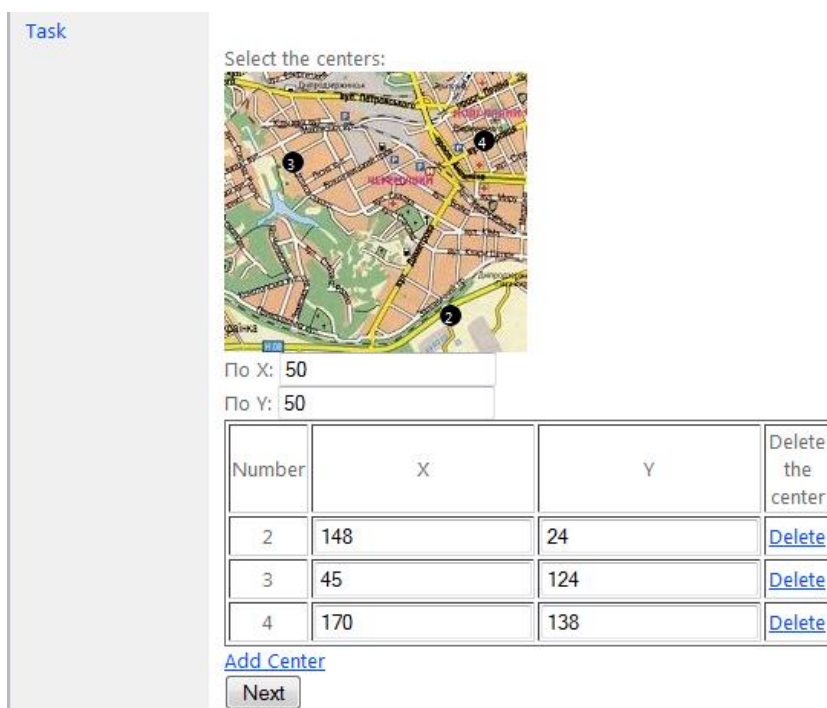


Рисунок 6 – Вид web-страницы после ввода параметров

задачи и возвращает результат её решения в виде, представленном на рис.7. Полученные результаты могут быть сохранены в виде изображения или текста.

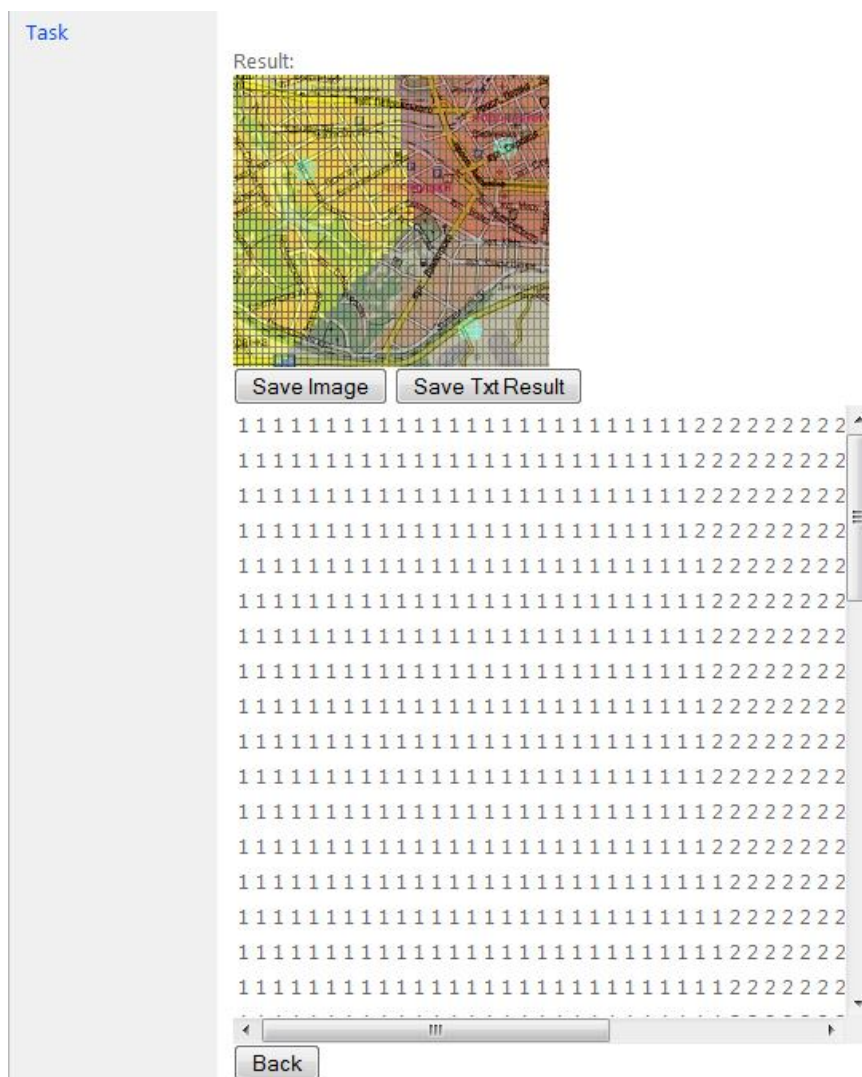


Рисунок 7 – Вид web-страницы с результатами решения задачи

Выводы. Разработан web-сервис и клиентское приложение, которое демонстрирует работу web-сервиса, для решения непрерывной задачи оптимального разбиения некоторого заданного множества на непересекающиеся подмножества без ограничений с заданными координатами "центров" этих подмножеств. Для реализации сервис-ориентированного проекта использовалась платформа .NET Framework 4.0.

ЛИТЕРАТУРА

1. Киселёва Е.М. Непрерывные задачи оптимального разбиения множеств: теория, алгоритмы, приложения: Монография / Е.М.Киселёва, Н.З.Шор. – К.: Наук. думка, 2005. – 564с.
2. Макки А. Ведение в .NET 4.0 и Visual Studio 2010 для профессионалов / А.Макки. – Вильямс, 2010. – 415с.

Поступила в редколлегию 18.06.2012.

Дніпродзержинський державний технічний університет
*Дніпродзержинський енергетичний технікум

МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ МЕРЕЖ ЯК ЗАСІБ ВИВЧАННЯ, ПРОЕКТУВАННЯ ТА ОПТИМІЗАЦІЇ РОБОТИ МЕРЕЖІ

Вступ. Постійне зростання числа комп'ютерних мереж, ускладнення їхньої інфраструктури й збільшення обсягів переданих даних створюють серйозні проблеми забезпечення ефективного керування мережними ресурсами як при адмініструванні й розвитку існуючої мережної інфраструктури, так і при проектуванні нових мереж і розробці мережних додатків. У цей час найбільш ефективним варіантом рішення задач оцінки ефективності роботи мережі, документування її поточного стану, оптимізації продуктивності, аналізу можливих удосконалень, а також надання рекомендацій для найбільш раціонального використання ресурсів мережі є використання спеціальних моделюючих систем.

