

РЕИНЖИНИРИНГ ГРАФІЧНИХ БАЗ ДАНИХ У СЕРЕДОВИЩІ ВІДКРИТОЇ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ BRL-CAD. МОДЕЛЮВАННЯ ПОВЕДІНКОВОЇ ЧАСТИНИ

С. С. Великодний, Ж. В. Бурлаченко, С. С. Зайцева-Великодна

Одеський державний екологічний університет

вул. Львівська, 15, корп. 1, каб. 330, м. Одеса, 65016, Україна. E-mail: velykodniy@gmail.com

Виконано моделювання наступних поведінкових діаграм: варіантів використання, станів, послідовності та діяльності. При формуванні проектної архітектури було використано розширену нотацію UML 2.5 та CASE-інструментарій Enterprise Architect 14.0. Дослідження проводяться, оскільки зараз став проявлятися особливий інтерес до комп'ютерної графіки, а також у зв'язку з інтенсивним розробленням та впровадженням систем автоматизованого проектування (САПР) у різних галузях виробництва та навчання. В результаті досліджень було виявлено, що висока швидкість рендерингу надає широкі перспективи для застосування САПР BRL-CAD у різноманітних галузях: військових, промислових чи навчальних застосуваннях. Потужний бік системи BRL-CAD – це надзвичайна швидкість засобів візуалізації, трасувальника променів та рендерингу. Виконані дослідження реінжинірингу ГБД важливі, оскільки дозволяють удосконалити САПР BRL-CAD шляхом високорівневого мовного оновлення.

Ключові слова: реінжиніринг, графіка, трасування, рендеринг, моделювання, код, діаграма, сутність.

РЕИНЖИНИРИНГ ГРАФИЧЕСКИХ БАЗ ДАННЫХ В СРЕДЕ ОТКРЫТОЙ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ BRL-CAD. МОДЕЛИРОВАНИЕ ПОВЕДЕНЧЕСКОЙ ЧАСТИ

С. С. Великодний, Ж. В. Бурлаченко, С. С. Зайцева-Великодная

Одесский государственный экологический университет

ул. Львовская, 15, корп. 1, каб. 330, г. Одесса, 65016, Украина. E-mail: velykodniy@gmail.com

Выполнено моделирование следующих поведенческих диаграмм: вариантов использования, состояний, последовательности и деятельности. При формировании проектной архитектуры была использована расширенная нотация UML 2.5 и CASE-инструментарий Enterprise Architect 14.0. Исследования проводятся, поскольку сейчас стал проявляться особенный интерес к компьютерной графике, а также в связи с интенсивной разработкой и внедрением систем автоматизированного проектирования (САПР) в различных отраслях производства и обучения. В результате исследований, было установлено, что высокая скорость рендеринга предоставляет широкие перспективы для применения САПР BRL-CAD в различных областях: военных, промышленных или учебных приложениях. Сильная сторона системы BRL-CAD – это чрезвычайная скорость средств визуализации, трассировщика лучей и рендеринга. Выполненные исследования реинжиниринга ГБД важны, поскольку позволяют усовершенствовать САПР BRL-CAD путем высокоуровневого языкового обновления.

Ключевые слова: реинжиниринг, графика, трассировка, рендеринг, моделирование, код, диаграмма, сущность.

АКТУАЛЬНІСТЬ РОБОТИ. В наш час є велика кількість програмних засобів, які виконують значну кількість спеціалізованих задач. Деякі з них прив'язані лише на одну галузь промисловості, інші – застосовуються у великій кількості, але тенденція йде шляхом спеціалізації програмних продуктів у цілому.

Однією з важливих складових частин систем автоматизованого проектування (САПР) є комп'ютерна графіка, що являє собою сукупність засобів та прийомів, за допомогою яких здійснюється введення, перетворення та виведення зі спеціалізованих середовищ графічної інформації.

Особливий інтерес до комп'ютерної графіки став проявлятися у зв'язку з інтенсивним розробленням та впровадженням у даний час САПР не тільки у машинобудуванні, приладобудуванні, радіоелектроніці, дизайні інтер'єрів, але і в інших галузях виробництва та навчання.

Відмінною складовою завдань комп'ютерної графіки є обробка графічних баз даних (ГБД), які, по суті являють собою «звичайні» БД, але в основу яких закладено математичні алгоритми відновлення зображення за сформованими статистичними коор-

динаційними даними. Такі можливості є далеко не у кожній САПР, але сучасні тенденції просто вимагають цього. Велика кількість програмних продуктів розроблюється з широким спектром моделюючих характеристик, BRL-CAD – це одна з таких САПР.

BRL-CAD – це спеціалізована крос-платформова система з відкритим кодом. Вона являє собою потужну 3D САПР для моделювання об'ємних тіл методами CSG. Ця САПР включає в себе інтерактивний геометричний редактор, паралельне трасування променів, рендеринг та геометричний аналіз.

BRL-CAD розроблялася близька 40 років та набула застосування у збройних силах США. Увесь BRL-проект працює із вихідного коду, а тому його можна використовувати на будь-яких платформах: GNU/Linux, MacOS, Solaris та Windows.

Об'єктом роботи є відкриті ГБД, що редагуються засобами BRL-CAD.

Предметом роботи є моделювання поведінкової частини реінжинірингу ГБД.

Проблема в загальному вигляді. Графічна інформація – це найбільш ємне і наочне уявлення великого обсягу інформації, однак, практичне застосування машинної графіки довгий час стримувалось відсут-

ністю відповідного обладнання та математичного забезпечення.

Логічність та формалізованість комп'ютерних моделей дозволяє виявити основні фактори, що визначають властивості досліджуваного об'єкта-оригіналу (або цілого класу об'єктів), зокрема, досліджувати відгук фізичної системи, що моделюється на зміни її параметрів та початкових умов.

Побудова комп'ютерної моделі базується на абстрагуванні від конкретної природи явищ або досліджуваного об'єкта-оригіналу і складається з двох етапів: спочатку створення якісної, а потім і кількісної моделі. Само комп'ютерне моделювання полягає у проведенні серії обчислювальних експериментів на ПК, метою яких є аналіз, інтерпретація та зіставлення результатів моделювання з реальною поведінкою об'єкта дослідження та, за необхідністю, подальше уточнення моделі.

Розв'язання поставлених проблем неможливе без глибокого проникнення в фізичну сутність досліджуваних явищ, розробки та вдосконалення відповідних теоретичних положень, впровадження досягнутих результатів у виробництво. Геометричні методи давно та успішно використовуються в багатьох галузях промисловості. Велику роль тут мають відіграти нові методи геометричного моделювання та їх реалізація в системах комп'ютерної графіки, що дозволить розв'язувати задачі спеціальних дисциплін.

