

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування

Кафедра Автоматизованих систем
моніторингу навколишнього середовища
та інформатики

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Створення тривимірної графічної моделі рельєфу
за допомогою BRL-CAD

Виконав: студент 2 курсу групи МКМ-20
спеціальності 122 Комп'ютерні науки

Нямцу Костянтин Євгенович

Керівник: професор каф. АСМНСІ, д. т. н.,
доцент Великодний Станіслав Сергійович

Консультант _____

Рецензент: д. ф.-м. н., доцент _____
Буяджи Василь Володимирович _____

Одеса 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ
Факультет Комп'ютерних наук, управління та адміністрування
Кафедра Інформаційних технологій
Рівень вищої освіти магістр
спеціальність 122 Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри _____
"28" жовтня 2021 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Нямцу Костянтину Євгеновичу(прізвище, ім'я, по батькові)

1. Тема роботи Створення тривимірної графічної моделі рельєфу за допомогою BRL-CAD
керівник роботи Великодний Станіслав Сергійович, д. т. н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «18» жовтня 2021 р. № 216 «С»
2. Строк подання студентом роботи 09 грудня 2021 р.
3. Вихідні дані до роботи: CAD/CAM/CAE-системи; методи і принципи автоматизованого проектування; методологія трасування променів; методологія накладення тіней; вихідний програмний код на ООП мові Java; CASE-інструментарій Enterprise Architect 12.0; вільне кросплатформове програмне середовище Ballistic Research Laboratory CAD (BRL-CAD) (v. 7.32.2)
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
вступ; аналітичний розділ; уніфікований діаграмний комплекс аналізу даних для створення тривимірної графічної моделі регіону, що містить діаграми: послідовності; класів, та варіантів використання; програмна реалізація та експериментальні дослідження імітаційної моделі тривимірної, графічної моделі регіону із описом: вихідних даних моделі, процесу організації структури моделі, процесу моделювання, процес написання програмних компонентів на мові програмування Java, числові висновки та аналіз результатів; висновки
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
графічний матеріал виконаний у вигляді презентації
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання “28” жовтня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Отримання завдання на роботу	28.10.2021		
2	Аналіз вихідних даних	30.10.2021		
3	Виконання аналітичного розділу	04.11.2021		
4	Складання вимог до програмного забезпечення	11.11.2021		
5	Формування структури тривимірної графічної моделі регіону	17.11.2021		
6	Процес завдання орієнтації та координат для тривимірної графічної моделі	22.11.2021		
7	Рубіжна атестація	22–26.11.2021		
8	Процес накладення трасування променів для цільової моделі	01.12.2021		
9	Процес виконання всіх розрахунків для моделі та накладення тіней	04.12.2021		
10	Процес виконання валідації відносно цільової моделі та створення документації	05.12.2021		
11	Формування висновків до роботи	06.12.2021		
12	Подання пояснювальної записки на кафедру	09.12.2021		
13	Перевірка пояснювальної записки на плагіат	10.12.2021		
14	Складання графічної частини	12.12.2021		
15	Підготовка доповіді на захист	15.12.2021		
16	Проходження рецензування	16.12.2021		
	Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)			

Студент _____

(підпис)

Керівник роботи _____

(підпис)

Нямцу К.Є

(прізвище та ініціали)

Великодний С. С.

(прізвище та ініціали)

АНОТАЦІЯ

«Створення тривимірної графічної моделі рельєфу за допомогою BRL-CAD». Нямцу Костянтин.

Мета роботи – створити тривимірну графічну модель рельєфу регіону засобами відкритої кросплатформеної САПР BRL-CAD. Об'єкт дослідження – система автоматизованого проектування BRL-CAD та її можливості. Предмет дослідження – моделі та методи проектування складних геометричних контурів.

Методи дослідження: методи проектування, методи моделювання, методи роботи з САПР BRL-CAD, методи створення діаграм, аналіз, збір даних, створення структури моделі, побудова цільової моделі, створення документації, проведення перевірок.

Результати, отримані автором, полягають у тому що було проведено повний аналіз САПР BRL-CAD, та здобуто навички по роботі з усіма модулями програмного забезпечення для подальшого моделювання регіону, а також побудовано UML діаграми, які описують архітектуру дизайну розробки графічної моделі регіону. Також створено тривимірну графічну модель рельєфу за допомогою САПР BRL-CAD, для облегшення подальших робіт по удосконаленню способів і засобів одержання даних про атмосферу радіофізичними методами.

Результатом розробки – став програмний продукт та інструкції із застосування необхідного програмного забезпечення, що доповнені коментарями, відносно роботи спроектованого рельєфу регіону засобами відкритої кросплатформеної САПР BRL-CAD.

Магістерська кваліфікаційна робота містить: 97 с.; 37 рис., 1 додаток, 27 використаних літературних джерел.

КЛЮЧОВІ СЛОВА: система автоматизованого проектування, каркасне моделювання, програмний засіб, графічний інтерфейс користувача, діаграма, модуль, графічна модель, тривимірний простір, система моделювання.

SUMMARY

"Creating a three-dimensional graphic relief model using BRL-CAD".
Niamtsu Kostiantyn.

The purpose of the work is to create a three-dimensional graphic model of the relief of the region by means of open cross-platform CAD BRL-CAD.

The object of research is the BRL-CAD computer-aided design system and its capabilities.

Subject of research – models and methods of designing complex geometric contours.

Research methods: design methods, modeling methods, methods of working with CAD BRL-CAD, methods of creating diagrams, analysis, data collection, creating a model structure, building a target model, creating documentation, conducting inspections.

The results obtained by the author are that a complete analysis of CAD BRL-CAD was performed, and skills were acquired to work with all software modules for further modeling of the region, as well as UML diagrams describing the design architecture of the graphic model of the region. A three-dimensional graphic model of the relief was also created with the help of BRL-CAD CAD, to facilitate further work on improving the methods and means of obtaining data on the atmosphere by radiophysical methods.

The result of the development was a software product and instructions for the use of the necessary software, supplemented by comments on the work of the projected relief of the region by means of open cross-platform CAD BRL-CAD.

Master's thesis contains: 97 pages; 37 fig., 1 appendix, 27 used literature sources.

KEY WORDS: computer-aided design system, frame modeling, software, graphical user interface, diagram, module, graphic model, three-dimensional space, modeling system.

ЗМІСТ

Перелік умовних позначок	8
Вступ	9
1 Вступ до основи моделювання BRL-CAD	13
1.1 Використання графічного інтерфейсу для відкриття або створення бази даних	19
1.2 Створення сфери з командного рядка	21
1.3 Використання insert команд	27
1.4 Конвенції про іменування BRL-CAD	30
1.5 Булеві вирази у BRL-CAD	31
1.6 CSG та BREP	41
2 Формати перетворень які підтримує BRL-CAD	44
2.1 Перетворення з формату AutoCAD DXF	44
2.2 Перетворення з формату STL	46
2.3 Перетворення з формату Pro / E	47
2.4 Перетворення з / на ASCII	52
3 Види графіки та графічної моделі	55
3.1 Векторні моделі	56
3.2 Растрові моделі	57
3.3 Тривимірні моделі	58
4 Принципи організації рендеренгу у САПР BRL-CAD	62
4.1 Розробка геометрії	66
4.2 Оцінювання геометрії	70
4.3 Розміщення фігур у тривимірному просторі в BRL-CAD	71
5 Аналіз та побудова тривимірної графічної моделі рельєфу Карпатського регіону	78
5.1 UML-діаграма послідовності побудови графічної моделі	78

5.2 UML-діаграма класів моделі.....	81
5.3 Основні команди і формули використані при моделюванні тривимірної графічної моделі карпатського регіону.....	83
Висновки.....	92
Перелік джерел посилання	95
Додаток А. Скрипт з компоненти Mged призначений для моделювання тривимірної моделі графічної області.....	98

ПЕРЕЛІК УМОВНИХ ПОЗНАК

ДВВ – діаграма варіантів використання.

ДК – діаграма класів.

ДП – діаграма послідовності.

ПЗ – програмний засіб.

ОС – операційна система.

САПР – система автоматичного проектування.

СУБД – система управління базами даних.

БД – база даних.

UML – unified modeling language.

MGED – multi-device geometry editor

BRL – ballistic research laboratory

CSG – constructive solid geometry

STL – standard template library

GUI – graphical user interface

RCC – right circular cylinder

BREP – boundary representation

CAD – computer-aided architectural design

DXF – drawing exchange format

NMG – objects of n-diverse geometry

SLA – service level agreement

PTC – parametric technology corporation

SGI – silicon graphics

PRT – crypt edit protected text format file

ВСТУП

На сьогоднішній день кожному користувачу, надається унікальна можливість завантажити на власний персональний комп'ютер, програмне забезпечення майже з будь-яким функціоналом, який задовольнив би його потреби у роботі, розробці, чи аналізі предметної області. Все це стає можливим, завдяки ситуації, яка складається на ринку ІТтехнологій. Суть її полягає в тому, що з кожним роком, кількість випущеного товару для ІТринку збільшується приблизно в два рази. Такий потік нового, покращеного програмного забезпечення, викликає у користувачів питання, який саме інструмент обрати, для того, щоб з його допомогою краще виконувати маніпуляції з даними у своїй роботі.

Існують декілька типів користувачів, деякі з них використовують платне ПЗ, інші ж використовують у своїх розробках та досліджах виключно ПЗ з відкритим вихідним кодом [1]¹⁾. Слід зазначити, що у наш час користувачів, які не платять за програмні засоби, дуже багато, а компанії які раніше випускали виключно платні продукти, поступово переходять до розробки, або до повністю безплатного забезпечення, або до такого, в якому є декілька платних функцій преміального типу, однак без них користувач, так само може виконувати потрібну йому роботу.

Платні програмні інструменти найчастіше, дуже добре рекламуються, саме тому вони мають дуже великий обсяг серед цільової аудиторії. На відміну від платного ПЗ, інструменти з відкритим вихідним кодом зазвичай, рекламуються без допомоги платної реклами, а завдяки тому, що деякі користувачі, які оцінили інструмент та попрацювали з ним, радять його на

¹⁾[1] Великодний С. С., Тимофєєва 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.

форумах іншим користувачам. Але є такі випадки, коли через те, що існує дуже багато альтернатив ПЗ у деякій сфері, користувачі випускають з уваги дуже цікаві та дійсно потужні засоби. Прикладом може послугувати така середа розробки та аналізу, як BRL-CAD.

З моменту свого існування BRL-CAD вже не один раз встиг зарекомендувати себе, як дуже ефективний та продуктивний інструмент для створення високої роздільної здатності, а також фізично реалістичної геометрії, яка потрібна, щоб виконувати балістичне проникнення радіолокаційного сигналу, та інших типів відповідних аналізів. Однак кілька комерційних пакетів САПР також набули популярності, в різних, великих організаціях.

Існує безліч переваг, пов'язаних із використанням комерційних пакетів у дослідженнях вразливості. Центр аналізу інформації про вразливість, визначив деякі найпоширеніші переваги у своєму сучасному звіті щодо геометричного моделювання.

Розглянемо їх більш детально:

- скорочення часу та зусиль на моделювання (виробники часто витрачають сотні годин на побудову детальних моделей САПР для впорядкування процесів проектування, виробництва та складання, наприклад, за допомогою комп'ютерного цифрового обладнання управління);
- збільшення фінансування та підтримки (оскільки комерційні пакети повинні підтримувати більшу базу користувачів, щоб залишатися конкурентоспроможними на відкритому ринку. Найпопулярніші пакети зазвичай мають достатнє фінансування та персонал для постійного їх розвитку, підтримки та вдосконалення);
- сумісність зі стандартизованими форматами (більшість комерційних пакетів мають пряму або непряму, через сторонніх постачальників можливість, перетворення на стандартні або посередницькі формати

геометрії, роблячи пакети сумісними з широким спектром інших форматів САПР);

- підтримка сторонніх надбудов (великі комерційні пакети, як правило, пропонують різноманітні плагіни для інших пакетів / утиліт).

Звичайно, комерційні пакети САПР також мають деякі загальні зобов'язання при використанні дослідження вразливості. Вони включають наступне:

- занадто багато деталей (комерційні файли геометрії часто містять занадто багато корисного для аналізу вразливості, тобто комерційні пакети САПР часто моделюють геометрію аж до рівня «гайок і болтів»);
- несумісні / недоступні формати файлів (деякі формати САПР використовують підхід до граничного представлення (BREP) для моделювання твердих тіл, який в основному несумісний із конструктивним підходом твердої геометрії, який використовує BRL-CAD);
- крім того, хоча більшість комерційних пакетів мають певні можливості для перетворення обміну даними, зберігання даних часто є у власному (і, отже, недоступному) рідному форматі (більше того, коли цільові описи перетворюються у формат, призначений для оцінки вразливості (тобто BRL-CAD або FASTGEN), вони часто вимагають ручної перевірки, коригування та додаткове моделювання);
- відносно повільна можливість випромінювання променів (комерційні пакети САПР, як правило, мають відносно низьку швидкість випромінювання променів, і випромінювання променів є основним способом опитування геометрії в оцінці вразливості);
- окрім типово високої вартості комерційного програмного забезпечення САПР (користувач часто стикається з рішенням,

інвестувати чи ні в апаратне забезпечення, що не є ПК (наприклад, UNIX робочі станції) для отримання максимальної продуктивності, особливо при великій, складній геометрії).

Мета роботи і завдання дослідження. Із розгляду поставленої проблеми, можна сформулювати мету магістерської роботи, яка полягає у створенні тривимірної графічної моделі рельєфу карпатського регіону засобами відкритої кросплатформної САПР BRL-CAD.

Об'єктом дослідження є система автоматизованого проектування BRL-CAD та її можливості.

Виконання зазначеної мети роботи може бути досягнуто після вирішення встановлених завдань:

- провести повний аналіз САПР BRL-CAD, навчитися користуватись усіма модулями програмного забезпечення для подальшого моделювання регіону;
- побудувати діаграми UML, які описуватимуть архітектуру дизайну розробки графічної моделі регіону;
- створити тривимірну графічну модель рельєфу за допомогою САПР BRL-CAD (для полегшення подальших робіт з удосконалення способів та засобів отримання даних про атмосферу радіофізичними методами).

Практичне значення отриманих результатів полягає в тому, що було виявлено нову, якісну, потужну САПР систему, за допомогою якої можливо моделювати великі моделі, а також масштабувати їх, на прикладі створення моделі рельєфу регіону.

Також удосконалено процес розробки цільової моделі за допомогою створених інструкцій з міграцій між САПР системами у BRL-CAD. Оскільки більшість комерційних систем САПР можуть створювати мозаїчні наближення своїх об'єктів BREP, деякі перетворювачі BRL-CAD використовують ці можливості. Крім того, згаданий раніше формат STL, що представляє тверді об'єкти повністю з трикутниками, пропонує спільноті

універсальний спосіб перетворення геометрії BRL-CAD майже будь-яку комерційну систему САПР.