Такі програмні системи самі генерують модель мережі на основі вихідних даних про її топологію й використовувані протоколи, про інтенсивність потоків запитів між комп'ютерами мережі, довжини ліній зв'язку, про типи використовуваного устаткування й додатків. Програмні системи моделювання можуть бути вузько спеціалізованими й досить універсальними, що дозволяє імітувати мережі всіляких типів. Якість результатів моделювання в значній мірі залежить від точності вихідних даних про мережу, переданих у систему моделювання [1, 2].

Розглянемо найбільш популярні системи моделювання.

BONeS (фірма Systems and Networks) – графічна система моделювання загальнопризначення для аналізу архітектури систем, мереж і протоколів. Описує моделі на транспортному рівні й на рівні додатків. Дає можливість аналізу впливу додатків типу клієнт-сервер на роботу мережі.

Netmaker (фірма OPNET Technologies) – проектування топології, засобу планування й аналізу мереж широкого класу. Складається з різних модулів для розрахунку, аналізу, проектування, візуалізації, планування й аналізу результатів.

Optimal Perfomance (фірма Compuware; Optimal Networks) – має можливості швидкого оцінного й точного моделювання, допомагає оптимізувати розподілене програмне забезпечення.

Prophesy (компанія Abstraction Software) – проста система для моделювання локальних і глобальних мереж. Дозволяє оцінити час реакції комп'ютера на запит, кількість робочих станцій для обслуговування активного устаткування, запас продуктивності мережі при поломці певного устаткування.

Сімейство CANE (компанія ImageNet) – проектування й реінжиніринг обчислювальної системи, оцінка різних варіантів. Моделювання на різних рівнях моделі OSI. Розвинена бібліотека пристроїв, що включає фізичні, електричні, температурні й інші характеристики об'єктів. Можливе створення своїх бібліотек.

Сімейство COMNET (фірма Compuware; CACI Products Company) – об'єктно-орієнтована система моделювання локальних і глобальних мереж. Дозволяє моделювати рівні: додатків, транспортний, мережний, каналний. Використовує всі відомі на сьогодні технології й протоколи, а також системи клієнт-сервер. Легко налаштовується на модель устаткування й технологій. Можливість імпорту й експорту даних про тополю-

гію й мережний трафік. Моделювання ієрархічних локальних і глобальних мереж; облік алгоритмів маршрутизації.

Сімейство OPNET (фірма OPNET Technologies) – засіб для проектування й моделювання локальних і глобальних мереж, комп'ютерних систем, додатків і розподілених систем. Можливість імпорту й експорту даних про топологію та мережний трафік. Аналіз впливу додатків типу клієнт-сервер на роботу мережі. Моделювання ієрархічних локальних і глобальних мереж; облік алгоритмів маршрутизації.

Одним з найбільш популярних на сьогоднішній день продуктів, призначених для моделювання комп'ютерних мереж всіх типів, а також імітації процесів у створених мережах, є пакет NetCracker Professional фірми NetCracker Technology [3, 4].

Як і всі сучасні програми даного типу, пакет оснащений засобами графічного проектування, що дозволяють будувати схеми мережі за допомогою спеціальної бібліотеки елементів мережної інфраструктури, яка надає користувачеві широкий вибір конкретних моделей обчислювальних і телекомунікаційних пристроїв різних фірм-виробників. Є також можливість створювати моделі пристроїв, що задовольняють вимогам користувача, регулювати рівень параметризації елементів бібліотеки, робити моделі порівнянними з реальними об'єктами, враховувати кількість класів об'єктів, що моделюються.

Графічний інтерфейс користувача являє собою модуль для взаємодії з підсистемами завдання робочого навантаження й топології мережі. Він забезпечує максимальну зручність для користувача за рахунок механізму drag-and-drop, наочності іконок, що позначають елементи мережі, можливості згорнути окремі фрагменти мережі.

Середовище прогону використовується для збору даних про функціонування моделі, що при необхідності відображається на екрані або діаграмою завантаженості, або в процентному співвідношенні. Є також можливість анімації процесу моделювання мережі. Можна припиняти або переривати роботу моделі, прокручувати назад анімаційну картинку й запускати повторно.

Підсистема аналізу результатів моделювання обробляє дані, зібрані при прогоні моделі, обчислює характеристики продуктивності й представляє результати в зручній для користувача формі. У значній мірі можливість цієї підсистеми залежить від тих даних, які збирає середовище прогону. Визначальними для цієї частини системи є кількість і тип характеристик, що збираються в результаті роботи моделі.

Постановка задачі. Метою роботи є проведення моделювання сегменту комп'ютерної мережі у пакеті NetCracker Professional. Для проектування структури мережі програма надає можливість вибору необхідного устаткування з вбудованої бази даних, а також додавання в базу даних і конфігурування нового обладнання різних типів. Користувач розміщує обрані компоненти на складальному полі, задає структуру і тип зв'язків між ними, визначає тип програмного забезпечення і характер трафіка між вузлами мережі. Надалі користувач має можливість указати перелік аналізованих характеристик, вигляд відображення статистичної інформації і виконати імітаційне моделювання проектованої мережі.

Типовий вигляд головного вікна програми *NetCracker* наведено на рис.1.

В лівій верхній частині головного вікна розташована *панель компонент*, що включені у базу даних.

Список пристроїв має кілька виглядів відображення:

– *Types* (Типи) – пристрої в списку групуються за типами, після чого у кожній групі можна виділити підтипи пристроїв за функціональними ознаками. Надалі пристрої можуть поділятися за виробниками (рис.2);

– *Vendors* (Виробники) – пристрої в списку групуються за виробниками, після чого у кожній групі можна виділити підгрупи, що відповідають типу пристроїв (рис.3);

– *User* (Користувальницьке) – відображення пристроїв, обумовлене користувачем. У свою чергу таке відображення може бути згруповане за типами або виробниками пристроїв.

Закладка *Compatible Devices* призначена для відображення списку сумісних пристроїв.