Зв'язок із науковими завданнями, планами та програмами. Проектування об'єктів машинобудування, промислового, цивільного будівництва та радіоелектроніки вступає в новий етап свого розвитку, коли разом зі зростанням складності проектів мають забезпечуватися скорочення термінів проектування й зменшення числа проєктувальників значною мірою за рахунок автоматизації проектування й комп'ютеризації інженерної праці.

Аналіз досліджень і публікацій. При виконанні аналізу, особливу увагу, у науковому сенсі, було приділено методам, що застосовуються при роботі із складними графічними об'єктами.

Нижче розглянуто основні наукові геометричні методи, що використовуються при графічному моделюванні об'єктів, процесів, явищ в техніці та тенденції їхнього розвитку [1]. За напрямком наукового дослідження обирається твердотільне моделювання об'єктів, що утворюються (та змінюються в часі) під впливом різних зовнішніх чинників.

Комп'ютерне моделювання є одним з ефективних методів вивчення будь-яких складних систем, що піддаються візуалізації. Комп'ютерні моделі простіше і зручніше досліджувати у силу їх можливості проводити так звані обчислювальні експерименти, у тих випадках, коли реальні експерименти ускладнені через фінансові або фізичні перешкоди або можуть дати непередбачуваний результат [2].

Тривимірний опис об'єкта (англ.: 3D) – це представлення об'єкта у трьох просторових вимірах. Як правило, ці виміри представлені у вигляді координат X , Y , та Z . Можливо мати дані з ідентичними координатами X та Y при відмінній координаті Z . Наприклад, для цифрового представлення океанічних потоків, використовують 3D [3].

Твердотільне моделювання – це найдосконаліший і достовірний метод створення копії реального об'єкта, природний спосіб вираження сутності виробу [4].

Рендеринг (англ.: rendering – відтворення, відрисовування) в комп'ютерній графіці – це процес отримання зображення за моделлю за допомогою комп'ютерної програми [5]. Тут модель – це опис тривимірних об'єктів (3D) суворо визначеною мовою або у вигляді структури даних. Такий опис може містити геометричні дані, положення точки спостерігача, інформацію про освітлення. Зображення – це цифрове растрове зображення. Зазвичай під рендерингом розуміють накладення текстури на уже готову твердотільну модель (solid-works) у машинобудуванні [5] та на каркас (framework) в інженерній графіці [6].

Трасування променів (англ.: ray tracing) у комп'ютерній графіці є способом створення зображення тривимірних об'єктів чи сцени за допомогою відстеження ходу променя світла крізь точку екрану і симуляції взаємодії цього променя з уявними об'єктами, що підлягають відображенню [6]. Цей спосіб дозволяє створювати надзвичайно реалістичні зображення, зазвичай значно вищої якості, ніж дає типовий алгоритм Scanline або ж метод відбивання променів (англ.: Ray casting), проте має значно вищу обчислювальну складність. Із цієї причини алгоритми трасування променів використовуються там, де немає суттєвих обмежень часу рендерингу.

Граничне подання – це опис меж об'єкту або абсолютного аналітичного завдання граней, що описують тіло [7]. Цей метод дозволяє створити якісне зображення геометричного твердого тіла, щоб встановити взаємну відповідність, потрібно задати координати або контури об'єктів, а також ескізи різних видів об'єктів, і вказати лінії зв'язків між даними видами.

Методи визначення складних контурів та векторизації растрових моделей було розглянуто у [8].

Одним з головних переваг тривимірного моделювання є швидке формування креслень. Використовувати результати моделювання можна і на подальших стадіях розробки виробів – це є ще однією перевагою твердотільного моделювання [9].

Основні принципи відкритого програмного забезпечення. Відкрите програмне забезпечення (англ.: open-source software) – це забезпечення, для якого є доступним (вихідний) програмний код, що забезпечує найкращі умови для вивчення такого програмного забезпечення та можливого подальшого внесення змін (удосконалень тощо) до нього [10].

Досить часто це поняття вважають тотожним вільному ПЗ, що не є абсолютно правильним. Найістотніша відмінність полягає в тому, що ліцензії на вільне ПЗ обумовлюють, що усі подальші модифіковані версії такого ПЗ теж повинні розповсюджуватись як вільні, в той час як більшість ліцензій на ПЗ з відкритими кодами надають повну свободу авторам модифікованих версій. В результаті вільне ПЗ завжди є ПЗ з відкритими вихідними кодами, але зворотне є вірним далеко не завжди.

Задачам створення програмних комплексів на базі відкритого вихідного коду, мовами «С» та «С++», що цікавлять нас з погляду реінжинірингу ГБД присвячено публікацію [11].

Наведемо визначальні характеристики, що стосуються відкритого ПЗ та технологій проектування ГДБ.

Сирцевий код (зазвичай просто «сирці», також «вихідний код», «програмний код», «джерельний код», «текст програми», англ.: «source code») – будь-який набір інструкцій або оголошень, написаних мовою програмування і у формі, що її може прочитати людина. Сирцевий код дозволяє програмісту спілкуватися з комп'ютером за допомогою обмеженого набору інструкцій.

Сирцевий код програми – це набір файлів, потрібних для перетворення з форми, доступної для читання людині, на деякі види комп'ютерного виконаного коду. Можливі два напрямки виконання сирцевого коду: транслювання у машинний код за допомогою компілятора (призначений для певної комп'ютерної архітектури) або виконання безпосередньо з тексту за допомогою інтерпретатора.

Одна з перших САПР з такими характеристиками з'явилась тому, що у 1979 році балістична науково-дослідна лабораторія армії США (U. S. Army Ballistic Research Laboratory (BRL), зараз – United States Army Research Laboratory, висловила потребу в інструментах, які могли б допомогти у комп'ютерному моделюванні та інженерному аналізі бойових систем та їх умов роботи.

Коли жодна з САПР, які існували на той час, виявились неготовими для досягнення цієї мети, розробники з BRL почали збирати набір утиліт, що здатні на інтерактивне переглядання та редагування дерев геометричних моделей. Цей пакет став відомий як BRL-CAD. Перший публічний реліз був зроблений у 1984 р. У грудні 2004 р. BRL-CAD став проектом із відкритим кодом. Дуже важливо, що BRL-CAD розповсюджується на умовах ліцензій *BSD та GNU LGPL.

З того часу ця САПР постійно розвивається, з'являються нові можливості, проте зараз вже само лінгвістичне забезпечення подання ГБД (мова «С») у середовищі САПР BRL-CAD потребує переходу (реінжинірингу) на високорівневі мови («С++» чи «С#»).

Мета поданої статті полягає у створенні, підключенні та реінжинірингу (еволюційному удосконаленні) графічних баз даних, як композиційного компоненту відкритої САПР BRL-CAD.

МАТЕРІАЛ І РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ. Для наочності сприйняття аналітиками та системними архітекторами, представимо реінжиніринг ГБД у відкритій САПР BRL-CAD із використанням методології UML із розширеною нотацією 2.5.