Розроблено інструкції щодо застосування необхідного програмного забезпечення, доповнені коментарями щодо роботи спроектованого рельєфу регіону засобами відкритої кросплатформової САПР BRL-CAD.

1 ВСТУП В ОСНОВИ МОДЕЛЮВАННЯ У BRL-CAD

З кінця 1950-х років комп'ютери використовувались для допомоги у проектуванні та вивченні систем бойових машин. Результатом стало скорочення часу та грошей, необхідних для переходу системи від графічної дошки до повномасштабного виробництва, а також підвищення ефективності випробувань та оцінки.

З 1979 року дослідницька лабораторія армії США, займається розробкою BRL-CAD для роботи з конструктивною, твердо-тільною геометрією. Однак справжня сила цього програмного засобу полягає у тому, що він може працювати з об'єктами будь якого ступеню складності. Він складає їх з невеликої кількості примітивних форм. Це становиться можливим завдяки використанню основних логічних операцій об'єднання, віднімання та пересічення, а також за допомоги привласнення реальних, матеріальних атрибутів. Також слід сказати про те, що пакет BRL-CAD включає в себе велику кількість корисних інструментів та функцій, які дозволяють користувачеві будувати середовище для польових випробувань, яке має високий рівень якості.

Зараз, пакет що налічує майже мільйон рядків коду С, BRL-CAD, перетворився на потужний конструктивний пакет моделювання твердої геометрії [2]¹⁾, який отримав ліцензію на понад двох тисячах сайтів по всьому світу. Він містить велику колекцію інструментів, утиліт та бібліотек,

¹⁾ [2] Великодный С. С. Методологические основы реинжиниринга систем автоматизированного проектирования. *Междун. науч. журн. «Управляющие системы и машины»*. 2014. № 2. С. 39–43.

включаючи інтерактивний редактор геометрії, трасування променів та загальні бібліотеки буферів кадрів, розподілену в мережі функцію обробки зображень та обробки сигналів, а також настоювану вбудовану мову сценаріїв. Крім того, BRL-CAD одночасно підтримує подвійні методи взаємодії, один з яких використовує командний рядок, а другий використовує графічний інтерфейс користувача (GUI). Для того щоб завантажити собі на комп'ютер пакет BRL-CAD, потрібно перейти на офіційний сайт за посиланням <https://brlcad.org>. Після переходу на сайт, в шапці сайту ми можемо бачити шість розділів. Перший розділ відповідає, як за завантаження самого програмного засобу, так і за підкачування окремих інструментів і бібліотек. Оскільки даний програмний засіб є багатоплатформовим, його може бути встановлено на будь-які існуючі, сучасні операційні системи. У цьому розділі зберігаються папки, в яких зібрані всі версії даної програми, починаючи з найперших і закінчуючи новітніми (рис.1).

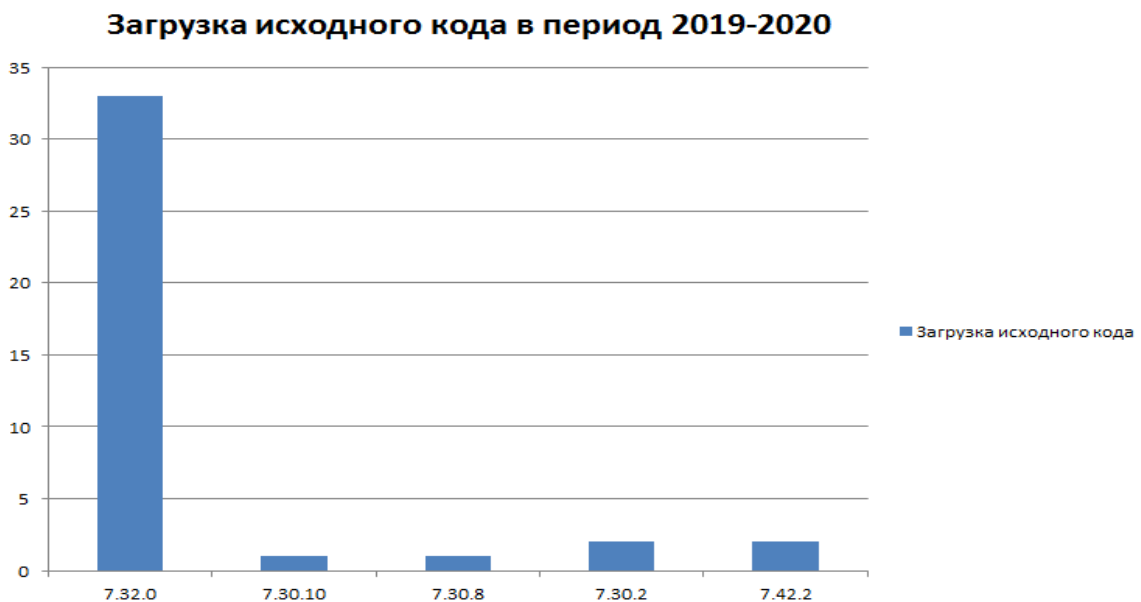


Рисунок 1 – Статистика по загрузці вихідного коду

Для кожної операційної системи передбачена своя папка з інструкцією і версіями самої програми. На сайті також встановлено лічильник, який

дозволяє вести статистику, та підраховувати кількість користувачів, які завантажили пакет, а також збирати інформацію о найбільш популярних ОС серед конкретної аудиторії користувачів. Саме такий вид збору інформації, надає розробникам змогу, виділити більше часу на розробку оновлень, саме для найпопулярніших серед аудиторії операційних систем, тим самим уникнувши різних недоліків, які можуть виникнути при установці та налаштуванні даного ПЗ.

Як демонструє статистика (рис. 2), найбільша кількість завантажень приходить саме на операційну систему Windows – 715 завантажень, друге місце займає Linux – 620 завантажень, а на третьому місці знаходиться MacOS з 83 завантаженнями.

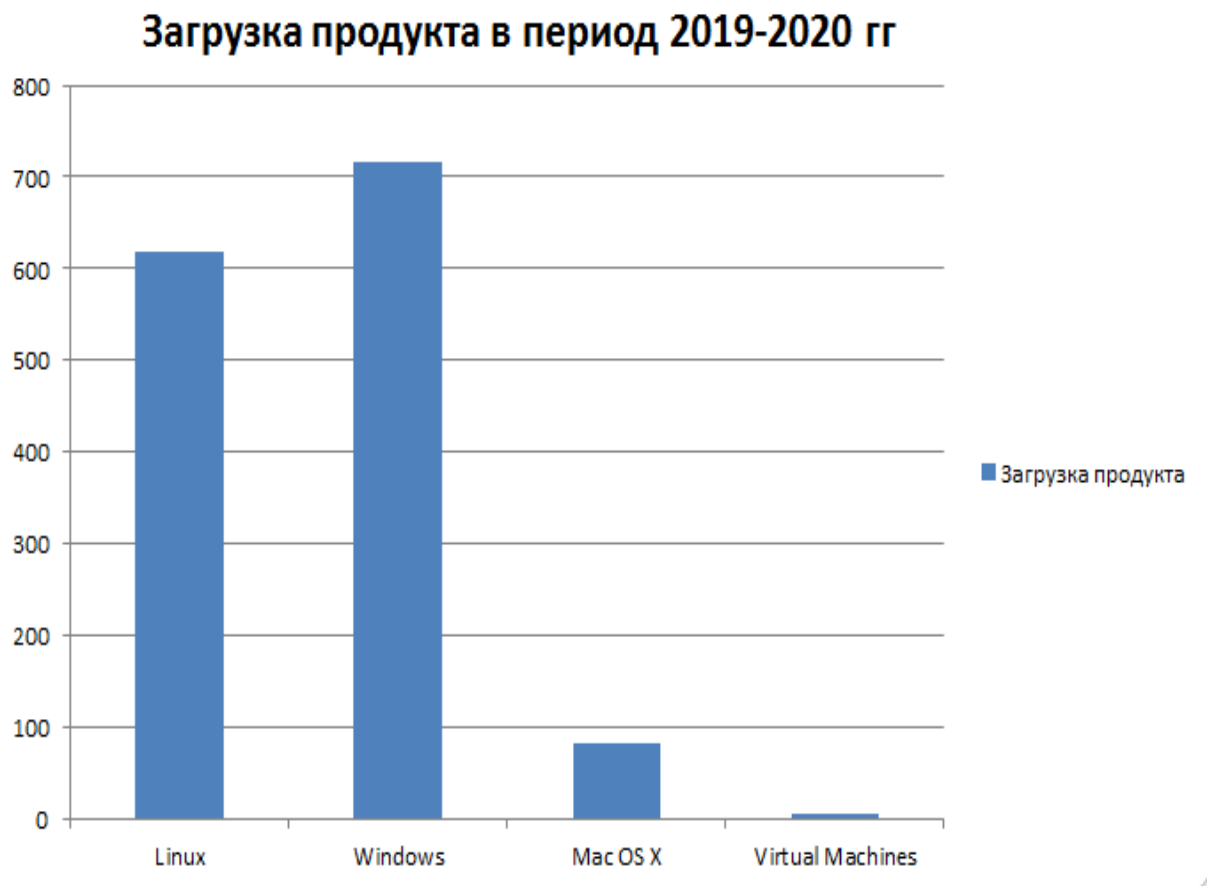


Рисунок 2 – Статистика по загрузці вихідного коду на певні ОС

Як ми можемо бачити BRL-CAD[3]¹⁾ не являється популярним програмним забезпеченням, невелика популярність BRL-CAD обумовлюється саме його минулим. До того, як стати програмою з відкритим вихідним кодом, цей різноманітних військових випробувань. Саме тому користувачі у всьому світі навіть не здогадувалися про існування такого потужного інструменту для роботи з конструктивною, твердо-тільною геометрією.

Другий розділ сайту надає інформацію про вихід нових оновлень, а також про виявлення багів, та їх виправлення. У цілому цей розділ може бути корисним користувачу. Завдяки йому він буде завжди у курсі речей, які пов'язанні з удосконаленням деяких інструментів BRL-CAD.

Наступний розділ надає повну інформацію о програмному забезпеченні, завдяки ньому користувач може познайомитись з історією розробки, та просування даного ПЗ.

Одним з найважливіших розділів на цьому сайті є розділ з документацією. Він містить багато різних посібників для роботи в BRL-CAD. Дане програмне забезпечення за своєю суттю складається із трьох підпрограм: MGED, Archerta RtWizard. Розробники рекомендують починати вивчення з інструменту MGED. Для його вивчення пропонується великий, детальний гайд. Після того як користувач оволодіє цим інструментом, рекомендується навчитись працювати, з двома останніми. Всю інформацію про них можна знайти в тому ж розділі. Далі користувач може переходити вже до стадії планування та моделювання проекту.

Останній розділ на сайті BRL-CAD, потрібен для того, щоб користувач міг надавати розробникам необхідний feedback у виді інформації про новітні баги чи неполадки, які пов'язані з самим програмним забезпеченням. Якщо у

¹⁾ [3] BRL-CAD Tutorial Series: Volume II – Introduction to MGED (загол. з екрану). URL: https://brlcad.org/w/images/c/cf/Introduction_to_MGED.pdf (дата звернення: 11.10.2020).

вас не працює BRL-CAD, технічна підтримка допоможе вам виправити вашу проблему. Єдиним недоліком підтримки BRL-CAD, є те що її працівники являються виключно англомовними, однак у наш час зовсім не обов'язково знати мову досконально, щоб пояснити технічній підтримці проблему, яка виникла у процесі використання програмного засобу.

Незважаючи на те, що BRL-CAD використовувався для широкого спектру інженерних та графічних застосувань, основним призначенням пакету продовжує залишатися підтримка балістичного та електромагнітного аналізів. Відповідно, розробники визнали, що моделювання CSG є найкращим підходом з точки зору точності моделі, ефективності зберігання, точності та швидкості обчислювального аналізу. Цей підхід на сьогодні, є одним з самих популярних у світі, та охоплює безліч різноманітних серед розробки, проектування, та аналізу.

Почнемо розглядати структуру BRL CAD з програмного забезпечення яке має назву MGED. Воно поставляється на комп'ютер користувача разом з пакетом BRL CAD. Для того, щоб запустити програму MGED, потрібно у командному рядку, вашої операційної системи, який розташовується прямо у вікні терміналу ввести команду `mgcd` після чого слід натиснути клавішу ENTER. Завдяки цієї маніпуляції буде викликано два головних вікна, які є дуже корисними у моделюванні складних систем, це: командне вікно MGED, а також графічне вікно MGED, яке являється основною родзинкою даного програмного продукту, іноді графічне вікно називають вікно для роботи з геометрією. Спочатку обидва вікна будуть порожніми, чекаючи наступних команд від користувача. Перебуваючи на будь якому етапі у роботі з даною програмою, user має можливість спокійно закрити вікно програмного засобу у будь який час. Для цього йому потрібно всього на всього ввести у рядок

команду quit та натиснути кнопку ENTER, після цього діяння усі вікна закриються[4]¹⁾.

Користувачі, які звикли до використання Microsoft Windows, або які просто часто працюють з цією операційною системою, можуть з легкістю віддати перевагу використанню меню, фішкою якого є те що воно, розкриває графічний інтерфейс користувача в верхній частині вікна команд або графіки, слід зазначити, що ці вікна є однаковими.

Ті ж користувачі, які більш звикли до використання операційної системи Linux чи MacOSX, найчастіше використовують саме вікно команд, це обумовлено тим, що ці користувачі дуже звикли до роботи з терміналом у Unix подібних системах, тож для них наявність командної строки у програмному забезпеченні, яке призначене для моделювання твердих геометричних систем, є цілою знахідкою.

Що стосується меню, слід сказати, що воно дуже зручно розділене на логічні групи, що допомагає користувачу, ще краще орієнтуватися в програмі MGED. Перш ніж ви зможете створити модель, вам необхідно відкрити нову базу даних або через вікно терміналу при запуску MGED або через графічний інтерфейс після запуску MGED (рис. 3).

¹⁾ [4] Velykodniy S. S. Analysis and synthesis of the results of complex experimental research on reengineering of open CAD systems. Applied Aspects of Information Technology. 2019. Vol. 2. No 3. P. 186–205. (кат. «Б») DOI: 10.15276/aait.03.2019.2.