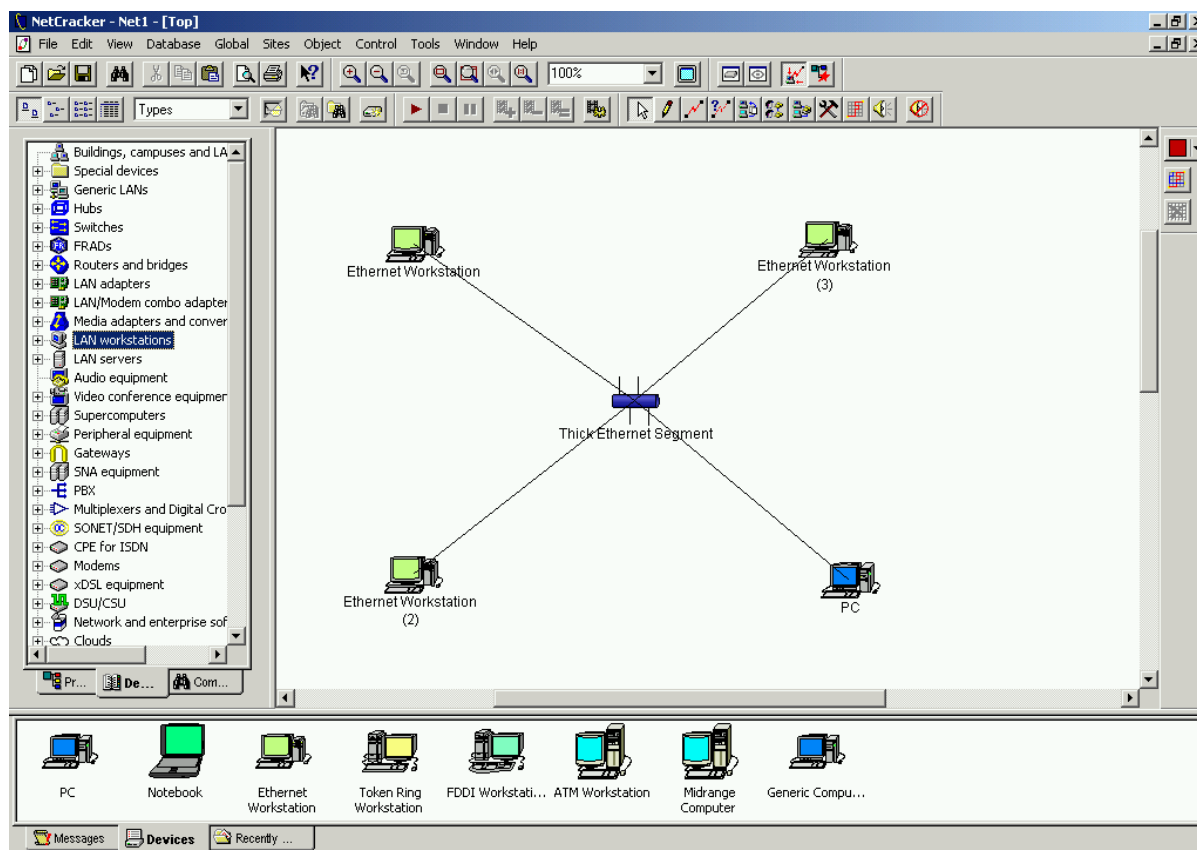


Рисунок 1 – Відображення основних складових частин головного вікна

У нижній частині головного вікна програми розташована панель пристроїв, що призначена для відображення пристроїв з обраної у панелі компонентів групи.

У правій верхній частині головного вікна програми розташовується вікно, що являє собою складальне поле. У цьому полі розташовують використовувані компоненти при проектуванні структури мережі.

Під час складання структури мережі використовувані пристрої розміщують у складальному полі і поєднують їх лініями зв'язку. Для розміщення пристрою в складальному полі обирають відповідний клас і тип пристрою у панелі компонентів. Після цього обраний пристрій необхідно перетягнути за допомогою миші в складальне поле і розмістити в потрібному місці.

Для відображення реальної структури мережі організації бажано використовувати такі класи компонентів: *City* (Місто); *Building* (Будинок); *Campus* (Університет); *Floor* (Поверх); *Room* (Кімната).

Кожний із вказаних об'єктів має своє складальне поле, де будують відповідну даному об'єктові частину мережі.



Рисунок 2 – Відображення пристроїв за типами

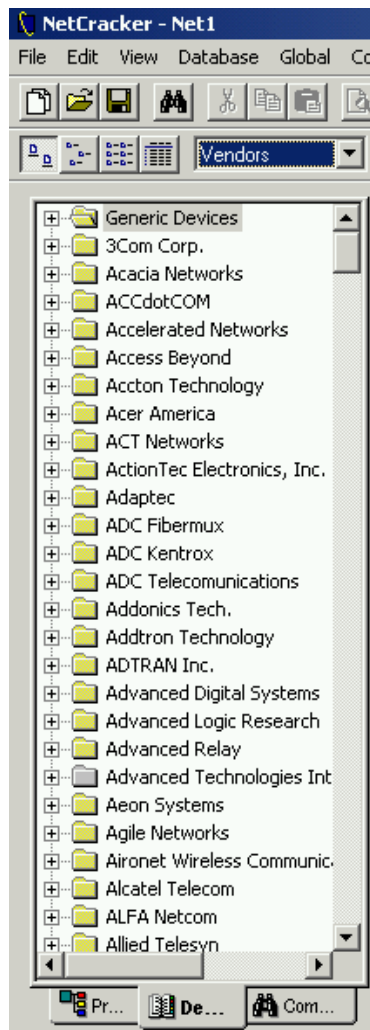


Рисунок 3 – Відображення пристроїв за виробниками

При виборі пристроїв, використовуваних для побудови мережі, насамперед варто враховувати такі параметри: необхідна кількість портів, необхідний тип портів, пропускна здатність, підтримувані транспортні протоколи, підтримувані протоколи маршрутизації, кількість слотів та ін.

Більшість використовуваних пристроїв вимагають установки визначених компонентів для виконання ними необхідних функцій. Для цього в панелі компонент обирають необхідне устаткування і перетягують його за допомогою миші на потрібний об'єкт.

Під час установки додаткового устаткування необхідно враховувати кількість і тип слотів. Наприклад, якщо пристрій не має портів *MCA*, то установити в нього мережну карту, що розрахована на шину *MCA*, неможливо. Або неможливо установити додаткове устаткування

в пристрій, якщо він не має вільних слотів.

Результати роботи. На рис.4 наведено приклад сегменту мережі *Ethernet* специфікації *10BASE5*, де робочі станції поєднуються за допомогою товстого коаксіального кабелю (*Thick Coaxial Cable*).