Діаграма варіантів використання. При побудові діаграми варіантів використання (ДВВ) розглянемо BRL-CAD. Побудована ДВВ для САПР BRL-CAD наведена на рис. 1. Специфікація ДВВ наведена у табл. 1.

Почнемо розгляд ДВВ. САПР BRL-CAD складається з 2-х основних модулів – бізнес-прецедентів (business use case) «Archer» та «MGED».

Таблиця 1 – Специфікація ДВВ

Тип	Різновид	Кількість, шт.
сутності	бізнес-актор (business actor)	1
	бізнес-прецеденти (business use case)	3
	варіанти використання (use case)	7
	границі використання (boundary)	2
зв'язки	включення (include)	2
	підключення (subscribe)	1
	розширення (extend)	5
	реалізація (realize)	8

У зв'язку із тим, що відштовхуватися будемо від того, що реінжиніринг відбувається із повним списком модулів, то за бізнес актора (business actor) ми візьмемо сам «BRL-CAD» – це відразу вказуємо у специфікаціях, тобто: бізнес актор – це актор, який безпосередньо задіяний у роботі з клієнтом.

Модуль «MGED» – це бізнес-прецедент (business use case) є основним для цього програмного продукту. У ньому відбувається створення твердотільних моделей та трасування променів.

Точно так саме і модуль «Archer», мають у собі стереотип бізнес типу (business use case) тому, що за допомогою цих двох модулів саме й виконується проектування ГБД (рис. 1).

Першим рівнем прецедентів є прецеденти з умовною назвою «Можливості», що відокремлені відповідною границею використання (boundary).

Кожен стереотип першого рівня реалізований в якомусь вигляді і завдання полягає в тому, щоб правильно показати цей вид. При запуску «MGED.exe» відображаються два інтерфейси: консольний (console) і графічний (GUI). У кожному з цих вікон знаходиться панель команд, з якої виконуються команди трасування.

Основні можливості (рис. 1) або прецеденти першого рівня (use case) зведено у табл. 2. Всі інші модулі підключаються тільки за допомогою модулів з сайту програми, але їх немає у вільному доступі. Прецеденти другого рівня – це функції (рис. 1), що відокремлені відповідною границею використання (boundary), які реалізують функціональні можливості (табл. 2).

Через консоль можлива реалізація простих фігур, створення з них складних і подальша робота з ієрархічним деревом. Прості фігури задаються за допомогою команд, можливо вказати їх чітке місце розташування. Складні фігури так само задаються за допомогою команд, але за деяких умов – мінімально дві прості фігури відображено на екрані і друге – ці фігури стикаються. Після цього можливий перегляд ієрархічного дерева і робота з ним.

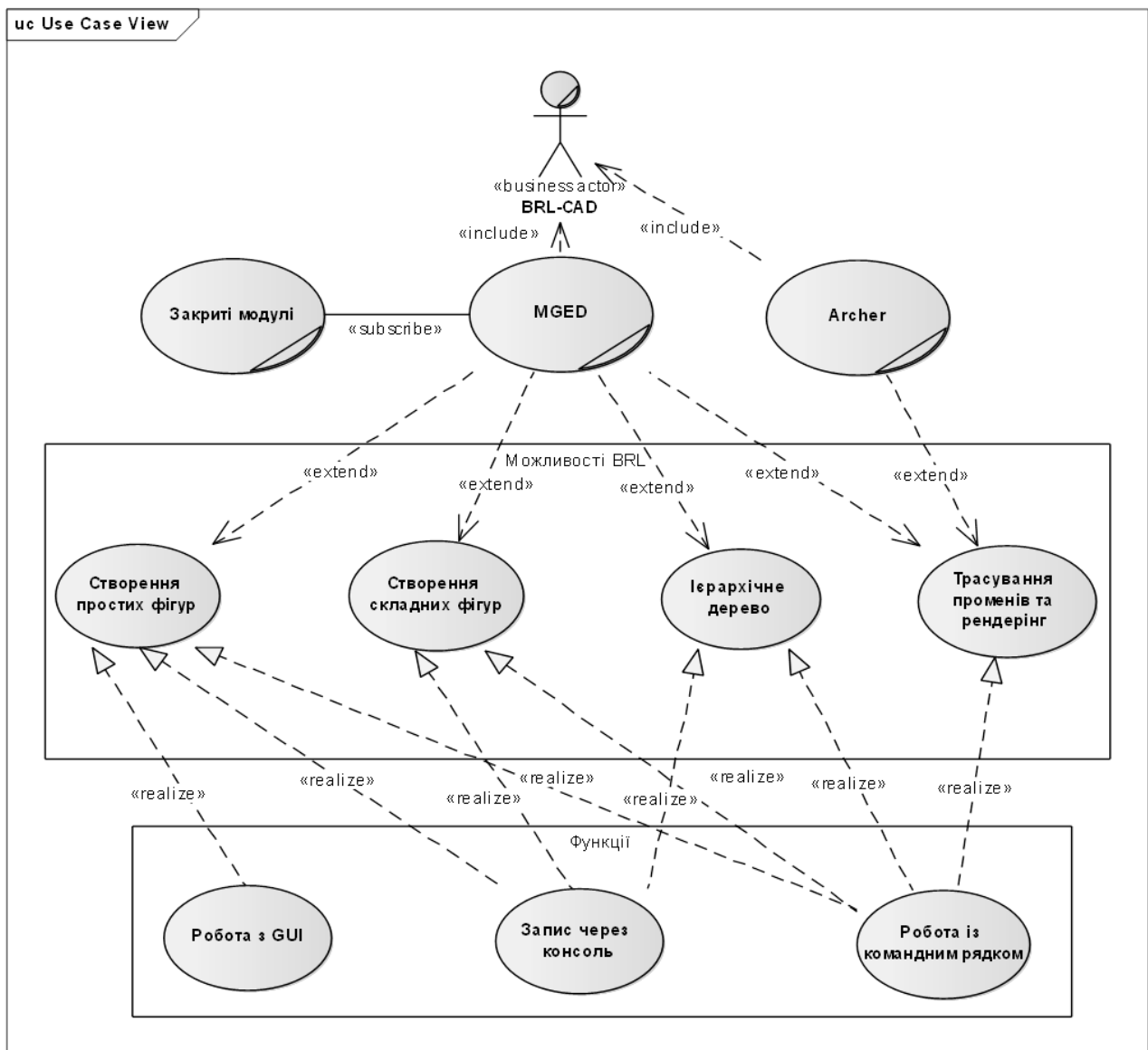


Рисунок 1 – ДБВ для САПР BRL-CAD

Таблиця 2 – Ієрархія прецедентів у BRL-CAD

Прецеденти:	
першого рівня (можливості):	другого рівня (функції):
«Створення простих фігур»	«Робота з GUI» (графічний інтерфейс)
«Створення складних фігур»	«Запис через консоль» (Console)
«Ієрархічне дерево»	«Робота із командним рядком» (Command Line)
«Трасування променів та рендеринг»	

Можливості графічного інтерфейсу досить обмежені (можливо тільки розміщення простих фігур так би мовити «на око»), зміни кута перегляду і можливість переглянути результат трасування променів.

