

К. О. СОЛОВЙОВА, д-р техн. наук, зав. каф СІ ХНУРЕ,
В. В. МОВЧАН, стажист-дослідник ХНУРЕ

РОЗРОБКА МОДЕЛІ ПРОГРАМНОГО ЗАСОБУ ВІЗУАЛІЗАЦІЇ МЕРЕЖЕВИХ СТРУКТУР

Розглядаються сучасні методи та існуючі програмні засоби візуалізації мережних структур. Визначаються вимоги до програмного засобу з візуалізації мережних структур. Визначаються недоліки існуючих програмних засобів. Розроблюється модель програмного засобу, що відповідає означеним вимогам. Створюється програмний засіб на основі розробленої моделі.

Рассматриваются современные методы и существующие программные средства визуализации сетевых структур. Определяются требования к программному средству визуализации сетевых структур. Определяются недостатки существующих программных средств. Разрабатывается модель программного средства, отвечающего определенным требованиям. Создается программное средство на основе разработанной модели.

The article deals with modern methods and existing software visualization of network structures. Identify requirements for the visualization software network structures. Identify the shortcomings of existing software. The author develops a model of software tools that meet certain requirements. It creates a software tool based on the developed model.

Постановка проблеми. Ускладнення усіх видів економічної діяльності в сучасних умовах призвело до зміщення акцентів розвитку спільноти з матеріальних на інформаційні. Разом із цим, значно зросла роль аналізу інформації та гостро постала проблема забезпечення аналітиків зручними та доступними засобами обробки та аналізу інформації.

Природним засобом представлення знань є представлення їх у вигляді графу [2]. Теорія графів набула широкого використання в таких несхожих сферах, як економіка, системологія [7], теорія управління, маркетинг, інформатика, хімія, програмування, логістика, схемотехніка та безліч інших [2, 3]. Практично, за допомогою теорії графів може бути, в тій чи мірі, представлена будь-яка інформація чи знання.

Проблему забезпечення аналітика, інструментом, що базується на теорії графів намагаються вирішити не перший рік - перші програмні реалізації такого інструменту з'явилися на початку дев'яностих років [1]. З того часу була створена чисельна множина різноманітних програм, які в тій чи іншій мірі виконують своє призначення бути засобом для відображення графових структур. Кожна з цих реалізацій має свої недоліки та переваги. Але ці програми мають один великий спільний недолік, який полягає в тому, що вони не є доступними, тобто мають високу ціну та не є крос-платформними. Крім того, у безкоштовних реалізаціях програмних засобів часто не передбачена підтримка кирилических шрифтів та використовуються спеціальні недокументовані формати вихідних даних, що ускладнює організацію

передачі даних у інші програми. Беззаперечно, існують програми, які можливо завантажити з мережі Internet. Та ці програми, зазвичай, мають або дуже обмежений функціонал, або не є безкоштовними і можуть бути використовувані без оплати лише протягом ознайомчого періоду.

Таким чином, актуальною є задача розробки моделі програмного засобу та її програмна реалізація, що не має означених недоліків та може використовуватися в Україні для вирішення проблеми забезпечення аналітика зручним засобом з візуалізації інформації.

Мета статті. Розробка моделі програмного засобу та її програмна реалізація, що не матиме означених вище недоліків, а саме: буде крос-платформною, підтримуватиме кирилическі шрифти, буде використовувати загальновідомий формат вихідних даних XML. Крім того, програмний засіб має бути відкритий.

Аналіз останніх досліджень і програмних засобів. Наукоємні продукти з використанням методів візуалізації інформації існують на світовому ринку останні 15 років. Є фірми, що надають бібліотеки і програмні комплекси, орієнтовані на візуалізацію графів загального призначення (Tom Sawyer software, ILOG software, Algorithmic Solutions Software GmbH, yWorks), а також системи програмного забезпечення (Imagix Corporation, Absint та ін.) [1]. «В додаток з'являється все більше фірм, що спеціалізуються на візуалізації бізнес-інформації, необхідної аналітикам різних підприємств та орієнтованої на спеціалізацію тих чи інших підприємств (Enterprise Solutions). Однею зі старіших компаній цього напрямку є фірма Inxight Software, Inc, що надає засоби візуалізації інформації для фінансових і біологічних фірм. З'явилися компанії, що надають на ринок продукти, які використовують нові методи в галузі візуалізації інформації, так звані методи візуальної аналітики. В зростаючій родині інструментів візуальної аналітики квітне програмне забезпечення фірми HiveGroup, яка використовує візуалізацію ієрархічних даних на ґрунті карти дерева (Treemap) і надає організаціям потрібний їм щоденний моніторинг складної діяльності за участю тисяч продуктів, проектів та продавців» [1].