Виконаємо моделювання заданого сегменту мережі у програмі *NetCracker*. У панелі компонент із групи *Lan workstation* оберемо необхідний тип робочих станцій та з групи *Generic LANs* – необхідний тип передаючого середовища. Приклад зібраного сегменту мережі *Ethernet* наведено у складальному полі головного вікна на рис.1.

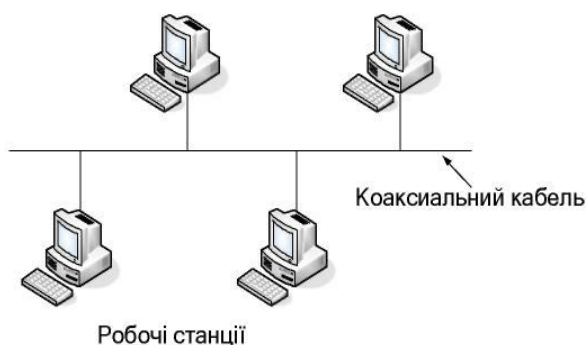


Рисунок 4 – Сегмент мережі Ethernet 10BASE5

Для підключення до мережі робочої станції (комп'ютеру) *PC* (1) на нього необхідно встановити мережну карту. Для цього зі списку *LAN adapters* меню компонент оберемо і перетягнемо на відповідний об'єкт необхідне устаткування.

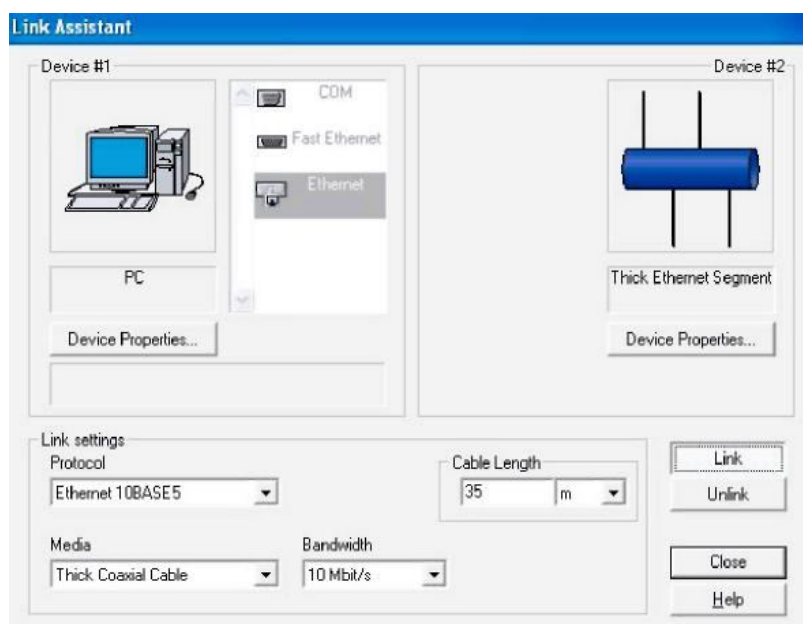



Рисунок 5 – Налаштування параметрів з'єднання


Для встановлення зв'язків між пристроями (між їх інтерфейсами або портами) необхідно скористатися кнопкою Link Devices . За допомогою миші обирають один із пристроїв, що поєднується, і, утримуючи кнопку миші, розтягують зв'язок до другого пристрою. Після цього відкривається діалогове вікно Link Assistant (рис.5), де здійснюється подальше конфігурування параметрів з'єднання. Під час початкового конфігурування надається можливість

визначити порти поєднуваних пристроїв і зв'язати їх натисканням миші на кнопку Link. Після цього стає доступною секція Link Settings, де налагоджуються параметри даного з'єднання:

- *Protocol* (використовуваний протокол) - *Ethernet 10Base5*, *Ethernet 10Base2*, *Ethernet 10Base-T*, тощо;
- *Media* (тип середовища передачі) – *Thick Coaxial Cable* (товстий коаксіальний кабель), *Thin Coaxial Cable* (тонкий коаксіальний кабель), (вита пара), тощо;
- *Bandwidth* (пропускна здатність середовища);
- *Length* (довжина з'єднання).

У більшості випадків указані параметри є фіксованими і не змінюються, хоча іноді надається можливість вибору декількох значень. Наприклад, під час з'єднання двох оптоволоконних модемів пропускна здатність може бути обрана з такого переліку значень: *T3*, *E3*, *DSn*, *Ocn*, *STSn*, *STMn*, або для аналогових модемів: 2400, 9600, 14400, 28800 тощо. Тип з'єднання в даному випадку є однаковий – *frame relay* (ретрансляція кадрів). Середовище передачі також фіксоване – *fiber-optic cable* (оптоволоконно). Під час з'єднання пристроїв, що мають порт *ISDN*, список типів з'єднань є більш широкий – *ISDN BRI*, *ISDN PRI*, *point-to-point leased line* (виділена лінія), *dial-up analog line* (аналогова телефонна лінія).

Після завдання структури мережі і топології зв'язків визначається склад і розташування використовуваного програмного забезпечення. Для цього в списку компонентів обирають категорію *Network and enterprise software*, де знаходять необхідне програмне забезпечення і розміщують його за допомогою миші у відповідний об'єкт мережі.

Надалі визначається трафік між вузлами мережі. Для завдання трафіка необхідно скористатися кнопкою Set Traffic  на панелі режимів покажчика миші і перейти у відповідний режим. Далі послідовно обирають пари абонентських станцій (АС) мережі, між якими задається трафік. Порядок натискань мишею на АС визначає напрямок передачі – на початку відзначається джерело передачі, далі приймач. Після цього відкривається діалогове вікно Profiles (рис.6), де задають тип і основні характеристики трафіка.