Всі функції які були описані в перших двох прецедентах так само можливо виконати через командний рядок. Але він створювався не для цього. Основною його функцією є підготовка ієрархії до ренде-

рингу, трасування променів та загальної обробки графіки. У налаштуваннях командного рядка можна задавати щільність тіла, кут променів, що падають на фігуру та більше 10-ти констант, які використовуються у класичному твердотільному моделюванні.

Реалізація результатів роботи командного рядка відображаються у графічному інтерфейсі.

Так саме до «MGED» підходить один прецедент (зв'язок-підключення (subscribe)) – «Закриті модулі», який теж має бізнес стереотип бізнес-прецеденти (business use case). Це спеціальні модулі для розширення можливостей проектування, які не можна дістати у вільному режимі, тому ми не будемо їх детально розглядати.

Крім «MGED» існує також модуль «Archer», який створений для більш зручної роботи з графічним відображенням і переглядом твердотільної моделі. Від нього йде тільки один прецедент (use case) з пакету «Можливості» – це «Трасування променів та рендеринг».

Також у «Archer» знаходиться графічний інтерфейс, консоль та командний рядок, які є прецедентами з пакету «Функції».

Далі опишемо зв'язки між ними. Почнемо із гори (рис. 1): «BRL-CAD» – це загальна назва САПР та модулі «Archer» й «MGED» є її складовими, тому між ними присутній зв'язок типу «включення» (include).

Всі прецеденти першого рівня – це можливості, вони є розширенням роботи модулів і тому між ними стоїть тип зв'язку «розширення» (extend).

Між першим і другим рівнем прецедентів, можливостями і функціями може знаходитися тільки один рівень зв'язку – «реалізація» (realize), оскільки функції реалізують можливості.

У ході виконання поданого підрозділу – була спроектована ДБВ для САПР BRL-CAD, що є концептуальною основою для виконання реінжинірингу ГБД.

Діаграма станів. Спроекуємо діаграму станів (ДС) реінжинірингу ГБД у середовищі САПР BRL-CAD (рис. 2).

На цій діаграмі відображаються стани, в яких може знаходитися програмний продукт. Основним логічним ланцюжком цієї діаграми є те, що існує «Початкова точка» (initial state) та «Кінцева точка» (final state), які символізують початок і завершення роботи з середовищем.

Відразу додамо, що початком роботи з програмним продуктом буде його «активація», тобто запуск, а закінчення роботи – це деактивація або закриття нашого продукту.

При активації роботи запускається буфер і компілятор. Буфер – це стек для команд обробки фігур, робота з якими відбувається у даний момент. Компілятор – це спеціальний модуль, що займається обробкою уводів команд та побудовою графічного результату на екрані після обробки цих команд.

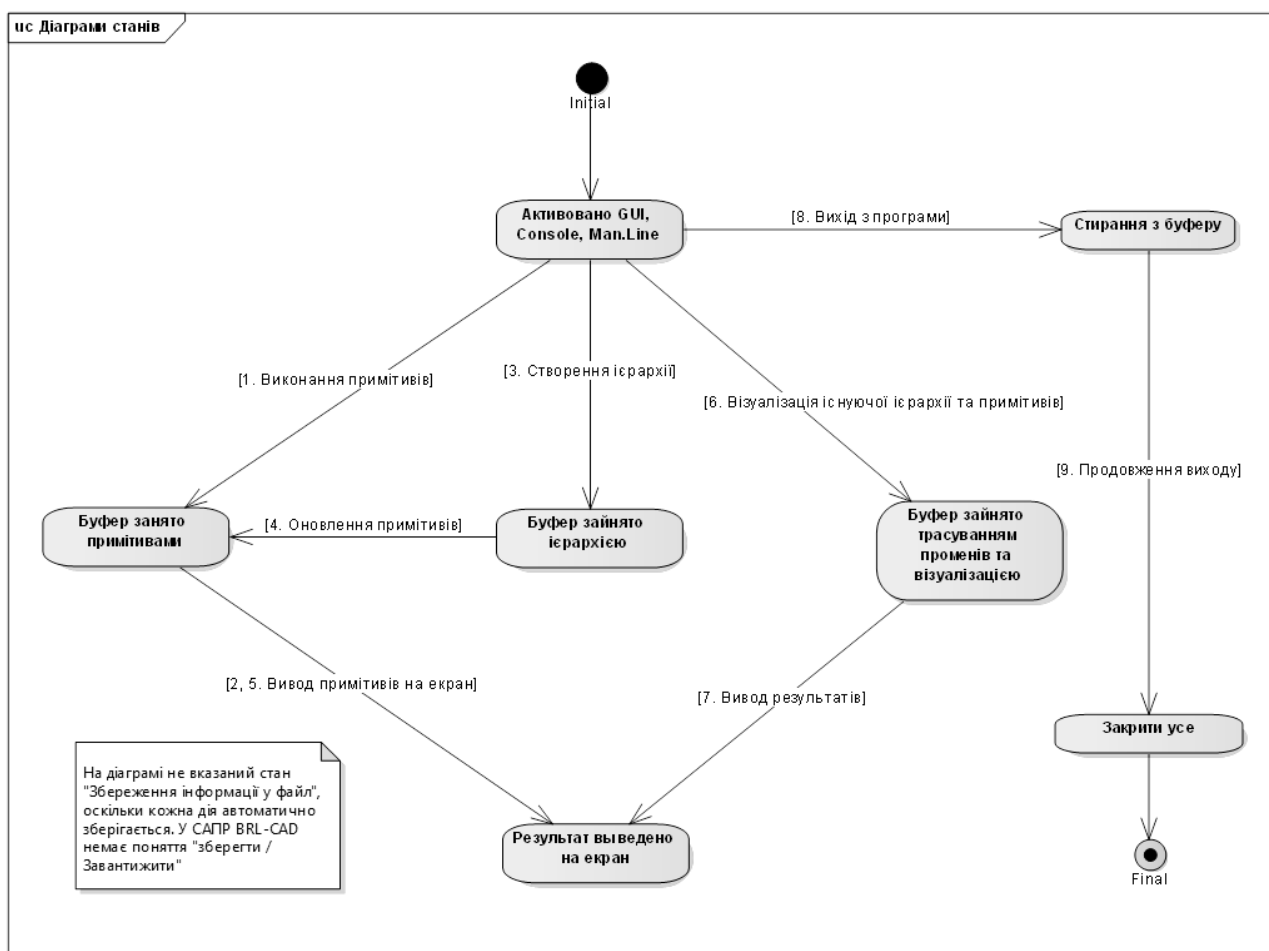


Рисунок 2 – ДС реінжинірингу ГБД у середовищі САПР BRL-CAD

Якщо дотримуватися стандартної методології реінжинірингу проекту: першою дією є створення примітиву при застосуванні можливості вивантажити у оновлений проєкт сформовані раніше ГБД.