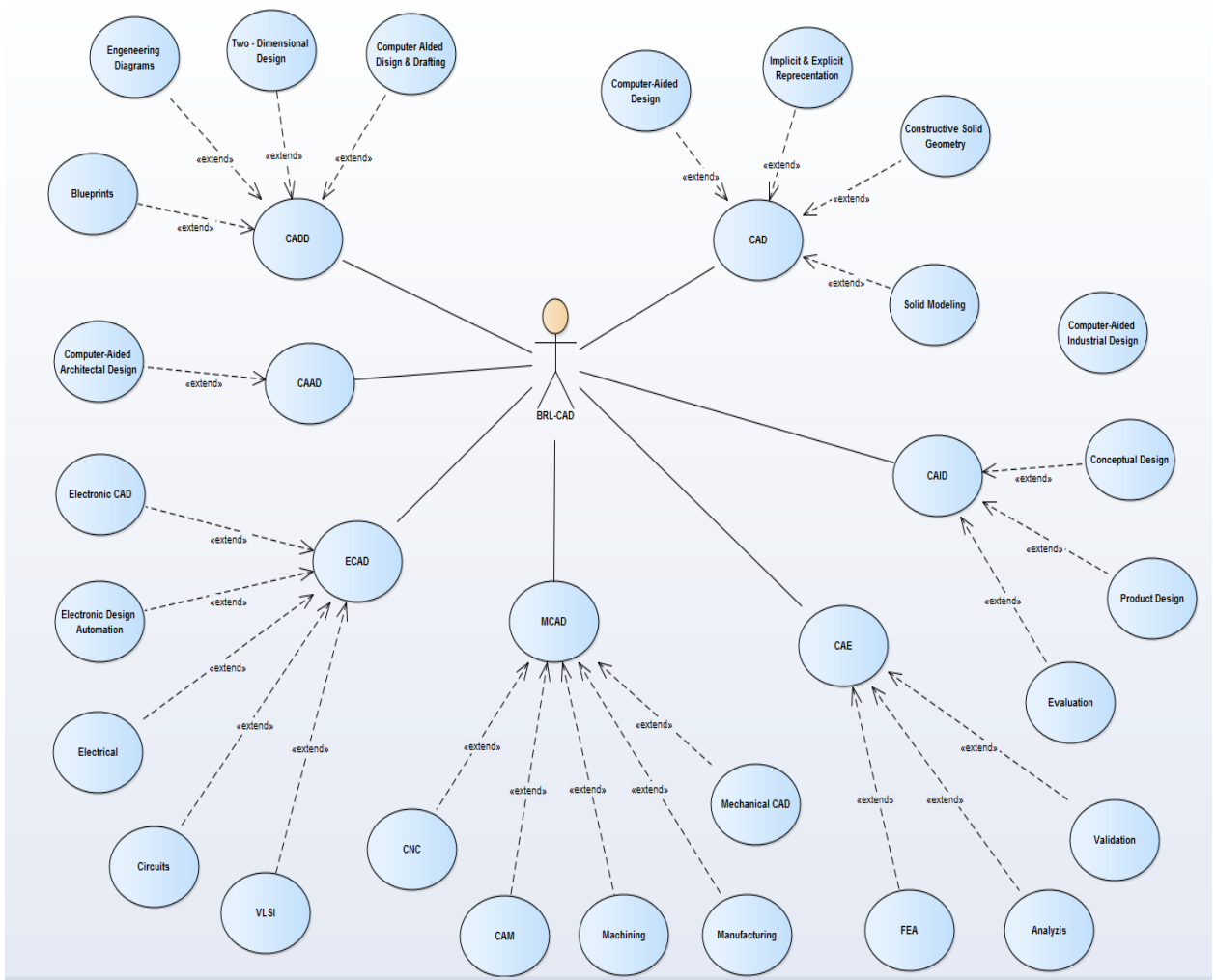


Рисунок 3 – Діаграма варіацій використання для опису можливостей системи

1.1 Використання графічного інтерфейсу для відкриття або створення бази даних

Після запуску MGED ви можете відкрити існуючу базу даних або створити нову базу даних [5] ¹⁾ за допомогою меню графічного інтерфейсу яке знаходиться у верхній частині вікна команд або графіки. Зробити це можливо за допомоги таких маніпуляцій(рис. 4), по-перше потрібно клацнути на

¹⁾ [5] Velykodniy S. Reengineering of open software system of 3-D modeling BRL-CAD. Innovative Technologies and Scientific Solutions for Industries.2019.No 3 (9), P. 62–71.(кат. «Б») DOI: https://doi.org/10.30837/2522-9818.2019_9.062.

кнопку File, яка знаходиться у вікнах на панелі зверху, а потім або Open, або New.

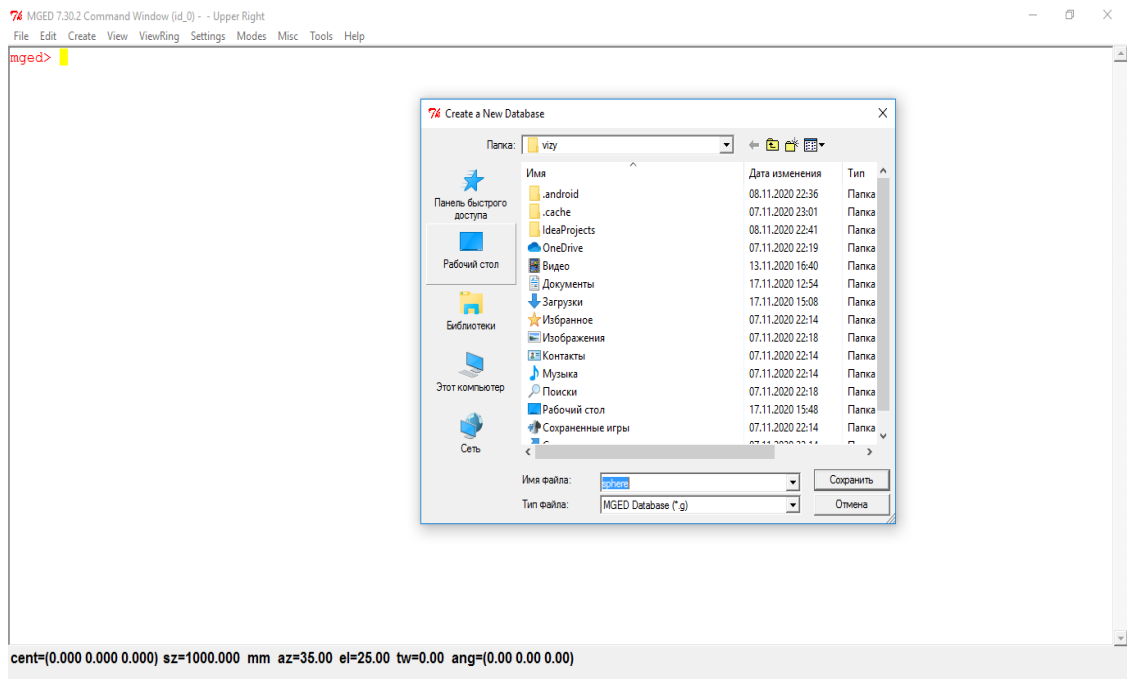
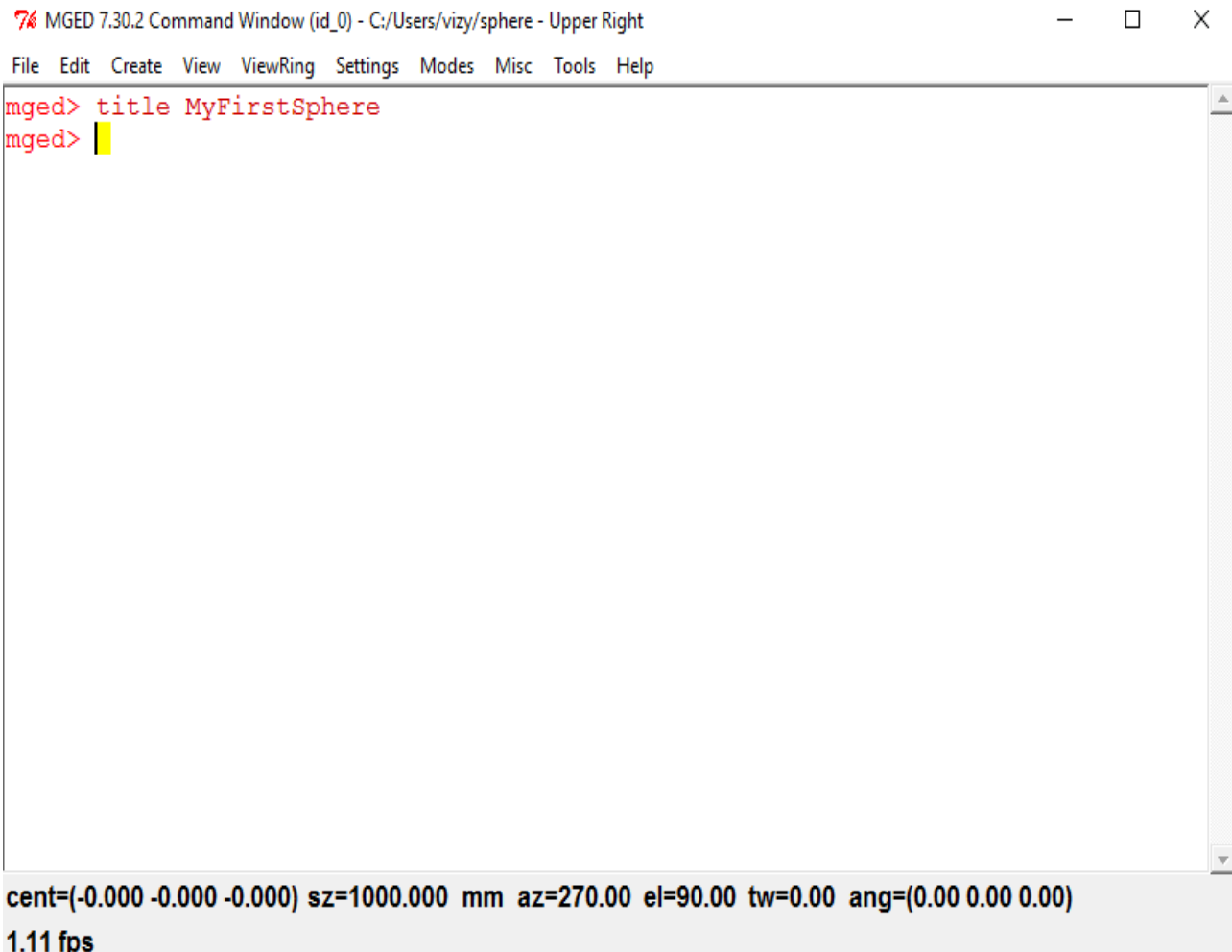


Рисунок 4 – Створення нової бази даних

Обидва варіанти викликають невелике діалогове вікно. У діалоговому вікні «Open» вас попросять ввести ім'я існуючого файлу, який містить цілу базу даних. У діалоговому вікні «New» вас попросять ввести ім'я нової бази даних. Після того, як користувач введе ім'я нової бази даних, йому потрібно натиснути на кнопку «ОК» у випадаючому вікні. Виконавши всі ці маніпуляції, у користувача, з'явиться БД з наприклад такою назвою – «sphere.g», де g є спеціальним розширенням, у якому конвертуються більшість файлів такого типу, які знаходяться у цієї системі для моделювання.

Ви можете назвати свою нову базу даних(рис.5), щоб забезпечити контрольний слід для вас або інших, хто може використовувати вашу базу даних. Після запрошення в командному вікні введіть заголовок, потім пробіл

і ім'я, яке відображає базу даних, яку ви збираєтеся створити. Коли ви зробили, натисніть клавішу ENTER.



```
MGED 7.30.2 Command Window (id_0) - C:/Users/vizy/sphere - Upper Right
File Edit Create View ViewRing Settings Modes Misc Tools Help
mged> title MyFirstSphere
mged> |
```

cent=(-0.000 -0.000 -0.000) sz=1000.000 mm az=270.00 el=90.00 tw=0.00 ang=(0.00 0.00 0.00)
1.11 fps

Рисунок 5 – Перейменування бази даних в терміналі

Наприклад:

```
mged> title MyFirstSphere <ENTER>
```

Зверніть увагу, що в версіях BRL-CAD до версії 6.0 заголовок обмежений всього 72 символами.

1.2 Створення сфери з командного рядка

Далі створюємо саму сферу. Є два способи для того щоб створити будь яку примітивну форму чи фігуру(рис. 6).

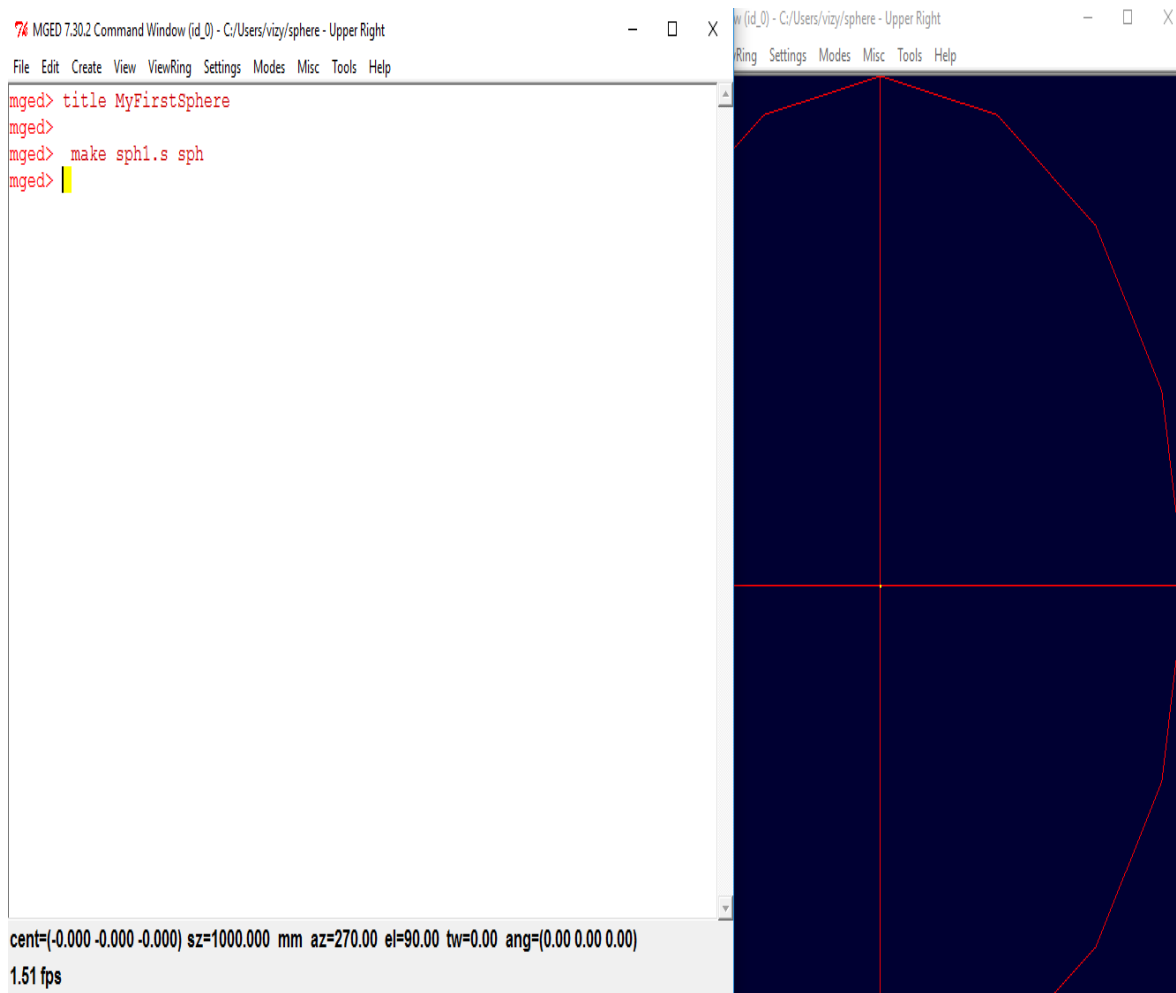


Рисунок 6 – Створення сфери через командний рядок

Слід зазначити, що користувач можете з легкістю створювати всі види форм, прямо за допомогою командного вікна. Теж саме можна робити через графічний інтерфейс, однак слід сказати, що в ньому є певні обмеження, що до створюваних форм.