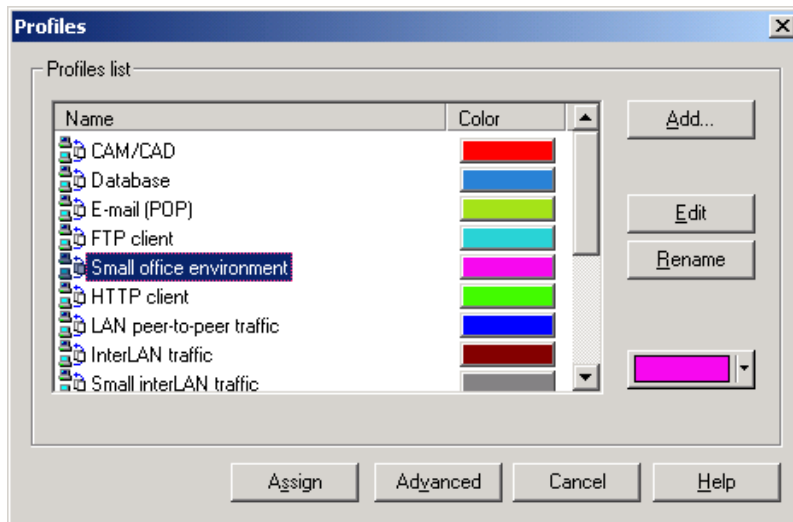


Рисунок 6 – Налаштування трафіка

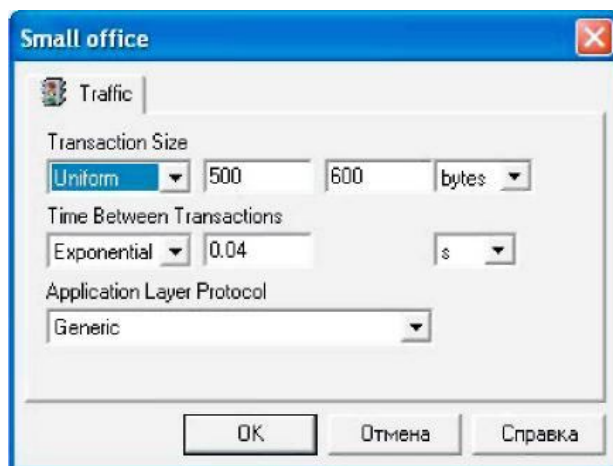


Рисунок 7 – Налаштування характеристик трафіка

кнопками Add, Remove, Edit і Rename діалогового вікна Profiles (рис.6). В стовпці Color обирають колір, яким при моделюванні будуть відображатися пакети відповідного типу трафіка. Характеристики типів трафіка, що використовуються за замовчуванням, можна змінити за допомогою команди головного меню Global→Profiles. Також для конфігурування потоків даних у мережі використовують команду Global→Data Flow.

Характеристики, що аналізуються, задаються командою головного меню Object→Define Statistics або командою Statistics контекстного меню. При цьому відкривається діалогове вікно Statistical Items (рис.8), де задають тип характеристики і спосіб відображення статистичної інформації в процесі моделювання. Це діалогове вікно має різний вигляд для різних типів об'єктів, тобто змінюється перелік характеристик, а також стають недоступними деякі способи відображення інформації. Окрім того, для одного і того самого об'єкта (обраного пристрою, з'єднання або потоку даних) можна обрати кілька способів відображення інформації (індикатор, число, графік).

Надалі здійснюється імітаційне моделювання роботи мережі. Для керування процесом моделювання користуються командами головного меню Control. Команда Start застосовується для запуску, Pause – припинення та Stop – для зупинки процесу моделювання. Команди Simulation Faster і Simulation Slower призначені для регулювання швидкості моделювання. Команди Animation Faster, Animation Slower і Animation

Тип трафіка обирають зі списку Profiles List за принципом "запити клієнта до сервера", тобто приймачем може бути тільки та АС, де функціонує відповідне програмне забезпечення (*HTTP/FTP Server, SQL Server, File Server*). Деякі типи трафіка є винятком з вказаного правила, наприклад, *Small office, LAN peer-to-peer traffic, InterLAN traffic* тощо. Завдання характеристик

трафіка між зазначеними АС виконується за натисканням кнопки Advanced. При цьому відкривається діалогове вікно Traffic from (АС-джерело) to (АС-приймач) (рис.7), де задають закон розподілу і діапазон значень для розміру запиту (*Transaction Size*) і інтервалу між запитами (*Time Between Transactions*), а також тип протоколу прикладного рівня (*Application Layer Protocol*).

За необхідності додати новий тип трафіка, видалити або змінити параметри існуючих типів трафіка слід скористатися

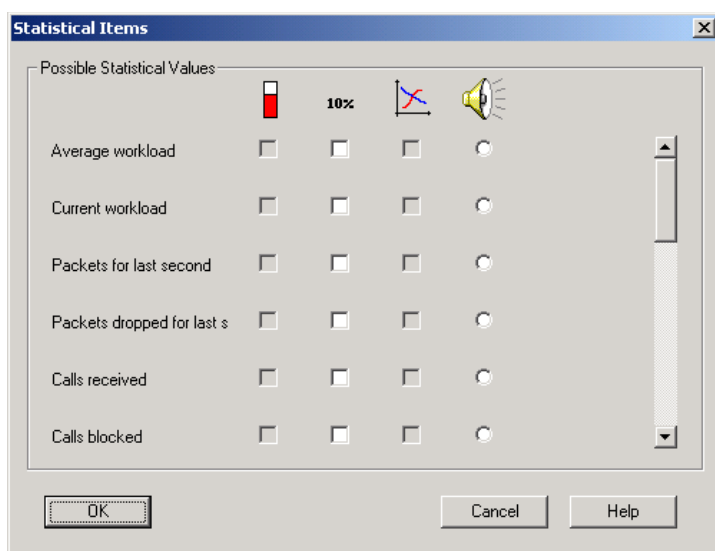


Рисунок 8 – Налаштування відображення статистичної інформації при моделюванні

команди відображується статистика за процентним співвідношенням кількості пакетів для вхідних (*Incoming Traffic*) і вихідних з'єднань (*Outgoing Traffic*).

Звіт, що узагальнює результати виконаної роботи, створюють, користуючись командами головного меню Tools→Reports.

Висновки. Програмне середовище моделювання NetCrackerPro є одним з найефективніших засобів вивчення, проектування та оптимізації функціонування комп'ютерних мереж. За його допомогою можуть бути вирішені такі задачі, як визначення продуктивності мережі при завданні топології й робочого навантаження, аналіз залежності пропускної здатності при зміні робочого навантаження на мережу, аналіз залежності пропускної здатності мережі при зміні її топології, підбір параметрів протоколів мережі для забезпечення максимальної пропускної здатності мережі при заданих топології й робочому навантаженні, визначення оптимальної топології й відношення пропускної здатності до вартості проєктованої мережі. Особливо корисно моделювання комп'ютерних мереж при їх вивчанні в навчальному процесі, тому що проектування мережі та оптимізація її роботи виконується користувачем на «віртуальній» мережі, що відкриває багаті можливості для використання максимально більшого числа параметрів. Це активізує пізнавальну активність користувачів, підсилює їх інтерес та мотивацію, формує інформаційно-комунікаційну компетентність.