Для цього з консолі вводиться команда створення певного виду примітиву, його розмір, положення (якщо становище не зазначено, то примітив створюється на початку координат). Далі відбувається наступний процес: буфер отримує певну інформацію, яку зберігає до тих пір, поки її обробляє компілятор.

Після того як компілятор обробив всю інформацію за поточною командою, при вірному результаті він видає дані на екран і записує їх у постійну пам'ять. Після того, як створено кілька примітивних фігур, приходить час будувати ієрархію з них, тобто створювати більш складні фігури.

Умовами створення більш складних фігур, як було вказано раніше є їх кількість (більше двох) та зіткнення (перетини). Як тільки команда про побудову складної фігури надходить до буферу, відбувається

обробка компілятором, але, тепер відбувається не тільки виведення на екран результату, а й заміна та оновлення даних з постійної пам'яті. Буфер після завершення команд автоматично очищується.

Наступним стандартним кроком після створення складних фігур є їх візуалізація. Для цього треба провести трасування променів: або з консолі, або з командного рядка задати фактуру: з чого складається об'єкт та ще десяток стандартних фізичних констант, після чого, необхідно зайти до спеціальної панелі командного рядка та виконати трасування. Процедура – аналогічна: «буфер-компілятор-відображення».

Єдиною відмінністю буде те, що результат не зберігається у постійну пам'ять, а тільки відображається на екрані. Для запису візуалізації використовується інший модуль під назвою «Archer».

На візуалізації і завершується робота з даним програмним продуктом і тому наступним дією є «деактивація» (final) (рис. 2).

Слід особливо виділити той факт, що у поданому програмному продукті є деякі особливості, які дозволяють виділяти його серед інших. Тут немає такого поняття як «Зберегти проект / Завантажити проект». Будь-яка дія оброблюється компілятором та автоматично зберігається у постійну пам'ять.

У такому підході є і як переваги так і недоліки.

Переваги: при випадках екстреного закриття програми вся інформація зберігається.

Недоліки: якщо зроблена помилка, то виправляти її доводиться за допомогою видалення всієї спроектованої фігури.

Після побудови ДС та виходячи з аналізу її змісту, можна зробити проміжні, але досить несподівані висновки, а саме: оскільки САПР BRL-CAD, що згодом набула статусу відкритої, створювалась для потреб американських збройних сил, то впливає припущення, що їх уподобанням була саме така оригінальна реалізація принципів роботи із САПР.

Діаграма послідовності. Діаграму послідовності (ДП) спроектуємо для того, щоб показати, які послідовні дії виконує програміст при роботі із середовищем BRL-CAD. На ДП згори до низу розставляються повідомлення (рис. 3), кожне може бути позначено відповідним ім'ям. За бажанням на ДП можна показати аргументи та деяку керуючу інформацію, також можна показати рефлексивні повідомлення, які об'єкт посилає самому собі, при цьому стрілка повідомлення вказує на ту ж саму лінію життя.

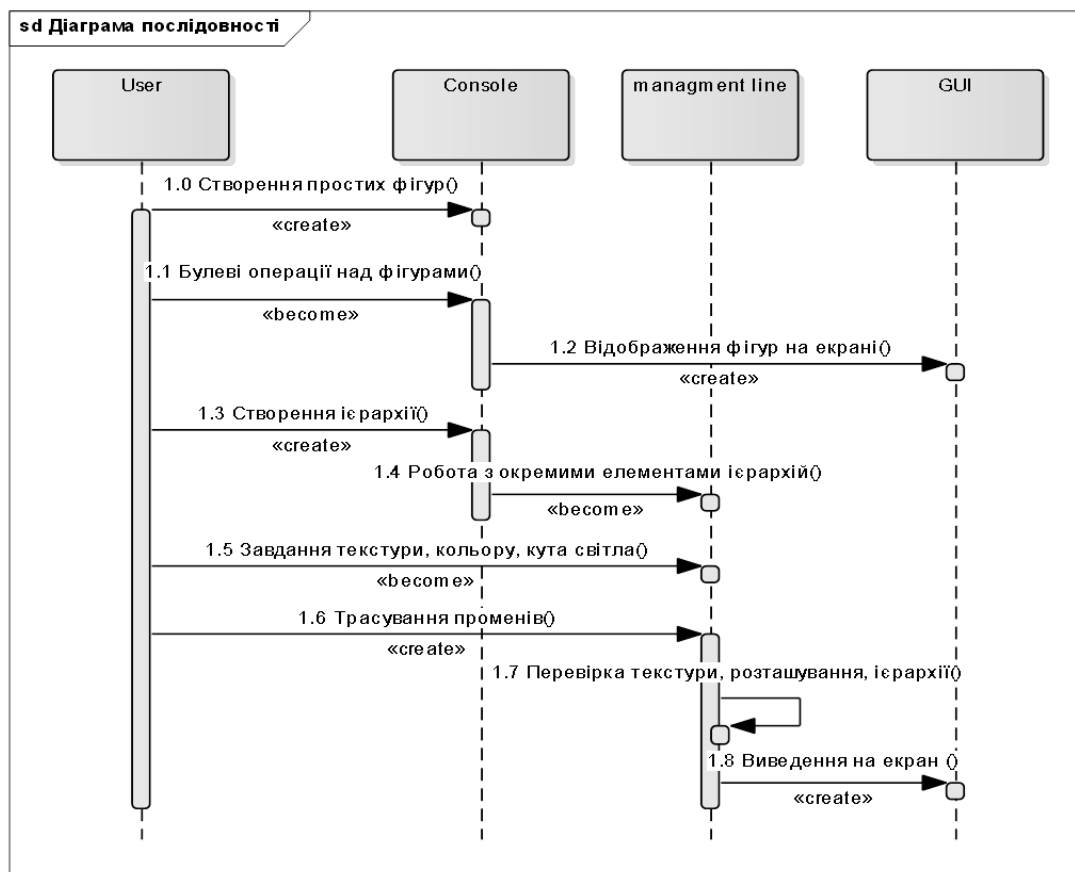


Рисунок 3 – ДП реінжинірингу ГБД у середовищі САПР BRL-CAD

Пояснимо кожен окрему сутність ДП.

Об'єкт або учасник (Object, Participant) позначається прямокутником, у якому зазначається інформація про учасника дій. Це, як правило, назва об'єкта та його клас, розділені двокрапкою. Наприклад:

«saveButton» або «saveButton: JButton» або «: JButton». Тобто, назву класу можна опустити або навпаки – не вказувати назву об'єкта, але щось одне з двох (об'єкт або клас) необхідно вказати.