Визначення вимог до програмного засобу. Проблема розміщення графу є невід'ємною частиною теорії графів та проявляється при роботі навіть із невеликими графами, та ще не є критичною. Але коли мова заходить про те, щоб візуалізувати граф навіть із декількома десятками вершин та ребер, проблема розміщення стає першорядною. Реальні ж задачі, деколи потребують візуалізації графу, що містить десятки тисяч вершин та ще більше ребер [1, 2, 6].

Тому першим мірилом якості програмного засобу є реалізація якомога більше методів візуалізації графу. Звісно, реалізація більшості методів потребує дуже багато часу, тому в це ж мірило слід віднести можливість розширення набору реалізованих методів.

З першого мірила природно походить друге – програмний засіб має бути спроектовано таким чином, щоб його можливо було розширювати. Тобто щоб реалізація, наприклад, нових методів візуалізації не потребувала переробки загальної моделі [4].

Дуже часто, інформація, що має бути представлена графічно у вигляді графу, є результатом роботи іншої інформаційної системи. Це можуть бути дані з якоїсь бази даних, наприклад, інформація про ієрархію персоналу підприємства, або результат роботи програмних систем з маркетингових досліджень, тощо. В будь якому разі, коли мова йде про взаємодію різних інформаційних систем, дуже гостро постає питання як саме має бути спроектована система, щоб її можливо було використовувати інтегровано з іншою [2]. Отже наступне мірило якості – можливість та якість організації взаємодії між системою, що розробляються, та іншими системами, які потенційно можуть бути використані для рішення різних етапів однієї задачі.

Як відомо, кожна складна система потребує підтримки. Система, що розробляється також не я виключенням, тому наступним мірилом слід вважати те, наскільки просто систему підтримувати [4].

Наступними мірлами слід виділити підтримку програмним засобом роботи із різними кодуваннями, в тому числі із кирилицею, та можливість роботи системи на різних операційних системах.

Останнім мірилом слід виділити так звану дружність до користувача, що мовою оригіналу називаються user-friendly. Під цим терміном мається на увазі зручність користування системою.

Отже кінцевий список мірил якості програмного засобу має виглядати так:

- можливість розширення списку реалізованих методів;
- розширюваність програмного засобу;
- можливість та якість організації взаємодії за іншими системами;
- зручність підтримки;
- робота із різними кодуваннями, в тому числі із кодуваннями кирилиці;
- зручність користування засобом.

Вибір шаблону проектування. Для досягнення найвищих показників якості згідно із такими визначеними мірлами, як розширюваність та зручність підтримки доцільно звернутись до шаблону проектування Model-View-Controller (MVC)

Основна ідея цього шаблону полягає в тому, щоб відокремити бізнес-логіку від інтерфейсу користувача та в результаті отримати систему, в якій можливо змінювати представлення, не змінюючи поведінку, та навпаки [4].

Згідно термінології MVC:

- модель (model) складає методи доступу до даних Системи, що зберігаються у зовнішньому джерелі або у самій моделі;
- представлення (view) відповідає за інтерфейс користувача (кнопки, поля для вводу тексту, таблиці, графіки, тощо),

- контролер управляє бізнес-логікою та взаємодією користувача, що діє через представлення, та даними.

Проста діаграма, що відображає взаємодію між моделлю, представленням та контролером представлена на рис 1. На діаграмі суцільними стрілками позначено безпосередній зв'язок, а пунктиром – опосередкований [4].

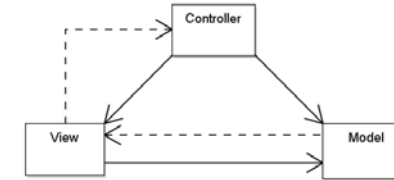


Рис. 1. Взаємодія між моделлю, представленням та контролером.

Найчастіший варіант потоку управління у MVC складаються з таких етапів [4]:

- а) користувач вступає у взаємодію з інтерфейсом користувача, наприклад, натискає на кнопку;
- б) контролер оброблює вхідну подію (event) від інтерфейсу користувача через обробник (handler);
- в) контролер повідомляє модель про дію користувача, що, можливо, призводить до зміни по моделі;
- г) представлення опосередковано використовує модель для зміни інтерфейсу користувача
- д) інтерфейс користувача очікує на подальші дії користувача, які, в свою чергу, починають цикл спочатку.

Шаблон MVC дозволяє мати декілька незалежних представлень моделі.

Побудування моделі. З ціллю підвищення ефективності розробки моделі програмного засобу візуалізації графів, було прийнято рішення використовувати спеціальну мову моделювання – UML (unified modeling language) [5].