ЛІТЕРАТУРА

1. Проектирование и диагностика компьютерных систем и сетей: Учебное пособие / Бондаренко М.Ф., Кривуля Г.Ф., Рябцев В.Г., Фрадков С.А., Хаханов В.И. - Киев: НМЦ ВО, 2000. - 306 с.
2. Инструментальные средства проектирования, имитационного моделирования и анализа компьютерных сетей: учеб. пособие для студ. вузов / Л. А. Пономаренко, В. И. Щелкунов, А. Я. Сляров. - 2-е изд., испр. и доп. - Харьков : Компания СМІТ, 2006. - 488 с.
3. Сайт компанії NetCracker Technology <http://netcracker.com/>.
4. Інструкція користувача NetCracker Tutorial.

Надійшла до редколегії 22.06.2012.

Дніпродзержинський державний технічний університет

АВТОМАТИЗОВАНІ РОБОЧІ МІСЦЯ В СТРУКТУРІ УПРАВЛІННЯ ПРОМИСЛОВИМ ПІДПРИЄМСТВОМ

Вступ. Ефективне управління підприємством (ресурсами, асортиментом, збутом) завжди мало визначальну роль для результатів роботи будь-якого промислового підприємства. Нині у зв'язку зі збільшенням масштабів виробництва, переходом підприємств на нові умови господарювання вимоги до якості та своєчасності управлінських рішень ще більше зростають. Як наслідок, стає все більш актуальною задача ефективного та швидкого опрацювання великих обсягів інформації. Цю проблему покликана вирішити комп'ютеризація управлінської діяльності.

Питання комп'ютеризації управлінської діяльності досліджувалися в працях багатьох науковців, які розглядали загальні принципи і підходи до проблеми [1-4]; аналізували конкретні програмні засоби та методологію вирішення окремих завдань управління за допомогою ПК [5-11]. В той же час питання функціонування підприємства при роботі в умовах всебічної інформатизації управління потребують подальшого розвитку досліджень, зокрема, в напрямку визначення місця та ролі автоматизованих робочих місць (АРМ) окремих фахівців в загальній організаційно-функціональній структурі управління підприємством.

Постановка задачі. Створення комп'ютерної інформаційної мережі управління, як правило, здійснюється на базі вже працюючого підприємства, тобто існуючої структури управлінської служби. При цьому виникають питання щодо структурної перебудови системи управління або автоматизації існуючих функціональних елементів системи. Прагнення забезпечити ефективність діяльності підприємства зумовлює необхідність створення ефективної комп'ютерної інформаційної системи управління з мінімальною перебудовою існуючої структурно-функціональної схеми.

Результати роботи. В межах підприємства всі функції управління логічно взаємозв'язані. Деякі з функцій можна однозначно віднести до компетенції одного з існуючих елементів традиційної системи управління, а деякі знаходяться на перетині області впливу кількох елементів. Для підвищення ефективності управління найбільш доцільно зосередити такі функції у межах окремих структурних підрозділів на базі впровадження відповідних АРМ управлінців (менеджерів). Такі АРМ призначені для того, щоб зв'язати складні розгалужені потоки необхідної інформації в одній точці системи управління. Саме в рамках цих елементів мають вирішуватися функціональні задачі і прийматися управлінські рішення, тобто об'єднуватися усі функції управління.

Структура відповідного комплексу АРМ може мати кілька варіантів. У першому випадку повноваження управляючого органу по видам ресурсів (діяльності) передаються в один із уже діючих на підприємстві відділів із виділенням у ньому посади відповідного менеджера. В іншому випадку створюється новий відділ із передачею відповідних повноважень йому. В будь-якому випадку АРМ кожного управлінця (менеджера) повинно бути оснащено функціональною підсистемою на основі потужної інформаційної бази, яка постійно оновлюється відповідними відділами. До кожного такого АРМ повинна надходити вся інформація про наявність і рух певного ресурсу підприємства або ж про стан діяльності в певному напрямку. У свою чергу, із кожного АРМ по-

винні виходити потоки управлінської інформації: аналітичних звітів, проектів заходів, обґрунтування управлінських рішень, планів та прогнозів.

Зауважимо, що структура системи управління сучасного промислового підприємства дуже розгалужена. Генеральному директору безпосередньо підпорядковані кілька заступників (начальників управлінь, відділів, головних спеціалістів). Кожному з таких управлінців, в свою чергу, підпорядковані певні відділи зі своєю структурою. Тому для наочності ілюстрації пропонуємо далі розглядаються лише фрагменти системи управління промисловим підприємством, пов'язані з управлінням основними ресурсами.

На рис.1 представлено перший варіант структури організації комплексу АРМ, коли кожному існуючому робочому місцю управлінської служби підприємства має відповідати своє АРМ із збереженням задач та підпорядкованості працюючих спеціалістів.