Розташовуються об'єкти (як правило) вздовж верхнього краю діаграми. Від прямокутника вниз опускається «Лінія життя» (life line) – відрізок, який позначає відведений об'єкту час життєвого циклу. Позначається вона пунктирною лінією.

У спроектованій ДП, стикаємося із розглянутими раніше модулями інтерфейсу: користувач «User»; консоль «Console»; командний рядок «Management line»; графічний інтерфейс «GUI». У деяких ДП, лінії життя обривають – для цього існує спеціальний знак. В нашому випадку – це не потрібно.

Активізація або фрагмент виконання (Activation Bar, Execution Occurances) позначається вузьким прямокутником, що розташовується на лінії життя. Він вказує на початок та завершення дії, у якій бере участь об'єкт. Оскільки лінія життя – це метафора часу, то прямокутник на лінії життя вказує на активізацію об'єкта продовж відносного часу.

Повідомлення (Message) – це стрілка від одного життя до іншого, що показує взаємодію об'єктів. Повідомлення бувають різні й відрізняються. Синхронне повідомлення позначається зафарбованою стрілкою, асинхронне – штриховою. Повернення показується пунктирною стрілкою, у зворотному напрямку.

У наведеній ДП (рис. 3), спроектовані фундаментальні кроки реінжинірингу, які необхідно пройти при роботі із ГБД. Оскільки спроектована ДП багатою мірою розповідає сама за себе у описаній методології послідовності, то не будемо описувати її у відповідному підрозділі статті, що б не викликати дублювання проілюстрованої інформації. Отже, у виконаному підрозділі статті було розроблено ДП для роботи із ГБД у середовищі САПР BRL-CAD, яка містить у собі поведінкову сценарну послідовність реінжинірингу ГБД.

Діаграма діяльності. Діаграма діяльності (ДД) – це спеціальна діаграма, на якій показано розкладання реінжинірингу на його складові частини. Під діяльністю (англ.: activity) розуміється специфікація поведінки, що виконується, у вигляді координованого послідовного чи паралельного виконання підлеглих елементів: вкладених видів діяльності та окремих дій (англ.: action), що з'єднані між собою потоками, які йдуть від виходу одного вузла до входу іншого.

У нашому випадку спроектовано ДД для створення ГБД у середовищі САПР BRL-CAD, яку представлено на рис. 4. Сутності, що містить ДД, зведено у табл. 4.

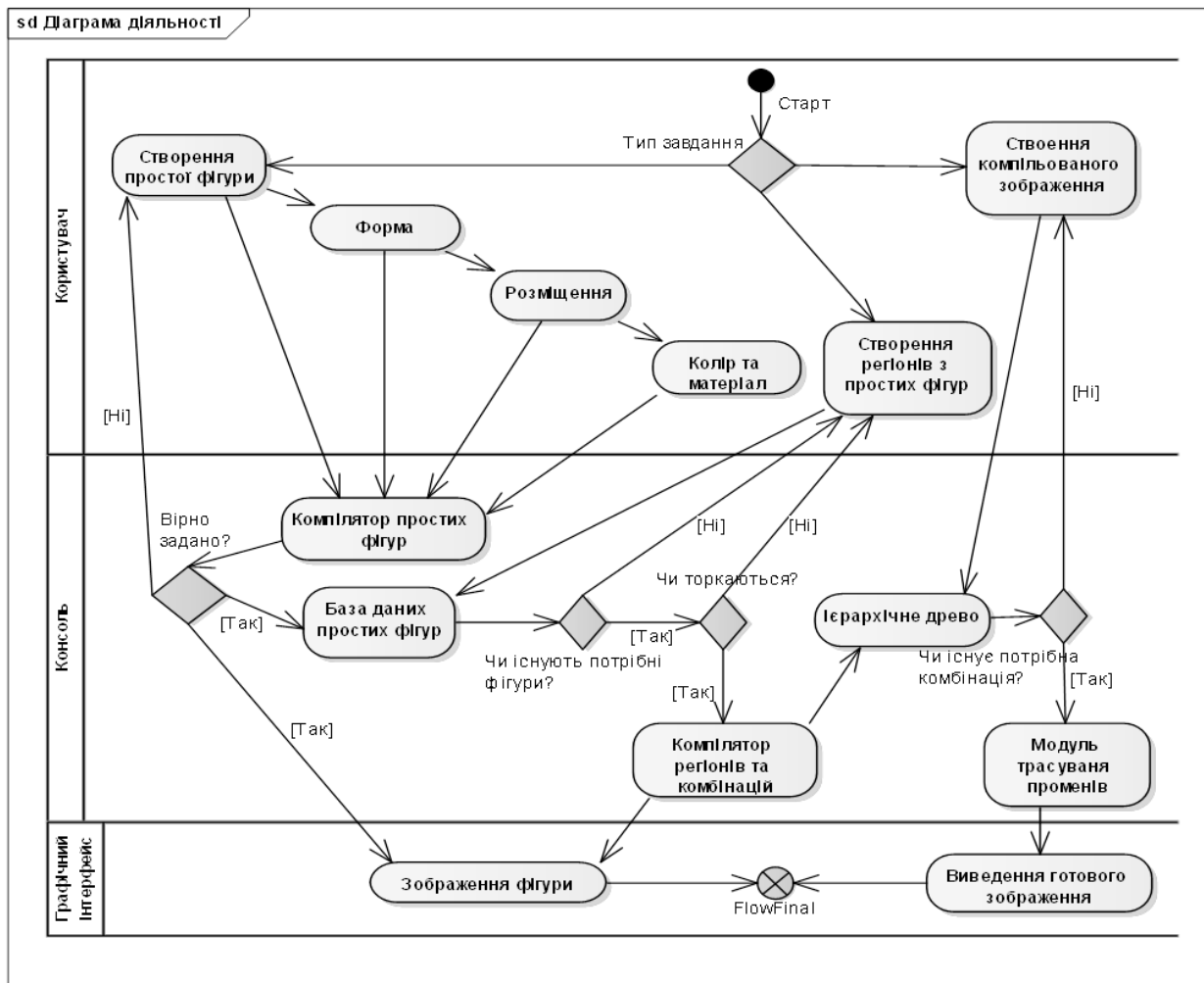


Рисунок 4 – ДД створення ГБД у середовищі САПР BRL-CAD

Таблиця 4 – Сутності ДД

Різновид сутності	Кількість, шт.
початкова активність (initial)	1
блоки-вирішення (decision)	5
стани активності (activity)	13
кінцева активність (flow final)	1

У поданій ДД, крім минулих задіяних, створено нові сутності-діяльності (activity), що розміщено на доріжках розділення діяльності (partition) або ДРД (табл. 5).