Після чого сфера буде створена і користувач зможе почати працювати з нею(рис. 7).

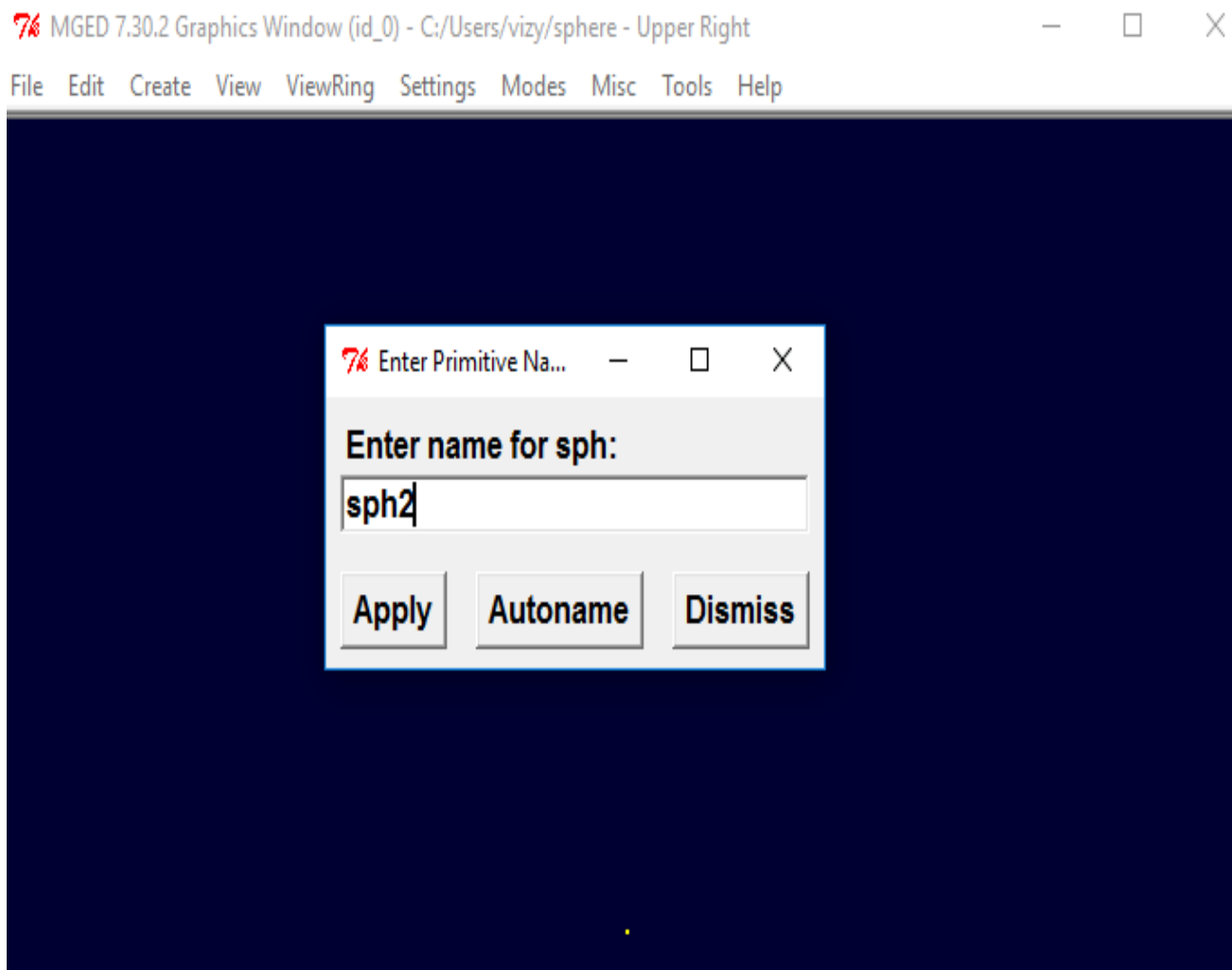


Рисунок 7 – Створення сфери у графічному вікні MGED

Сама сфера створюється дуже легко, для цього користувач повинен всього на всього у командному рядку програмного забезпечення, ввести дану строку:

```
make sph1.s sph <ENTER>
```

Також, як було описано вище, є інший спосіб створення сфери [6]¹⁾. Він полягає у використанні системи меню графічного інтерфейсу, дубльованої у верхній частині вікон Command and Graphics.

Спочатку треба очистити вікно графіки за допомогою описаної раніше команди quit. Потім у вікні «Graphic» натиснути «Create» і з'явиться випадаюче меню з різними доступним і примітивними типами фігур.

Обираємо sph (для сфери) у категорії Еліпсоїди. Після цього відкриється діалогове вікно. Потрібно клацнути у порожньому текстовому полі та ввести команду sph2.s.

Далі натиснути кнопку «Apply» або натиснути ENTER. У вікні графіки буде створено та намальовано нову сферу.

Коли ви створюєте фігуру через графічний інтерфейс, фігура автоматично перейде в режим редагування, щоб ви могли її змінювати за необхідності, а параметри фігури, які визначають її положення, орієнтацію та розмір, будуть переглянуті.

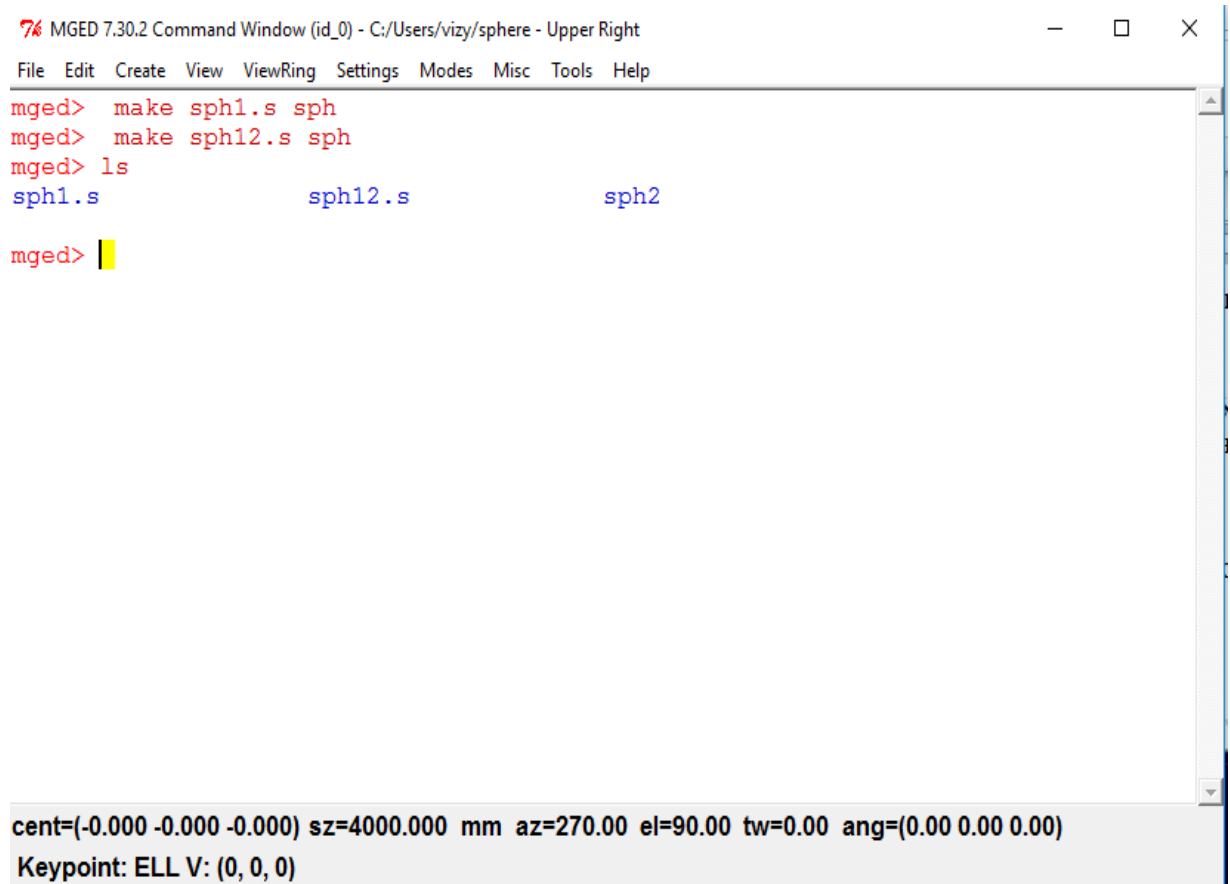
Слід зазначити, що коли користувач працює з деякою моделлю, яка описує певну поведінку у будь якій сфері, йому може знадобитись команда для того, щоб переглянути параметри фігури, яку він конструює, у вікні команд (висота, ширина, радіус).

Ви можете легко перерахувати атрибути фігури, набравши команду l (для «списку») у рядку командного вікна наступним чином: l shape_name, та натиснути клавішу <Enter>, для підтвердження.

Якщо ви намагаєтесь ввести команду в командному вікні і не бачите що символи уводяться, швидше за все, фокус на цьому вікні не встановлений (тобто введення з клавіатури все одно спрямоване в інше вікно). Залежно від

¹⁾[6]Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання структурної частини. Вісник Кременчуцького національного університету ім. Михайла Остроградського. 2019. Вип. 3 (116). С. 130–139. (кат. «Б») DOI:10.30929/1995-0519.2019.3.130-139.

конфігурацій вашої системи, фокус встановлюється на window weither, переміщуючи курсор у вікно або клацаючи на вікні (рис. 8).



```

7% MGED 7.30.2 Command Window (id_0) - C:/Users/vizy/sphere - Upper Right
File Edit Create View ViewRing Settings Modes Misc Tools Help
mged> make sph1.s sph
mged> make sph12.s sph
mged> ls
sph1.s          sph12.s          sph2
mged> |
cent=(-0.000 -0.000 -0.000) sz=4000.000 mm az=270.00 el=90.00 tw=0.00 ang=(0.00 0.00 0.00)
Keypoint: ELL V: (0, 0, 0)

```

Рисунок 8 –Список всіх створених об’єктів у базі даних

Окрім перегляду вмісту фігури, ви можете також захотіти перерахувати вміст бази даних, щоб побачити, які елементи були створені. Щоб переглянути вміст бази даних, введіть у вікні командного вікна: ls<ENTER>.

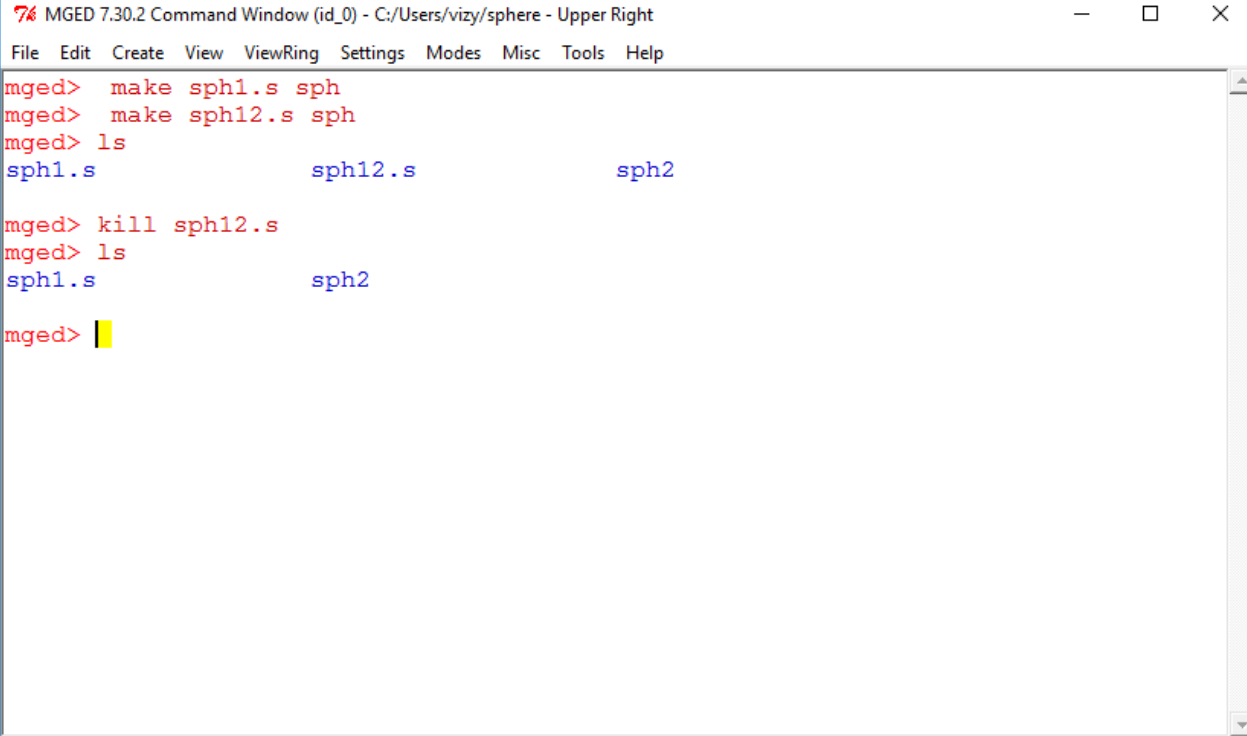
Іноді під час створення моделі вам може знадобитися виключити фігуру або об’єкт із бази даних. Для цього використовується команда kill.

Наприклад, якщо ви хочете вбити форму sph1.s, то введіть у вікні командного вікна:

```
kill sph12.s <ENTER>
```

Усі зміни негайно застосовуються до бази даних, тому немає команди «зберегти» або «зберегти як». Подібним чином, в даний час не існує команди «скасувати», щоб повернути те, що ви видалили, тому переконайтеся, що ви дійсно хочете остаточно видалити дані перед використанням команди kill.