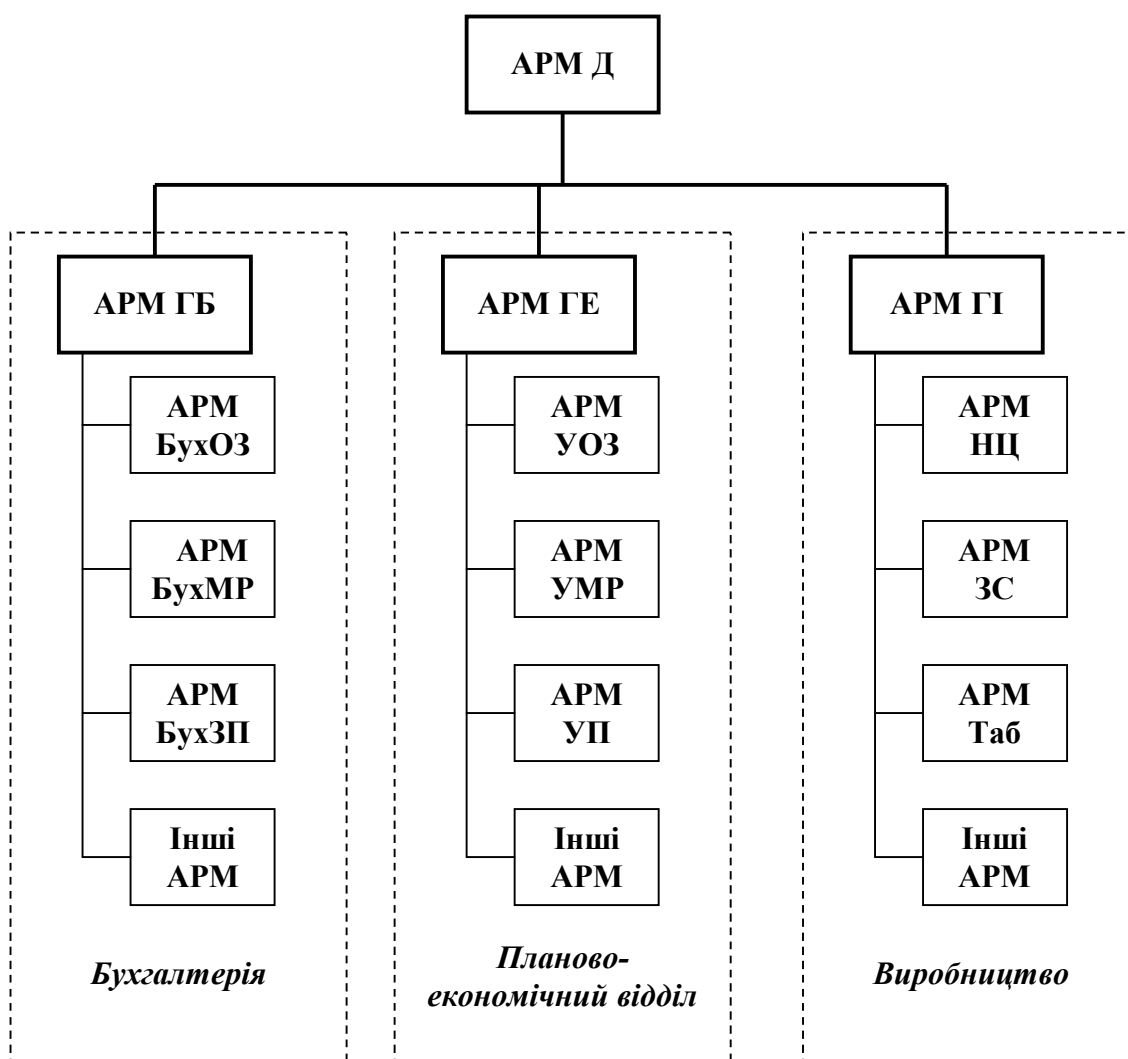


Рисунок 1 – Структура системи АРМ підприємства (I варіант)

На рис.1 застосовано наступні позначення: АРМ Д, АРМ ГБ, АРМ ГЕ, АРМ ГІ – автоматизовані робочі місця директора, головного бухгалтера, головного економіста і головного інженера відповідно; АРМ БухОЗ, АРМ БухМР, АРМ БухЗП – автоматизовані робочі місця бухгалтерів з основних засобів, матеріальних ресурсів та заробітної

плати; АРМ УОЗ, АРМ УМР, АРМ УП – автоматизовані робочі місця економістів з управління основними засобами, матеріальними ресурсами і персоналом; АРМ НЦ, АРМ ЗС, АРМ Таб – робочі місця фахівців, безпосередньо пов'язаних з виробничим процесом та розташованих у виробничих підрозділах (начальників цехів, завідувачів складами і табельників відповідно).

Другий варіант відповідає випадку, коли окремо виділяється група АРМ менеджерів за функціональними ознаками: з управління основними засобами (АРМ УОЗП), з управління матеріальними ресурсами (АРМ УМРП), з управління персоналом підприємства (АРМ УПП) та інші. В такому випадку функціональні АРМ або виділяються в окремий відділ (відділ менеджменту або АСУ), або створюються в межах існуючих підрозділів, але підпорядковуються напряму заступнику директора з управління (головному менеджеру) через АРМ ГМ (рис.2, 3). I та II рівні розмежовують лінійні рівні управління підприємства (організації).

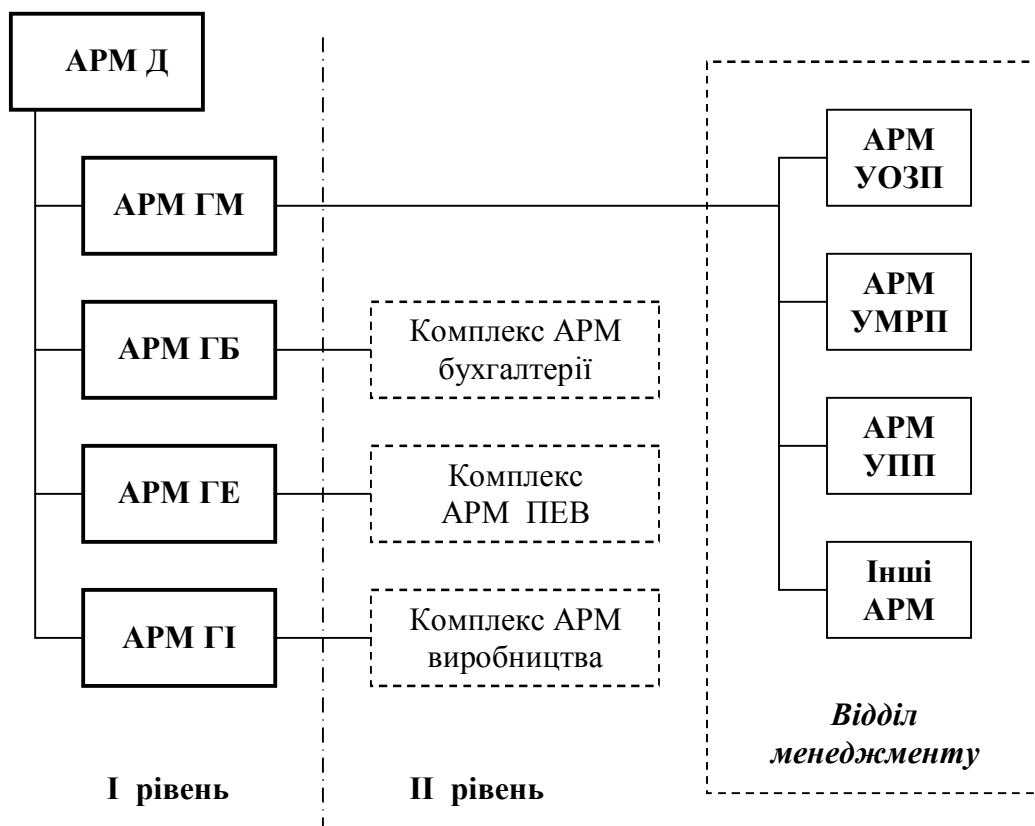


Рисунок 2 – Структура системи АРМ підприємства (II варіант)

Зауважимо, що в більшості існуючих організаційних структур управління підприємствами (організаціями) посада заступника директора з управління (головного менеджера) в структурі управління відсутня. Отже, організація комплексу АРМ за II і III варіантами потребує більших змін в системі управління, а головне – змін в технології управління.