Таблиця 5 – Сутності, що розміщено на ДРД

Назва ДРД	Діяльність
«Користувач»	«Старт» (початкова активність)
	«Тип завдання» (блок-вирішення)
	«Створення компільованого зображення»
	«Створення простої фігури»
	«Форма»
	«Розміщення»
	«Колір та матеріал»
«Консоль»	«Створення регіонів з простих фігур»
	«Компілятор простих фігур»
	«База даних простих фігур»
	«Вірно задано?» (блок-вирішення)
	«Чи існують потрібні фігури?» (блок-вирішення)
	«Чи торкаються?» (блок-вирішення)
	«Чи існує потрібна комбінація?» (блок-вирішення)
	«Ієрархічне дерево»
«Модуль трасування променів»	
«Графічний інтерфейс»	«Компілятор регіонів та комбінацій»
	«Зображення фігури»
	«Виведення готового зображення»; Кінець потоку активності (flow final).

Опишемо структуру потоків керування на ДД (рис. 4).

За базовий візьмемо звичайний початок процесу – створення простої фігури. У наведеній діаграмній нотації, щоб створити просту фігуру, нам потрібно не тільки ввести команду назви, але й вказати «Форму», «Розміщення», «Колір та матеріал», як заплановано у ДД.

Після того, як указано дані параметри: вся інформація передається до компілятору простих фігур (на рівні фізичного модуля коду, компілятор один, але для наочності розділимо його на підрівні).

Особливість цієї діаграми – це блоки-вирішення (decision), які позначають перевірку виконання умови, і, у разі виконання цієї умови, проводять операції далі, в іншому випадку – повертає туди, куди вказує стрілка негативного потоку керування.

Отже, після того як інформація надходить до компілятору, відбувається саме така перевірка умови. Припустимо, що умови не виконуються, і тоді на екран видається напис за типом «Shape is not create» (фігуру – не створено) і відбувається відсилання до початку команди або переривається поточна. У разі якщо перевірку пройдено успішно, то виконується відразу дві основних дії: графічне відображення та надходження інформації до основного файлу, з усіма даними проекту.

Наступною операцією, після створення пари простих фігур, є створення складних фігур («регіонів»). При спробі створення регіонів відбувається процес перевірки умови, яка необхідна для створення складних фігур. Якщо всі вони виконані, то виконується вже не дві, а три основних дії: запис до пам'яті, відображення на екрані та, найголовніше для регіонів, – оновлення ієрархічного дерева виведення складних фігур.

Тепер, після створення складних ієрархічних фігур, можна виконувати трасування зображення. У розглянутому прикладі все відбувається у програмному модулі «MGED», оскільки на даному етапі реінжинірингу нас цікавить найшвидший, а не презентаційний варіант.

Трасування променів також відбувається після перевірки виконання деяких умов. Хоч можна трасувати й примітиви, але це не практикується у силу марності такої дії, оскільки крім складних складових фігур, потрібна перевірка за фізичними константами. Для правильного трасування необхідно вказати: кут падіння світла, рівень світла, що пропускається, матеріал з чого зроблена фігура тощо. Коли всі ці параметри зазначено вірно: буде отримана цілком придатна для використання зображення (складова ГБД).

Отже, у даному підрозділі було спроектовано ДД для створення ГБД у САПР BRL-CAD. Ця діаграма дозволяє визначати: стани, шляхи та цикли проходження процесу реінжинірингу ГБД.

ВИСНОВКИ. САПР BRL-CAD є прийнятною у застосуванні для досвідченого проектувальника, проте для початківця або студента процес її застосування виявиться дуже ускладненим. У глобальній мережі немає жодних матеріалів українською або російською мовами, які хоча б поверхнево описували роботу із системою у режимі «інструкція користувача». Матеріали англійською мовою – поверхневі та містять тільки декілька десятків консольних команд.

При детальній аналізі середовища було виявлено наявність двох модулів, що містяться у структурі САПР, які допомагають потенційному користувачеві системою швидко конструювати необхідні ГБД.

Також фундаментальною властивістю пакету можна назвати здатність підтримувати конструювання та аналіз візуальних моделей на основі складних об'єктів, що складаються з великого набору графічних примітивів.

В цілому, при написанні поданої статті, було досягнуто встановлену мету, яка полягала у створенні, підключенні та подальшому реінжинірингу ГБД, як композиційного компоненту відкритої САПР BRL-CAD.

Після проведених досліджень можна зробити висновки: потужний бік системи – це надзвичайна швидкість засобів візуалізації, трасувальника променів та рендерингу. Після порівняння з аналогами, можна стверджувати, що процес візуалізації є одним із найшвидших серед існуючих.

Остання перевага надає широкі перспективи для застосування САПР BRL-CAD у різноманітних галузях: військових, промислових чи навчальних застосуваннях, таких як системи проектування та аналізу у машинобудуванні, механічні вузли, архітектурні споруди, будова молекул тощо.

Передумовою виконання реінжинірингу було моделювання проектного каркасу (системної архітектури) на основі якої буде складено майбутні технологічні засади створення ГБД для підключення до оновленої відкритої САПР BRL-CAD.

Результатом моделювання стало створення поведінкових діаграм: варіантів використання, станів, послідовності та діяльності. При формуванні проектної архітектури було використано розширену нотацію UML 2.5 та CASE-інструментарій Enterprise Architect 14.0.

Найближчою перспективою дослідження стане створення структурних діаграм: об'єктів, класів, компонентів, композитної структури, огляду взаємодії та розгортання, як незамінних при виконанні реінжинірингу структурно-статичних моделей ГБД відкритої САПР BRL-CAD.

Автори хочуть виразити велику подяку корпорації BRL-CAD за можливість відкритого користування і тестування вихідних файлів та плагінів й підтримку крос-платформної методології.

ЛІТЕРАТУРА

1. Великодний С. С. Методи реінжинірингу програмних систем. *Технологии приборостроения*. 2014. Спец. вып. С. 65–68.
2. Норенков И. П. Автоматизированное проектирование. Москва: МГТУ им. Н. Э. Баумана, 2000. 188 с.

3. Ли Дж., Уэр Б. Трёхмерная графика и анимация. Изд. 2-е. Москва: Вильямс, 2002. 640 с.

4. Концевич В. Г. Твердотельное моделирование машиностроительных изделий в Autodesk Inventor. Киев, Москва: ДиаСофтЮП, ДМК Пресс, 2007. 672 с.

5. Херн Д., Бейкер М. П. Компьютерная графика и стандарт OpenGL. Изд. 3-е. Москва: Вильямс, 2005. 1168 с.

6. Энджел Э. Интерактивная компьютерная графика. Вводный курс на базе OpenGL. Изд. 2-е. Москва: Вильямс, 2001. 592 с.