Ви можете сказати, що фігура була усунена, скориставшись командою ls у вікні команд [7] ¹⁾, щоб перерахувати вміст бази даних. В командному вікні введіть: ls <ENTER> . Ви повинні побачити дві фігури у списку: sph1.s та sph2.s (рис.9).



```

MGED 7.30.2 Command Window (id_0) - C:/Users/vizy/sphere - Upper Right
File Edit Create View ViewRing Settings Modes Misc Tools Help
mged> make sph1.s sph
mged> make sph12.s sph
mged> ls
sph1.s          sph12.s          sph2
mged> kill sph12.s
mged> ls
sph1.s          sph2
mged>

```

cent=(-0.000 -0.000 -0.000) sz=4000.000 mm az=270.00 el=90.00 tw=0.00 ang=(0.00 0.00 0.00)
Keypoint: Ell V: (0 0 0)

Рисунок 9 –Видалення елемента бази даних командою kill

Іноді, коли ви вводите команди у вікні команд, ви робите помилку при введенні. MGED може емалювати синтаксис emacs або vi для редагування командного рядка. Ви також можете використовувати клавіші зі стрілками

¹⁾ [7] Грунина Е.В., Калачёв О.Н. Методика проектирования в CAD/CAM Cimatron УП для гравирования профиля на юбилейной медали. Ярославль: ЯГТУ, 2008. 58 с.

для редагування команд. Клавіші зі стрілками вліво та вправо переміщують курсор у поточному командному рядку.

Введення команди ENTER у будь-якому місці командного рядка виконує команду. Зверніть увагу, що клавіші BACKSPACE і DELETE видаляють один символ ліворуч від курсору. MGED зберігає історію введених команд. Стрілка вгору та вниз клавіші дозволяють вибрати раніше виконану команду для редагування та повторного виконання.

За замовчуванням використовується синтаксис emacs[8]¹⁾. Перегляньте список натискань клавіш, ефектів та способів вибору між двома наборами команд для Emacs в інтернеті.

Пам'ятайте, щоб будь-коли залишити програму, введіть з командного рядка букву q або слово quit, а потім натисніть клавішу ENTER. Ви також можете вийти з програми, вибравши Вихід у меню Файл.

1.3 Використання insertкоманд

Створимо нову базу даних та назвемо її «shape.g». Назвемо нашу базу даних – myShapes. Почнемо з активації вікна команд (зазвичай, клацнувши в будь-якому місці в вікно).

Потім у рядку MGED введемо команду:

```
make sph1.s sph<Enter>
```

Ця команда створює сферу, з тією назвою, яку введе користувач. Тепер коли фігура кулі створена, малюнок каркаса повинен з'явитися у вашому вікні графіки.

¹⁾ [8] Великодний С. С. Метод представлення оцінки реінжинірингу програмних систем за допомогою проектних коефіцієнтів. Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 1 (7). С. 34–42. (кат. «Б») DOI: <https://doi.org/10.30837/2522-9818.2019.7.034>.

Щоб зробити rcc(правий круговий циліндр)(рис. 10) з командного вікна, наспотрібно ввести:

```
make rcc1.s rcc<Enter>
```

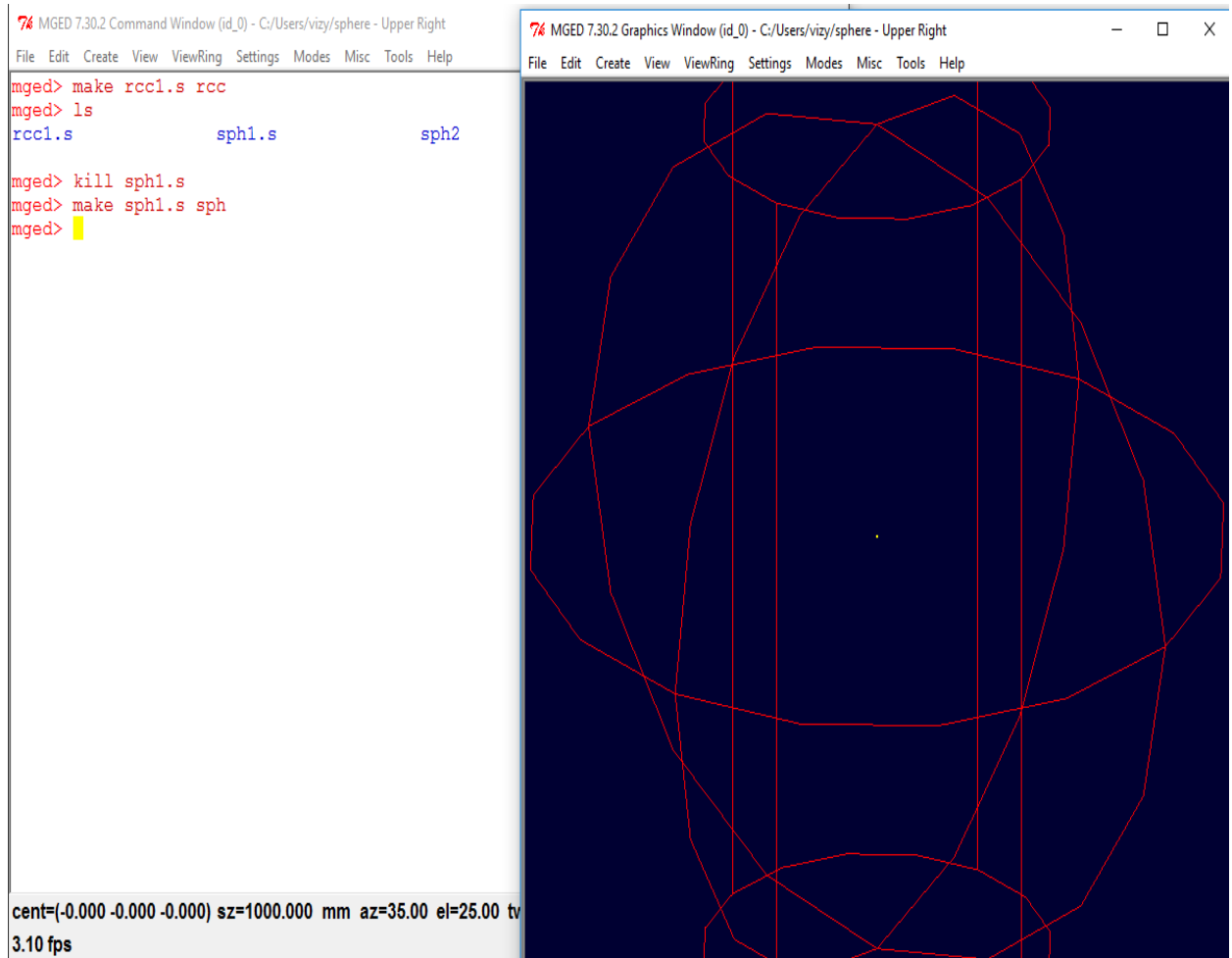


Рисунок 10 – Сфера та правий круговий циліндр

Тепер у вашому графічному вікні повинен відобразитися великий RCC, який з подання за замовчуванням az35, el25 виглядає так, ніби перетинає сферу, яку ви створили раніше [9] ¹⁾. Використання команди make

¹⁾ [9] Великодний С. С. Ідеалізовані моделі реінжинірингу програмних систем. *Радіоелектроніка, інформатика, управління*. 2019. № 1. С. 150–156. DOI: 10.15588/1607-3274-2019-1-14.

– це швидкий і простий спосіб створення фігури; однак для більшості моделей потрібні фігури з певними параметрами, такими як висота та радіус. Отже, більш точним способом створення цих фігур є використання команди `in` (`insert`).

За допомогою команди `Z` (`zap`) очистимо графічне вікно. Тепер ми готові створити сферу за допомогою команди `in`. За запитом MGED введіть:

```
in sph2.s sph<Enter>
```

MGED відповідь: – введіть X , Y , Z вершини. Ви повинні сказати MGED, де розташувати вершину (центр) вашої кулі у просторі. Введіть у підказці MGED:

```
4 4 4 <Enter>
```

Тепер ваша сфера буде розміщена на $(x, y, z) = (4,4,4)$, виміряна в міліметрах. Зверніть увагу, що цифри розділені пробілами, а після них в іде клавіша ENTER. Далі MGED буде просити вас ввести радіус, введемо значення:

```
3 <Enter>
```

Радіус вашої кулі буде 3 мм. Далі наведено діалогове вікно, яке має з'явитися у вашому вікні команд (включаючи відповідні відповіді).

Останній рядок цього діалогового вікна – це просто запис швидкості комп'ютера у нанесенні фігури. На даний момент це не має реальної користі для користувача. Тепер створено сферу, і у вашому графічному вікні має з'явитися каркасний малюнок, схожий на той, що створений за допомогою команди `make`.

Щоб зробити правильний круговий циліндр, введіть у вікні командного вікна:

```
in rcc2.s rcc <Enter>
```

Далі введемо для rcc всі параметри аналогічно до сфери. Тепер у вас повинні з'явитися нові версії форм сфери та rcc. Зверніть увагу, як ці дві фігури за розміром порівнюються з першими двома, які ви створили. Rcc тепер пропорційний кулі і розміщується в просторі ліворуч у вашому графічному вікні. Вказавши розміри фігур та їх розташування у просторі, ви змогли створити модель точніше (рис. 11).

```

MGED 7.30.2 Command Window (id_0) - C:/Users/vizy/sphere - Upper Right
File Edit Create View ViewRing Settings Modes Misc Tools Help
mged> make rcc1.s rcc
mged> ls
rcc1.s          sph1.s          sph2
mged> kill sph1.s
mged> make sph1.s sph
mged> in sph3.s sph
Enter X, Y, Z of vertex: 4 4 4
Enter radius: 3
sph3.s
mged>
cent=(-0.000 -0.000 -0.000) sz=1000.000 mm az=35.00 el=25.00 tw=0.00 ang=(0.00 0.00 0.00)
3.44 fps

```

Рисунок 11 – Створимо сферу за допомоги команди Insert

1.4 Конвенції про іменування BRL-CAD

Кожного разу, коли я створював кулю, або rcc, я давали їй іншу назву. MGED байдуже, яке ім'я ви надасте фігурі, але під час розробки моделей ви

виявите, що це допомагає мати певні формули або домовленості при іменуванні фігури. Зауважте також, що кожне ім'я повинно бути унікальним у базі даних, а для випусків BRL-CAD до 6.0 імена [10] ¹⁾обмежуються 16 символами.

Іноді фігурам було присвоєно імена залежно від типу їх фігури та порядку їх створення. Це було зроблено, оскільки фігури не мали реальної функції, крім як приклади.

Однак, коли користувач створює реальні моделі, він ймовірно захоче призначити імена, таким ж чином, як це було зроблено для назв радіокомпонентів, які базувались на їхніх функціях (наприклад, `btn` для кнопки, `ant` для антени тощо). Якщо ви працюєте з більш досвідченими моделями, зв'яжіться з ними, щоб побачити, який набір конвенцій вони використовують. Якщо ви працюєте наодинці, розробіть набір правил, що відповідають вашим іменам, а потім послідовно використовуйте його.

1.5 Булеві вирази BRL-CAD

Логічні операції мають вирішальне значення для процесу моделювання. Порядок поєднання фігур та оператори, що використовуються для комбінування фігур, визначатимуть, як програма MGED інтерпретує вашу модель. Правильне використання булевих виразів для модифікації геометричних фігур є ключовим навиком конструктивного моделювання твердих тіл.

Концептуально в MGED є два об'єкти, які підтримують логічні операції. Один називається комбінацією, інший – регіоном. Типову геометричну фігуру в MGED називають примітивною, однак одиничних

¹⁾ [10] Трухнин Н.М., Филиппович К.В. Макрорасширения в GrafCAM – путь к наращиванию функциональности. Пермь: ООО «Евразия Лимитед», 2006. 72 с.

примітивів часто недостатньо для повного опису складної форми об'єкта, що моделюється. Отже, поєднання двох або більше примітивних фігур в інші фігури (так звані комбінації) за допомогою булевих операторів дозволяє майстерно імітувати форму і форму найскладніших об'єктів. Властивості матеріалу пов'язані з регіонами. Як і комбінації, регіони використовують булеві операції для створення складних фігур.

Різниця полягає в тому, що регіони – це фігури, що мають властивості матеріалу. Вони займають тривимірний простір, а не просто визначають форму в просторі. Ви можете сприймати примітиви та комбінації як схему об'єкта. Фактичний об'єкт створюється при створенні регіону. Наприклад, ви можете скласти план такого предмета, як кружка для кави, але потім створити цю кружку з різних типів матеріалу (наприклад, керамічний або скляний). Незалежно від матеріалу, план є однаковим.

Коли булеві операції використовуються для побудови складних фігур з більш простих фігур, ми можемо назвати результат комбінацією фігур. Коли вони використовуються для визначення іншої логічної або ієрархічної структури в базі даних, результат може називатися групою або монтажне поєднання.

Логічні операції мають вирішальне значення для процесу моделювання. Порядок поєднання фігур та оператори, що використовуються для комбінування фігур, визначатимуть, як програма MGED інтерпретує вашу модель. Правильне використання булевих виразів для модифікації геометричних фігур є ключовим навиком конструктивного моделювання твердих тіл.

Концептуально в MGED є два об'єкти, які підтримують логічні операції. Один називається комбінацією, інший – регіоном. Типову геометричну фігуру в MGED називають примітивною, однак одиничних примітивів часто недостатньо для повного опису складної форми об'єкта, що моделюється. Отже, поєднання двох або більше примітивних фігур в інші фігури (так звані комбінації) за допомогою булевих операторів дозволяє

майстерно імітувати форму і форму найскладніших об'єктів. Властивості матеріалу пов'язані з регіонами. Як і комбінації, регіони використовують булеві операції для створення складних фігур. Різниця полягає в тому, що регіони – це фігури, що мають властивості матеріалу. Вони займають тривимірний простір, а не просто визначають форму в просторі. Ви можете сприймати примітиви та комбінації як схему об'єкта. Фактичний об'єкт створюється при створенні регіону. Наприклад, ви можете скласти план такого предмета, як кружка для кави, але потім створити цю кружку з різних типів матеріалу (наприклад, керамічний або скляний). Незалежно від матеріалу, план є однаковим.

Коли булеві операції використовуються для побудови складних фігур з більш простих фігур, ми можемо назвати результат комбінацією фігур. Коли вони використовуються для визначення іншої логічної або ієрархічної структури в базі даних, результат може називатися групою або монтажне поєднання. Три логічні оператори, використані програмою MGED, це об'єднання, віднімання та перетин. Ви можете використовувати булеві операції для комбінування фігур для отримання більш складних фігур:

- форми об'єднання: допомагають об'єднати дві фігури;
- форми віднімання фігур: видаляють об'єм однієї фігури з іншої;
- форми перетину: використовуються лише, якщо частини двох фігурякі перекриваються (оператор об'єднання -u, об'єднує фігури так, що будь-яка точка хоча б в одній з них буде частиною результату, union – потужний та часто використовуваний оператор).