Впровадження пропонованих комплексів АРМ сприяє практичному удосконаленню структури управління будь-якого підприємства (організації). При цьому під вдосконаленням мається на увазі не тільки встановлення оптимальної кількості управлінського апарату і кількості працівників у виробничих підрозділах згідно з існуючими

нормативами, а перш за все – поліпшення системи планування, обліку і контролю за основними показниками діяльності підприємства за допомогою широкого та всебічного впровадження обчислювальної техніки, розвитку комп’ютерних мереж, засобів зв’язку, застосування сучасних програмних продуктів: технологій управління, інформаційних технологій.

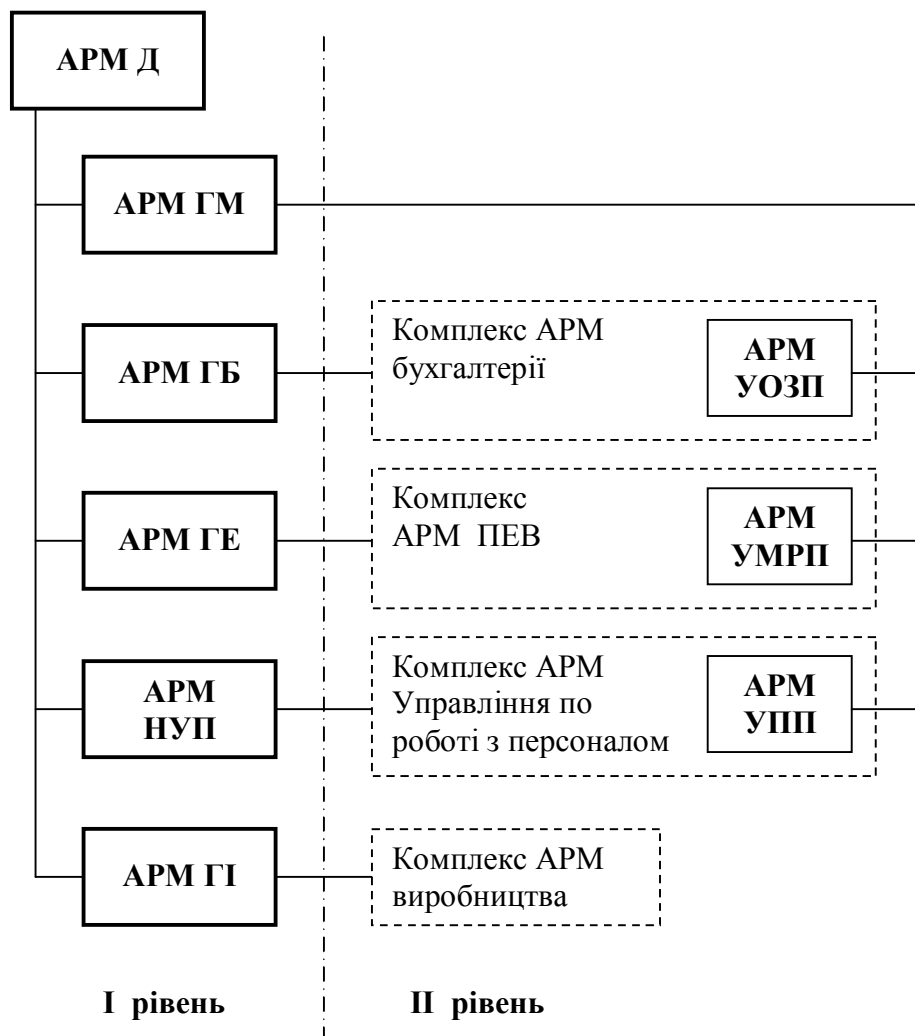


Рисунок 3 – Структура системи ARMS підприємства (III варіант)

Висновки. Запропоновано підходи до створення інформаційної комп’ютерної системи управління підприємством на базі використання комплексу взаємопов’язаних ARMS, які визначають розташування ARMS окремих фахівців, що має сприяти розробці функціонального наповнення (технічного завдання) до розробки відповідних робочих місць управлінців підприємства в умовах створення загальної управлінської інформаційної системи.

ЛІТЕРАТУРА

1. Вовчак І.С. Інформаційні системи та комп’ютерні технології в менеджменті / І.С.Вовчак. – Тернопіль: Карт-бланш, 2001. – 286с.
2. Гордієнко І.В. Інформаційні системи і технології в менеджменті / І.В.Гордієнко. – К.: КНЕУ, 2003. – 259с.

3. SAP R/3: Менеджмент / [под ред. М.Ребштока, К.Хильдербранда; пер. с нем.]. – Мн.: Новое знание, 2001. – 208с.
4. Пономаренко В.С. Проектування автоматизованих економічних інформаційних систем: навч. посібник / В.С.Пономаренко, О.І.Пушкар, Ю.І.Коваленко. – К.: ІЗМН, 1996. – 312с.
5. Завгородній В.П. Автоматизація бухгалтерського обліку, контролю, аналізу та аудиту / Завгородній В.П. – К.: А.С.К., 1998. – 768с.
6. Оксанич А.П. Інформаційні системи і технології маркетингу / А.П.Оксанич. – К.: Професіонал, 2008. – 320с.
7. Пономаренко В.С. Інформаційні системи в управлінні персоналом / В.С.Пономаренко. – Харків: ХНЕУ, 2008. – 336с.
8. Сендзюк М.А. Інформаційні системи в державному управлінні / М.А.Сендзюк. – К.: КНЕУ, 2004. – 339с.
9. Сиротинська А.П. Інформаційні системи підприємств малого бізнесу / А.П.Сиротинська. – К.: Центр учбової літератури, 2008. – 264с.
10. Грановський С.С. Про один підхід до управління матеріальними ресурсами промислових підприємств / Грановський С.С. Карімов Г.І. // Економіка: проблеми теорії та практики: [збірник наукових праць]. – Дніпропетровськ: ДНУ. – 2001. – Випуск 71. – С.33-36.
11. Грановський С.С. До питання координації управління матеріальними ресурсами / Грановський С.С., Карімов Г.І. // Економіка: проблеми теорії та практики: [збірник наукових праць]. – Дніпропетровськ: ДНУ. – 2003. – Випуск 76. – С.45-49.

Надійшла до редколегії 06.06.2012.