7. Снук Г. 3D-ландшафты в реальном времени на C++ и DirectX 9. Изд. 2-е. Москва: Кулиц-пресс, 2007. 368 с.

8. Козир А. Є., Славко Г. В. Алгоритми і методи визначення складних контурів динамічних об'єктів з використанням технологій web-графіки. *Вісник Кременчуцького національного університету імені Михайла Остроградського*. 2016. Вип. 3 (98). Ч. 1. С. 20–26.

9. Потемкин А. В. Трёхмерное твердотельное моделирование. Москва: Компьютер-Пресс, 2002. 296 с.

10. Великодний С. С., Тимофеева О. С., Зайцева-Великодна С. С., Нямцу К. Є. Порівняльний аналіз властивостей відкритого, вільного та комерційного програмного забезпечення. *Інформаційні технології та комп'ютерна інженерія*. 2018. №1 (41). С. 21–27.

11. Жуковський В. В. Про деякі підходи до створення програмних комплексів комп'ютерного моделювання підземних процесів. *Вісник Кременчуцького національного університету імені Михайла Остроградського*. 2017. Вип. 2 (103). Ч. 1. С. 64–73.

GRAPHIC DATABASES REENGINEERING IN BRL-CAD OPEN SOURCE CAD ENVIRONMENT. MODELING OF THE BEHAVIOR PART

S. Velykodnyy, Zh. Burlachenko, S. Zaitseva-Velykodna

Odessa State Environmental University

vul. Lvivska, 15, ap. 330 (1), Odessa, 65016, Ukraine. E-mail: velykodnyy@gmail.com

Purpose. To implement creation, connection, and reengineering (evolutional improvement) of graphical databases (GDB) as a compositional component of BRL-CAD open source CAD. **Methodology.** Distinctive part of the tasks of computer graphics is the processing of the GDB, which, in essence, are "ordinary" databases, but which are based on mathematical algorithms for image reconstruction using established statistical coordination data. Not every CAD system has such capabilities, but current trends just require it. Large number of software products are developed with a wide range of simulations, and BRL-CAD is one such CAD. Computer models are easier and more convenient to explore due to their ability to carry out so-called computational experiments, in cases where real experiments are complicated due to financial or physical obstacles or can give unpredictable results. **Originality.** BRL-CAD CAD is constantly developing, new opportunities are emerging, but now the linguistic support of the GDB ("C" language) in the CAD environment requires BRL-CAD to be transferred (reengineered) into high-level languages ("C++" or "C #"). For clarity of analysts' and system architects' perception, the article presents the reengineering of GDB in the BRL-CAD open source CAD system using behavioral chart models. **Results.** After comparing with analogues, it can be argued that the rendering process is one of the fastest existing ones. The result of the simulation has been the creation of the following behavioral diagrams: use case, state machine, sequence and activity. In the design architecture, the extended UML 2.5 notation and the Enterprise Architect 14.0 CASE toolkit have been used. **Practical value.** After the conducted researches, a fundamental property of a package is the ability to support the construction and analysis of visual models based on

complex objects consisting of a large set of graphic primitives. High rendering speed provides broad prospects for the use of BRL-CAD CAD systems in various fields: military, industrial or training applications, such as engineering and analysis systems in mechanical engineering, mechanical engineering assemblies, architectural structures, molecular structure, etc. **Conclusions.** Computer simulation is one of the most effective methods for studying any complex systems undergoing visualization. The powerful side of the BRL-CAD system is the extraordinary speed of visualization, tracer and rendering. Finished GDB reengineering allows to upgrade BRL-CAD CAD through high-level language upgrade. References 11, tables 5, figures 4.

Key words: reengineering, graphics, tracing, rendering, modeling, database, code, diagram, entity.

REFERENCES

1. Velykodniy, S. S. (2014), "Metody reinzhynirnyngu programnyh system" [Methods of reengineering software systems], *Tekhnologii priborostroeniya [Instrumentation technology]*, Spec. nom., pp. 65–68.
2. Norenkov, I. P. (2000), "Avtomatizirovannoe proektirovanie" [Automated design], MGTU im. N. E. Baumana, Moscow, 188 p.
3. Li, Dzh., Uer, B. (2002), "Trekhnernaya grafika i animatsiya" [3D graphics and animation], Vil'yams, Moscow, 640 p.
4. Kontsevich, V. G. (2007), "Tverdotel'noe modelirovanie mashinostroitel'nykh izdeliy v Autodesk Inventor" [Solid modeling of machine-building products in Autodesk Inventor.], DiaSoftYuP, DMK Press, Kyiv, Moscow, 672 p.
5. Khern, D., Beyker, M. P. (2005), "Komp'yuternaya grafika i standart OpenGL" [Computer graphics and the standard OpenGL], Vil'yams, Moscow, 1168 p.
6. Endzhel, E. (2001), "Interaktivnaya komp'yuternaya grafika. Vvodnyy kurs na baze OpenGL" [Interactive computer graphics. OpenGL introductory course], Vil'yams, Moscow, 592 p.
7. Snuk, G. (2007), "3D-landshafty v real'nom vremeni na C++ i DirectX 9" [Real-time 3D landscapes in C++ and DirectX 9], Kudits-press, Moscow, 368 p.
8. Kozyr, A. Je., Slavko, G. V. (2016), "Algorytmy i metody vyznachennja skladnyh konturiv dynamichnyh ob'ektiv z vykorystannjam tehnologij web-grafiky" [Algorithms and methods for assigning folding contours of dynamic ob'ects to video technologies], *Visnyk KrNU im. M. Ostrograds'kogo [Scientific journal "Transactions of Kremenchuk Mykhailo Ostrohradskyi National University"]*, no. 3 (98), part 1, pp. 20–26.
9. Potemkin, A. V. (2002), "Trekhnernoe tverdotel'noe modelirovanie" [Three-dimensional solid modeling], Komp'yuter-Press, Moscow, 296 p.
10. Velykodniy, S. S., Tymofieieva, O. S., Zaitseva-Velykodna, S. S., Niamtsu, K. Ie. (2018), "Porivnjal'nyj analiz vlastyvostryj vidkrytogo, vil'nogo ta komercijnogo programnogo zabezpechennja" [Comparative properties analysis of open, free and commercial software], "Informacijni tehnologii' ta komp'juterna inzhenerija" [Information Technology And Computer Engineering], no. 1 (41). pp. 21–27.
11. Zhukovs'kyj, V. V. (2017), "Pro dejaki pidhody do stvorennja programnyh kompleksiv komp'juternogo modeljuvannja pidzemnyh procesiv" [About some approaches to creation of software complexes of computer simulation of underground processes], *Visnyk KrNU im. M. Ostrograds'kogo [Scientific journal "Transactions of Kremenchuk Mykhailo Ostrohradskyi National University"]*, no. 2, part 1, pp. 64–73.

Стаття надійшла 04.03.2019.