Коли у примітивної форми віднімається друга, що накладається одна на іншу, в результаті друга форма зникає разом із будь-яким загальним об'ємом, який вона мала з першою фігурою.

Оператор Subtraction (знак мінус) означає віднімання або різницю. Ця операція особливо корисна при видовбуванні кузова, видаленні незвичної форми шматка примітивної форми або обліку перетину країв стін, плит, трубопроводів або інших з'єднаних форм. Пунктирна червона лінія, яка

з'являється після використання цього оператора, вказує на те, фігура, яку ми віднімаємо, протягається всередині іншої фігури праворуч. Ця частина, що накладається, частково не видна в зображення з променями.

Булева операція перетину, що позначається оператором `,` `+` (знак плюс), поєднує дві примітивні фігури, які перекривають одна одну, зберігаючи лише їх загальний обсяг (неперекритих областей не буде). Простий спосіб зрозуміти перехрестя – це думати про фігури як про дороги. Перехрестя – це місце перекриття двох доріг. Хоча багатьом людям операції перетину важче зрозуміти, ніж об'єднання та віднімання, незвичні / складні фігури можна виразити за допомогою оператора перетину. Наприклад, ви можете змодельювати збільшувальну лінзу, як перетин двох сфер. Операція перетину рідко корисна, якщо наприклад принаймні дві фігури не перекриваються. Перетин двох фігур, що не мають спільних точок (тобто, немає перекриття), є нульовим набором, тому воно взагалі не включає точок простору (рис. 12).

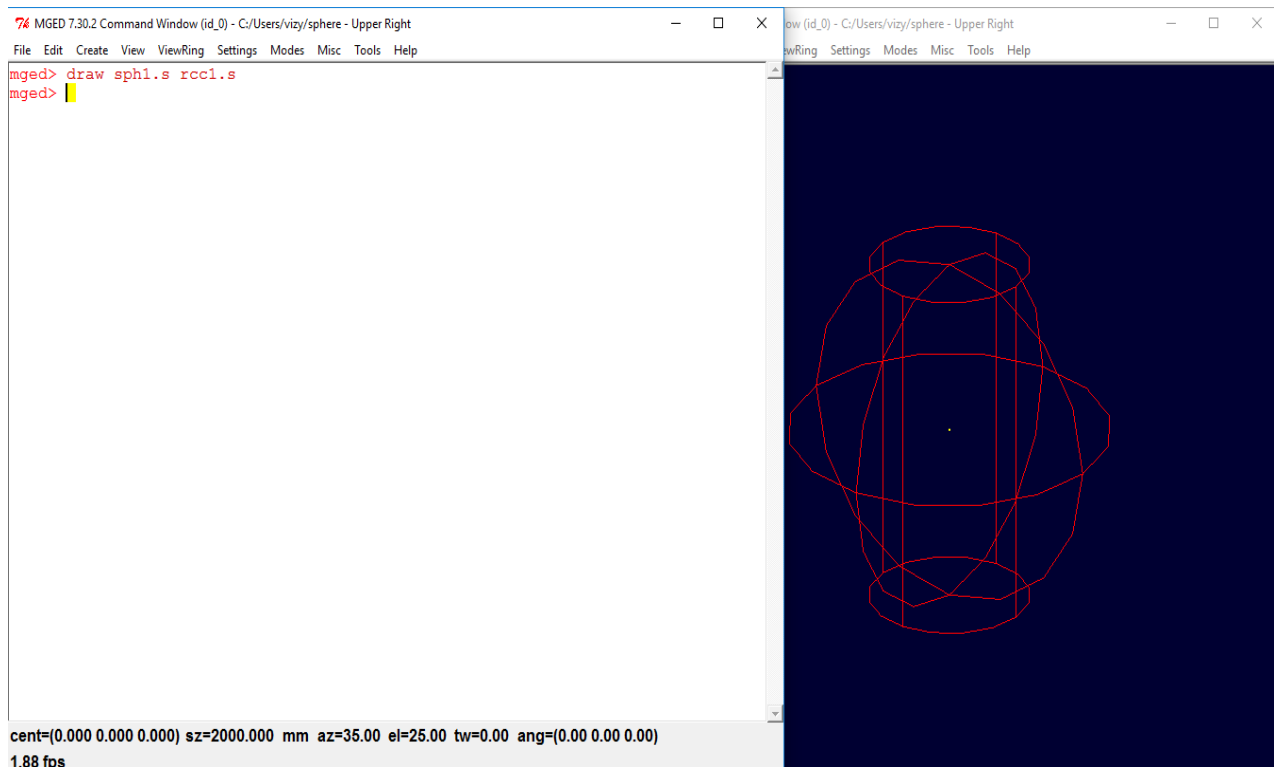


Рисунок 12– Прорисовка вибраних у терміналі елементів

Існує одне важливе обмеження при використанні булевих операторів віднімання та перетину. Повинна бути перша фігура, з якої можна відняти або перехрестити другу фігуру. Якщо у вас є лише одна фігура в межах регіону або комбінації, оператор буде ігноруватиметься, і оператор об'єднання буде використовуватися завжди.

Почнем з відкриття бази даних sphere.g, яку ми вже створили раніше. У вікні командного вікна введемо команду:

```
draw sph2.s rcc2.s <ENTER>
```

Це дозволяє побачити форми, які ми використовуватимемо для створення наших регіонів(рис. 13).

The screenshot shows a terminal window titled "MGED 7.30.2 Command Window (id_0) - C:/Users/vizy/sphere - Upper Right". The window contains the following text:

```

mgd> ls
rcc1.s          sph1.s          sph2          sph3.s

mgd> draw sph1.s rcc1.s
mgd> r part1.r u rcc1.s - sph1.s
Defaulting item number to 1001
Creating region with attrs: region_id=1000, los=100, material_id=1

mgd> |

```

At the bottom of the window, the following status information is displayed:

```

cent=(0.000 0.000 0.000) sz=2000.000 mm az=35.00 el=25.00 tw=0.00 ang=(0.00 0.00 0.00)
7.54 fps

```

Рисунок 13 – Створення нового регіону на основі двох фігур

Як видно, дві фігури повинні виглядати приблизно так. У вікні команд введіть наступне:

```
r part1.r u rcc2.s - sph2.s <ENTER>
```

Ця команда повідомляє MGED[11]¹⁾, що підкоманда `r` створює регіон, `part1.r` називає його першою частиною (`part1`), а `-` об'єднує дві форми, а `rcc2.s` позначає, як раз таки другу форму (рис. 14).

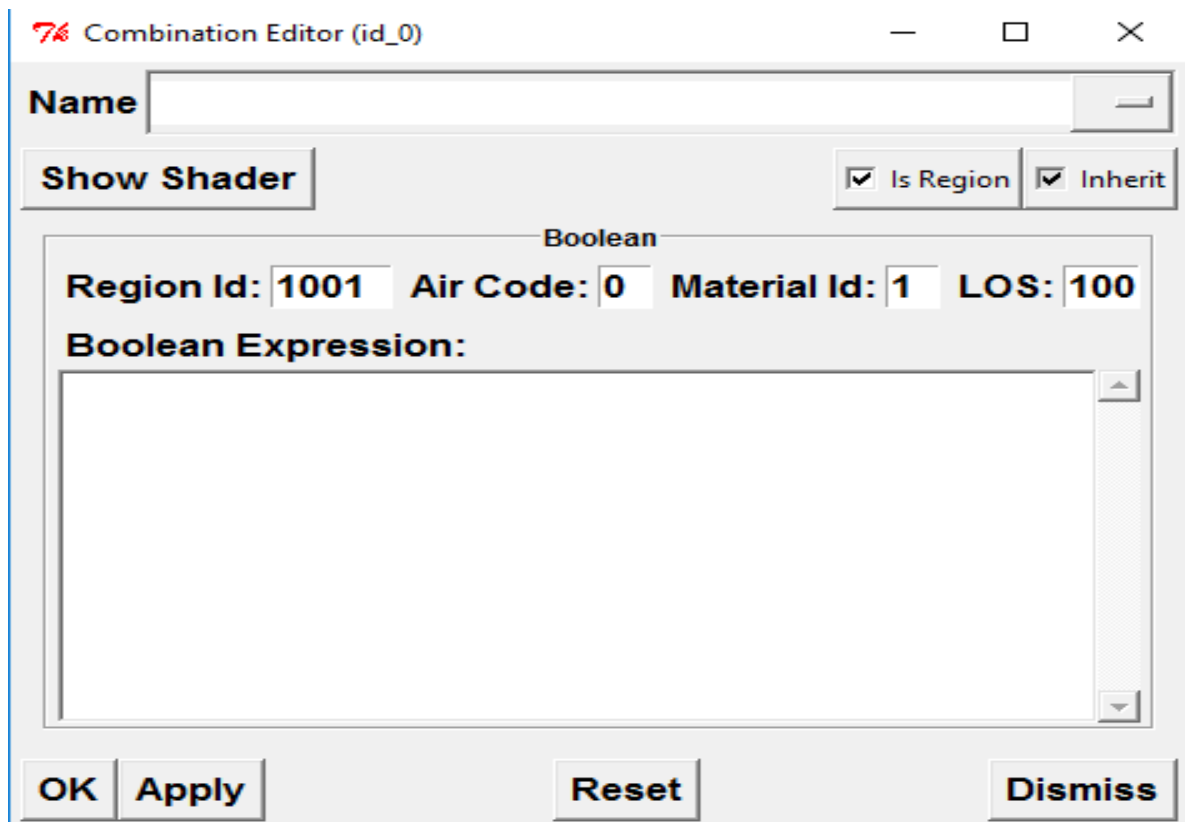


Рисунок 14 – Використання графічного інтерфейс для створення нового регіону

¹⁾ [11] Филиппович К.В., Попович И. ToolStore – среда для ведения библиотеки инструментов в верификаторе CNC-Verify системы САПР-ЧПУ/2005. Пермь: ООО «Евразия Лимитед», 2005. 36 с.

Ми застосували властивості матеріалу до об'єктів із командного рядка. Тепер ми будемо використовувати графічний інтерфейс, щоб зробити теж саме. У меню Редагування виберіть Комбінований редактор. З'явиться діалогове вікно. Натисніть кнопку праворуч від поля введення імені, а потім натисніть Вибрати з усіх. З'явиться спадне меню із регіонами, які ви створили. Виберіть part1.r.

Перший член завжди має малу літеру u для оператора. другий та наступні члени можуть використовувати -, + або u за потреби. Процес визначення операторів, і в якому порядку, обговорюється в більш досконалому навчальному посібнику.

Клацніть на кнопку поруч із кольором і виберіть у спадному меню червоний. Тепер натисніть кнопку ОК у нижньому лівому куті діалогового вікна. Це застосує ваші зміни та зачинить панель. На даний момент у нас відображаються лише примітивні форми, а не регіон. Перш ніж ми зможемо простежити промені, ми повинні видалити примітивні фігури з дисплея та намалювати область. В іншому випадку ми не зможемо побачити область із властивостями кольору, які ми застосували. Ми можемо зробити це, набравши:

В part1.r

У меню Файл відкрийте панель керування Raytrace і натисніть кнопку Raytrace.Зображення, яке ви отримаєте, має виглядати схоже на зображення зліва, яке подається далі. Зверніть увагу, що для простеження вікна може знадобитися кілька хвилин, залежно від швидкості вашої конкретної системи.

Ви побачите, що сферичний «укус» був вийнятий з верхньої частини циліндра. Далі ми зробимо синю область, використовуючи оператор перетину замість віднімання. Ще раз, ми починаємо зі створення регіону:

```
r part2.r u rcc2.s + sph2.s <ENTER>
```

Для порівняння з підходом графічного інтерфейсу, який використовується для створення part1.r, скористаємося командним рядком, щоб призначити колір part2.r:

```
mater part2.r plastic 0 0 255 0 <ENTER>
```

Нарешті, викиньте цей новий регіон на дисплей наступним чином:

```
v part2.r<ENTER>
```

Тепер простежте об'єкт променем. Він повинен виглядати подібним до попереднього зображення праворуч (рис. 15).

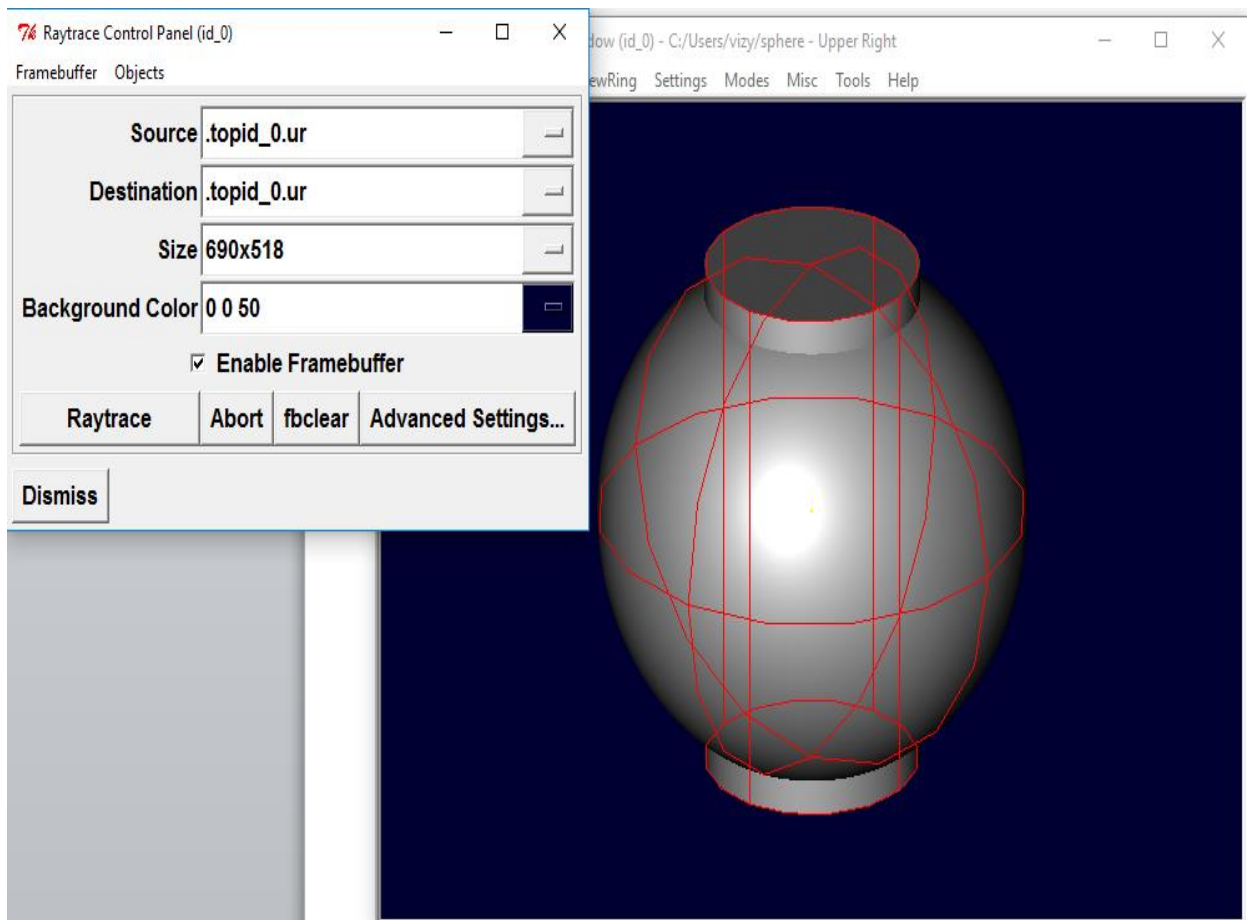


Рисунок 15 – Виконання функції Raytrace для нашої заготовки

Зображення, яке ви отримаєте, має виглядати схоже на зображення зліва, яке подається далі. Зверніть увагу, що для простеження вікна може знадобитися кілька хвилин, залежно від швидкості вашої конкретної системи.