Отже, для опису функціональної частини програмного засобу та варіантів його використання було розроблено діаграму варіантів використання (use-case diagram). В ході аналізу було визначено, яким функціоналом має володіти програма для забезпечення повноцінної роботи аналітика:

- а) відображення всього графу;
- б) відображення найкоротших шляхів між двома вибраними користувачем вершинами;
- в) відображення повного графу, побудованого відносно вибраної користувачем вершини та обмеженого вибраною користувачем глибиною;
- г) у користувача має бути можливість проглядати весь список вершин, які присутні у графі;
- д) редагування вершин та ребер;

- е) масштабування;
- ж) можливість збереження та завантаження графу у форматі XML;
- з) додавання до графу нових вершин та зв'язків;
- і) видалення вершин та ребер;
- к) управління відображенням імен та деяких інших властивостей вершин і ребер на графіку;
- л) можливість зміни розташування вершин самим користувачем так, як він того забажає.

В результаті аналізу задачі виявлено три ключові сутності – вершина, ребро та графік, на якому граф має відображатися. Також було прийнято рішення, що сам граф не варто розглядати як сутність тому, що усі властивості графу повністю відображаються його складовими, тобто вершинами та ребрами.

Згідно із прийнятим шаблоном MVC та визначеними атрибутами сутностей, була розроблена діаграма класів [5]. У спрощеному вигляді діаграма представлена на рис 2.

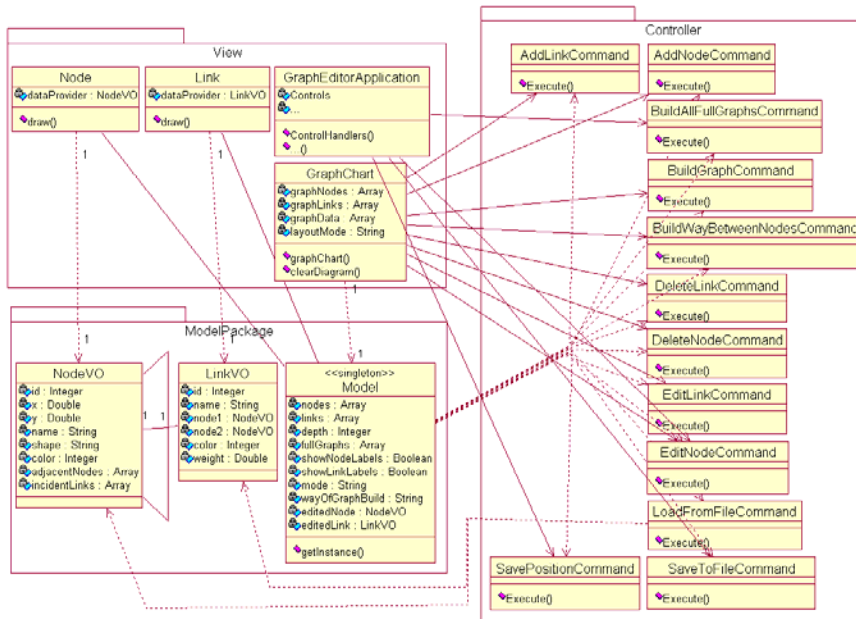


Рис. 2. Діаграма класів

Висновки. Розроблена модель програмного засобу та її програмна реалізація є крос-платформною, підтримую роботу з кириличними шрифтами, використовує стандартизований формат даних XML. Завдяки використанню шаблону проектування MVC розроблена модель також відповідає вимогам

розширюваності та зручності підтримки. Таким чином виконано усі вимоги, які були заявлені у меті.

Список літератури: 1. Апанович З.В. От рисования графов к визуализации информации. Новосибирск: препринт, 2007 2. Апанович З.В. Средства для работы с графами большого объема: построение и оптимизация компоновочных планов // Системная информатика: Сб.науч. тр. – Новосибирск: Изд-во СОРАН, 2006. – Вып. 10. Методы и модели современного программирования. – С. 7–58. 3. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ-Петербург, 2003. – 1104 с. 4. Catalyst - Web Framework [<http://www.catalystframework.org/>] 5. Г. Буч, Д. Рамбо, А. Джекобсон. Язык UML. Руководство пользователя, Москва: Изд-во. ДМК, 2006. 6. Balzer M., Deussen O., Lewerenz C. Treemaps for the Visualization of Software metrics// ACM Symp. on Software Visualization (SoftVis), 2005. 7. М.Ф. Бондаренко, Е.А. Соловьева, С.И. Маторин.– Основы системологии: Учебн. Пособие. Харьков: Изд-во Харьк. техн. Ун-та радиозлектр., 1998, ISBN 5-7763-92217.

Надійшло у редколегію 15.12.09