Ви побачите, що сферичний «укус» був вийнятий з верхньої частини циліндра. Далі ми зробимо синю область, використовуючи оператор перетину замість віднімання. Ще раз, ми починаємо зі створення регіону:

```
r part2.r u rcc2.s + sph2.s <ENTER>
```

Для порівняння з підходом графічного інтерфейсу, який використовується для створення part1.r, скористаємося командним рядком, щоб призначити колір part2.r:

```
mater part2.r plastic 0 0 255 0 <ENTER>
```

Нарешті, викиньте цей новий регіон на дисплей наступним чином:

```
v part2.r<ENTER>
```

Тепер простежте об'єкт променем. Він повинен виглядати подібним до попереднього зображення праворуч.

Не забудьте очистити графічне вікно та намалювати нову область або комбінацію, перш ніж намагатися простежити модель. Raytracer ігнорує область або комбінацію, які не намальовані у графічному вікні. Колір каркасу –це ваша підказка. Якщо він не відображає кольори, які ви призначили (наприклад, все намальовано червоним, навіть якщо ви призначили інші кольори), ви не очистили екран примітивних фігур і не намалювали нову область або комбінацію, оскільки час, коли ви це зробили.

Коли ви використовуєте оператор перетину, порядок, у якому ви вказуєте фігури, не має значення. Ми отримали б однакові результати, якби вказали логічну операцію як:

```
r part2.r u sph2.s + rcc2.s
```

Однак при використанні оператора віднімання порядок двох фігур дуже важливий. Давайте зробимо область із порядком фігур, зворотнім до того, що використовувався для part1.r:

```
r part3.r u sph2.s - rcc2.s
```

Цього разу ми не потурбуємось встановити колір. (Коли для об'єктів не встановлено колір, raytracer (rt) використовуватиме білий колір. Однак ці об'єкти можуть виглядати сірими через кількість світла в сцені.)

Тепер давайте простежимо промені всіх трьох об'єктів, які ми створили разом. Щоб намалювати відразу три регіони, ми могли б набрати:

```
v part1.r part2.r part3.r z
```

Зробити це один раз – не проблема. Однак, якби це були три частини, які склали якийсь складний об'єкт, ми могли б хотіти мати можливість намалювати їх все зручніше. Щоб полегшити малювання колекції об'єктів разом, ми створюємо комбінацію збірок щоб зібрати їх усіх разом. Ми створимо купол dome.c для наших трьох регіонів.

Це досягається наступною командою:

```
comb dome.c u part1.r u part2.r u part3.r.
```


Зверніть увагу на схожість цієї команди з командою r , яку ми використовували для створення регіонів [12] ¹⁾. Різниця між регіоном і комбінацією полягає в тому, що міркування не обов'язково складаються лише з одного виду матеріалу. Кілька предметів з різних матеріалів можуть скласти комбінацію збірок, наприклад, щойно створену нами.

Форми, які ми створили тут, досить прості. У кожному випадку уніфікується одна примітивна фігура, і операції віднімання або перетину виконуються на цій єдиній примітивній фігурі [13] ²⁾. Ви повинні знати, що можна використовувати набагато складніші булеві рівняння для створення форми об'єктів. Коли ви хочете створювати такі об'єкти, пам'ятайте про перевагу логічних операцій. У булевих позначеннях, які ми використовуємо, оператори віднімання та перетину мають вищий пріоритет, ніж має оператор об'єднання.

1.6 CSG та BREP

Існує два основних підходи до твердого моделювання: CSG та BREP. При моделюванні CSG об'єкт представляється як булева комбінація простих примітивних форм (наприклад, сфери, циліндри, конуси тощо) (рис. 16).

¹⁾[12] Дискретная математика. Часть II. Математическая логика, : [Учеб.пособие.]: Э.Р. Зарипова, М.Г. Кокотчикова, Л.А. Севастьянов. – М.: РУДН, 2013. – 116 с.

²⁾ [13] Редькин Н.П. Дискретная математика. - М.: ФИЗМАТЛИТ, 2009. - 264 с. - ISBN 978-5-9221-1093-8.

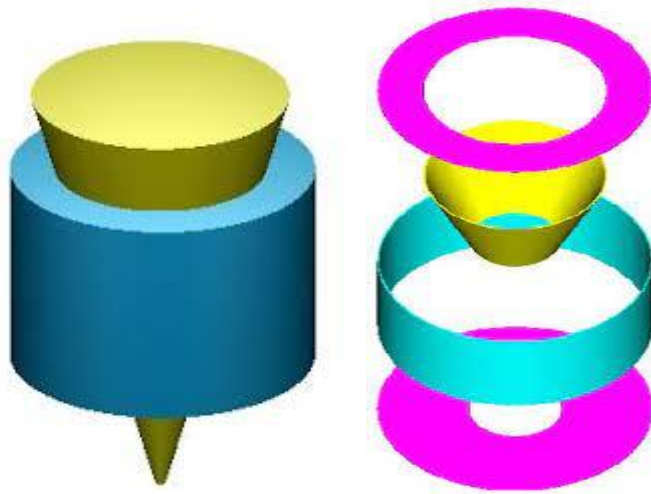


Рисунок 16 – Підходи до представлення екструзійної матриці.

У моделюванні BREP об'єкт представлений набором поверхонь (наприклад, граней, трикутників та сплайнів), які «зшиті» разом, щоб повністю закрити об'єкт. Більшість основних комерційних пакетів твердого моделювання, які зараз розповсюджуються, використовують підхід BREP, хоча вони мають певні можливості CSG. З іншого боку, BRL-CAD – це в першу чергу система моделювання CSG з деякими можливостями BREP.

Для BRL-CAD доступний простий тип BREP у формі трикутників; його називають примітивом «Мішок трикутників» [14] ¹⁾. Якщо об'єкт BREP складається з усіх трикутників, подання BRL-CAD може бути точним дублікатом. Але частіше об'єкт BREP складається з гладких криволінійних поверхонь, тобто представлення трикутника BRL-CAD буде наближенням об'єкта. Зверніть увагу, що це наближення може бути зроблено якомога ближче до справжньої поверхні, але за рахунок більших, менших трикутників. Таким чином, вищий ступінь плавності означає більший вплив на розмір файлу та продуктивність будь-якої програми, яка намагається використовувати всі трикутники. Оскільки більшість комерційних систем

¹⁾[14] Боггс У., Боггс М., UML. Rational Rose: Секреты эффективного проектирования сопровод. объектно-ориентиров. прилож. Москва: «Лори», 2000. 570 с.

САПР мають можливість створювати мозаїчні наближення своїх об'єктів BREP, деякі перетворювачі в BRL-CAD використовують ці можливості.

Крім того, згаданий раніше формат STL, який представляє тверді об'єкти повністю із трикутниками, пропонує спільноті універсальний (хоча і неочищений) спосіб перетворення геометрії BRL-CAD (за допомогою перетворювача g-stl) у майже будь-яку комерційну систему САПР.

2 ФОРМАТИ ПЕРЕТВОРЕНЬ ЯКІ ПІДТРИМУЄ BRL-CAD

Розглянемо таблицю (рис.17), яка містить основні формати файлів (крім ASCII), які в даний час можуть перетворюватися на BRL-CAD.

CAD Format	Import Converter
<u>AutoCad DXF</u>	<u>dx-f-g</u>
<u>Elysium Neutral Facceted</u>	<u>enf-g</u>
EUCLID	<u>euclid-g</u>
IGES	<u>iges-g</u>
STL	<u>stl-g</u>
JACK	jack-b
TANKILL	<u>tankill-g</u>
NASTRAN	<u>nastran-g</u>

Рисунок 17 – Конвертери імпорту BRL-CAD.

Слід зазначити що, позначення -g у кінці кожного імені перетворювача вказує, що перетворений файл матиме формат стандартної геометрії (.g) BRL-CAD. Нижче буде описано тільки перетворення з самих популярних форматів у формат моделювання який підтримує система BRL-CAD.

2.1 Перетворення з формату AutoCAD DXF

DXF – це власна специфікація даних Autodesk Inc., розроблена для підтримки зв'язків з AutoCAD, популярним програмним забезпеченням САД компаній. Формат, який було впроваджено в багатьох системах САПР (особливо тих, що працюють на ПК), є простим та відносно обмеженим, однак він перетворився на усталений засіб обміну технікою малюнків. Формат DXF дозволяє вказати одиниці виміру, що використовуються у файлі DXF, проте, не всі Файли DXF містять цю інформацію.

Таким чином, якщо інформація про одиниці не знайдена у файлі DXF, міліметри (що також є типовою та базовою одиницею виміру в BRL-CAD) є передбачуваною одиницею виміру. Звичайно, якщо міліметри не є правильним вибором для конкретного користувача або програми, можна використовувати опцію s (масштаб) (рис. 18).

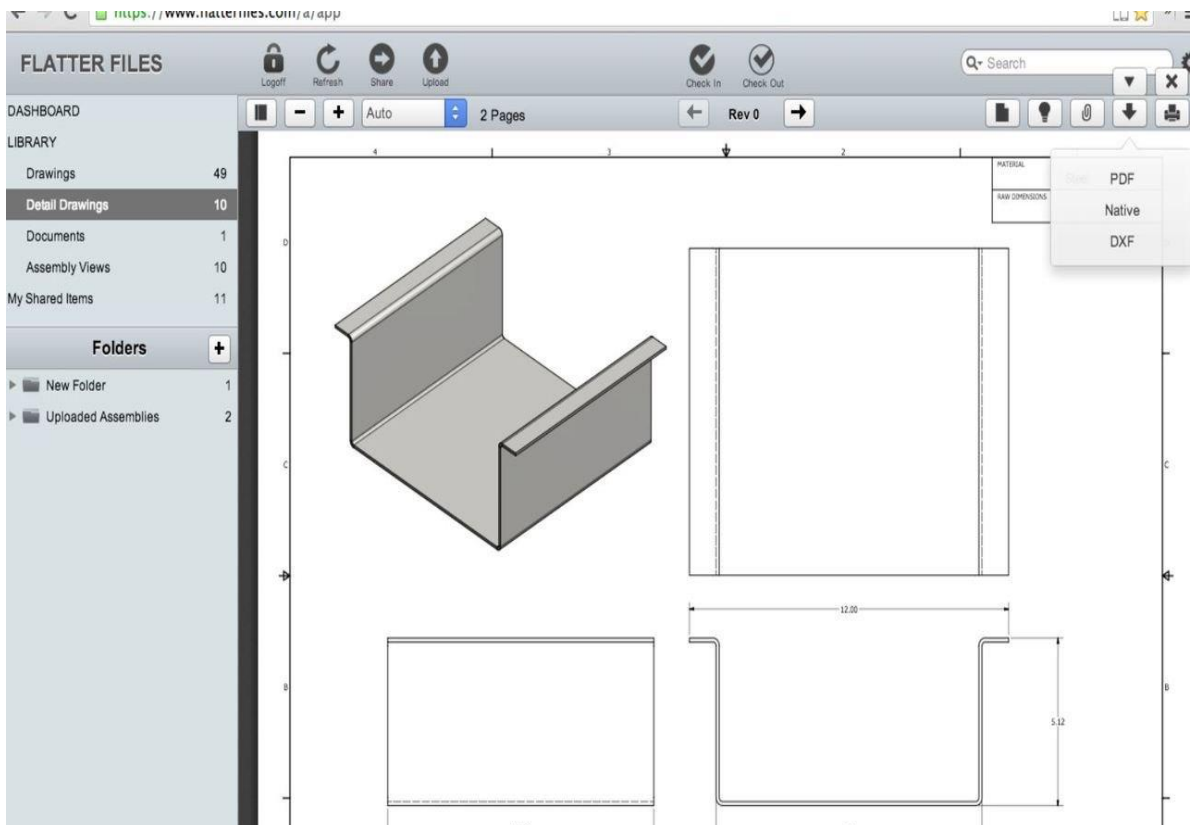


Рисунок 18 – Підготовка DXF файлу для імпорту

Ця опція та інші доступні опції для dxf-g перетворювача будуть описані далі в тексті. При використанні цього перетворювача багатокутники та сітки перетворюються на примітиви BOUND. Кожен шар стає областю. Лінії, кола та дуги стають об'єктами n-різноманітної геометрії (NMG), і точки стають сферами.

Команда для перетворювача dxf-g така:

```
dxf-g [параметри] input.dxfoutput.g
```

Параметри команди такі:

- *c* – вказує, що використовується лише перший колір, який зустрічається для шару у файлі DXF; додаткові кольори для того самого шару ігноруються [15]¹⁾.
- *d* – налагодження; виробляє дамп ядра при відмові.
- *s* – застосовує вказаний коефіцієнт масштабу до даних DXF (приймає аргумент).
- *t* – встановлює відстань допуску (у міліметрах) (за замовчуванням 0,005) (приймає аргумент).

2.2 Перетворення з формату STL

Формат STL був розроблений компанією 3D Systems, Inc. у 1980-х роках для використання зі своїм стерео-літографічним апаратом (рис. 19).

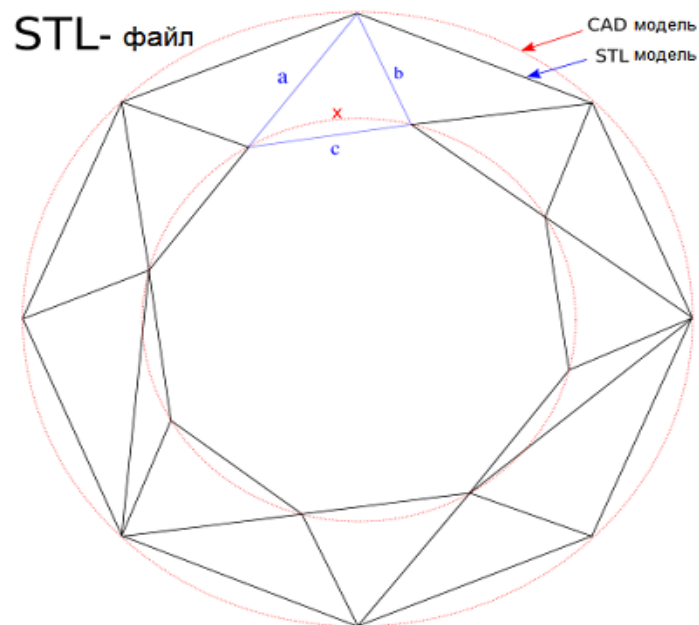


Рисунок 19 – Модель STL файлу для імпорту.

¹⁾[15] A.J.Medland, P.Burnett CAD/CAM in practice: a manager's guide to understanding and using CAD/CAM systems. KoganPage, 2003. 673 p.

Пристрій SLA створює фізичну тривимірну модель на основі файлу формату STL. Через свою простоту формат STL став галузевим стандартом для обміну тривимірними моделями. На жаль, ця простота також має певні обмеження. Формат складається лише з трикутників, і кожен трикутник представлений трьома вершинами та поверхневим вектором нормалі.

Оскільки вершини для кожного трикутника явно перераховані, а не індексовані зі списку, топологія повинна виводитися прийнятною системою, що іноді може призвести до неправильної геометрії. Файли STL можуть бути як ASCII, так і двійковими. Формат ASCII включає можливість включати більше однієї суцільної частини та необов'язкове ім'я для кожної частини, тоді як двійковий формат може лише підтримувати одну цілісну частину без імен. Конвертер stl-g перетворює формат STL у BRL-CAD. Формат STL – це цілком трикутники. Отримана база даних BRL-CAD буде складатися з одного або декількох регіонів та верхнього рівня комбінація з назвою «всі», яка містить усі вироблені регіони. Кожен регіон буде складатися з одного примітиву BOM. Зверніть увагу, що файл STL формату ASCII містить можливість містити більше однієї суцільної частини. Створені регіони будуть називатися відповідно до імені, вказаного у файлі STL, якщо в командному рядку не вказано ім'я. Якщо у файлі STL не вказано жодного імені, а користувач не вводить імені, тоді будуть побудовані регіони, створені в базі даних BRL-CAD від імені файлу STL [16]¹⁾.

2.3 Перетворення з формату Pro / E

Розповсюджуваний корпорацією Parametric Technology Corporation (PTC), Pro / E є одним із найбільш широко використовуваних комерційних

¹⁾[16]Мацяшек Л. А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML. Москва: Вильямс, 2002. 432 с.

пакетів САПР для проектування, проектування та виготовлення продукції. Довгий список основних корпорацій, які використовують програмне забезпечення Pro / E для управління життєвим циклом продукції, включає Boeing, Rolex, Audi, Dell, Nike, Maytag, Braun та Hewlett-Packard (рис. 20).

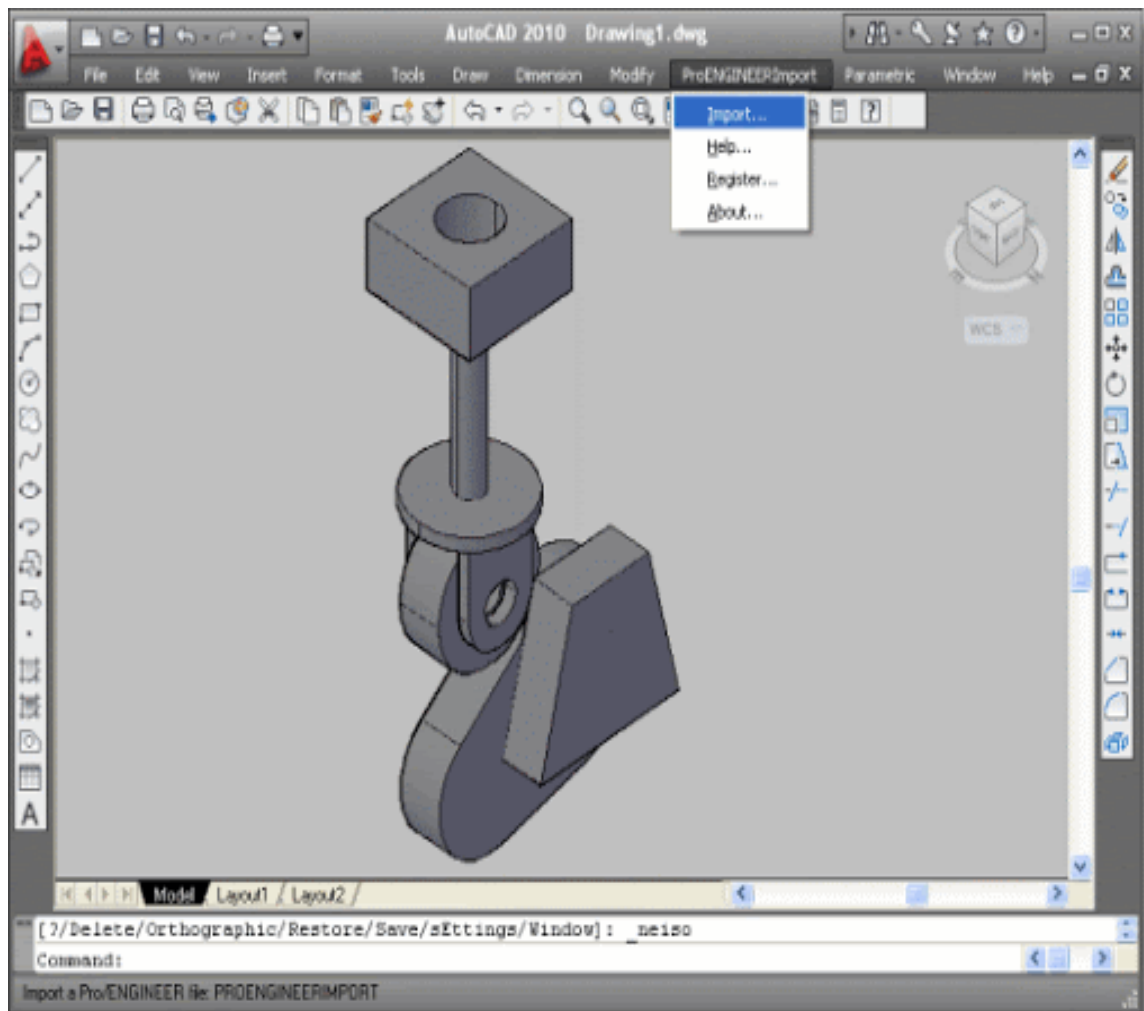


Рисунок 20 – Модель PRO/Eфайлу для імпорту.

Через популярність Pro / E у оборонному співтоваристві, перетворювач Pro / E-to-BRL-CAD є однією з найважливіших утиліт перетворення, яку пропонує BRL-CAD. Зауважте, що на відміну від перетворювачів для інших форматів, перетворювач Pro / E більше не активований командним

рядком[17]¹⁾.Цей перетворювач був написаний за допомогою модуля Pro / Toolkit Pro / E, і тому він працює як частина програма Pro / E та графічний інтерфейс.Відповідно, для того, щоб використовувати перетворювач, користувач повинен мати місце Pro / E, а також розподіл BRL-CAD.В даний час перетворювач підтримується лише на машинах Silicon Graphics (SGI) з процесорами MIPS, що працюють під управлінням операційної системи Irix.Вихідний код для цього перетворювача включений у двійковий розподіл, тому користувачі можуть скомпілювати його для різних платформ, якщо у них є модуль Pro / Toolkit для цієї платформи.

Моделі Pro / E складаються з двох елементів: деталей та вузлів.Файли деталей (які позначаються розширенням .prt) є основними будівельними елементами геометрії Pro / E.Збірні файли (які позначаються розширенням .asm) складаються з деталей та / або інших вузлів.Перетворювач створює область BRL-CAD для кожної частини Pro / E, що перетворюється, і BRL-CAD комбінація для кожної збірки Pro / E, що перетворюється.Кожен із цих регіонів буде складатися з одного примітиву BOUND.

Перетворення геометрії з Pro / E у BRL-CAD є двоступеневим процесом. Цей перетворювач спочатку створює форму баз даних BRL-CAD у форматі ASCII. Потім користувач перетворює ці бази даних у двійкову форму за допомогою утиліти asc2g.

Крім того, оскільки файли Pro / E для більшості бойових машин настільки великі (для їх зберігання часто потрібен стос компакт-дисків), цілі геометрії, як правило, не можна завантажувати одночасно. Таким чином, користувачеві BRLCAD часто доводиться перетворювати геометричну систему за системою (наприклад, двигун, трансмісія та підвіска), а потім об'єднувати (тобто об'єднувати) їх разом в один файл геометрії BRL-CAD.

¹⁾[17] Калачев О. Н. Диалоговая система KON7 для расчета размерных цепей при механообработке. Ярославль: ЯГТУ, 2005. 93 с.

Pro / E широко використовує геометрію, на яку посилаються. Посилання—це метод, за допомогою якого множинні входження об'єктів створюються шляхом багаторазового посилання на один об'єкт з різною орієнтацією та розташуванням для кожного посилання. Ці посилання дублюються у BRL-CAD з використанням комбінацій та матриць перетворень. У деяких випадках —наприклад, коли геометрія використовується з кодами вразливості, які вимагають, щоб кожен регіон мав унікальний ідентифікаційний номер —користувачам може знадобитися (рис. 21) використовувати команду `xpush` у MGEDпісля завершення перетворення для заміни посилань [18] ¹⁾.

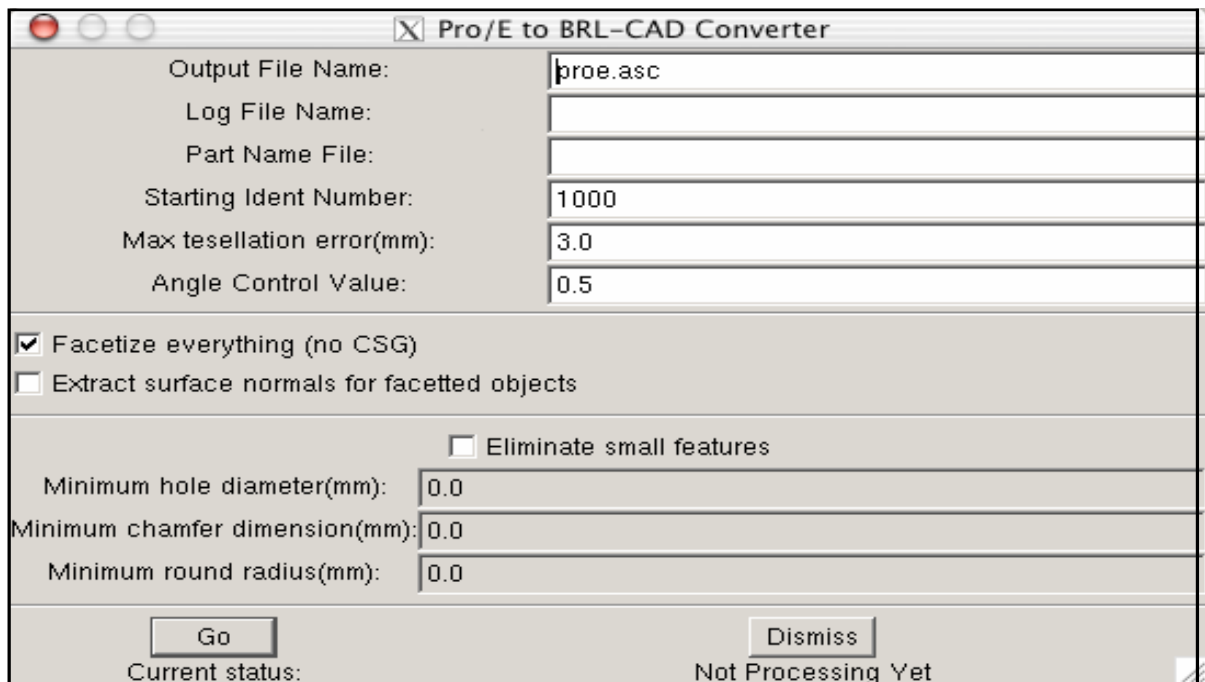


Рисунок 21 – Діаогове вікно конвертера Pro / E у BRL-CAD.

¹⁾[18] Филиппович К.В. Некоторые аспекты настройки пользовательских предпочтений в САПР-ЧПУ/2005. Пермь: ООО «ЕвразияЛимитед», 2005. 42 с.

Команда для запуску програми Pro / E визначається кожним установником. Після запуску Pro / E програма шукає файл з ім'ям protk.dat у кількох конкретних місцях, одне з яких знаходиться в поточному каталозі.

Цей файл повідомляє Pro / E про модулі Pro / TOOLKIT, які він повинен завантажувати. Існує файл protk.dat для перетворювача Pro / E-to-BRL-CAD, і він включений у розподіл за каталог pro-engineer. Користувачі повинні скопіювати цей файл у каталог, де вони будуть запускати Pro / E. Після того, як цей файл буде на місці, Pro / E завантажить конвертер під час запуску. Якщо завантаження успішно, користувачі побачать повідомлення «Встановлення перетворювача ProE-BRL завершено».

Після запуску Pro / E та завантаження модуля перетворювача користувач може відкрити будь-яку модель Pro / E, яку хоче конвертувати [19] ¹⁾. Процес перетворення починається з вибору пункту ProE-BRL у спадному меню Файл Pro / E. Потім з'явиться діалогове вікно перетворювача.

У наведеному нижче списку подано опис використання та функціонування первинних елементів у діалоговому вікні.

Зверніть увагу, що для багатьох входів у поле попередньо завантажено розумні стандартні значення.

Крім того, якщо у користувача виникає запитання щодо будь-якого з вікон входу, прапорців або кнопок, він може навести мишу на них, щоб побачити коротке пояснення їх використання:

- ім'я вихідного файлу (це ім'я файлу для отримання вихідних даних ASCII);
- ім'я файлу журналу (якщо передбачено, до цього файлу буде записано детальний журнал стану);

¹⁾ [19] Schultz D.F. What's Wrong With Postprocessors Numerical Control ComputeSciences. 1998. №1. P. 11–15.

- файл імені деталі (якщо надано, перетворювач використовуватиме цю інформацію для зіставлення деталей Pro / E та назви збірок у зазначені імена об'єктів BRL-CAD);
- цей файл (це просто текст файл із кожним рядком із зазначенням назви деталі Pro / E або збірки та об'єкта BRL-CAD ім'я, розділене пробілами. Зауважте, що назви BRL-CAD не повинні містити жодних спеціальні символи, такі як «/», «[», «]» або пробіли); початковий ідентифікаційний номер (ідентифікаційні номери будуть присвоєні отриманому BRL-CAD регіони [20] ¹⁾ послідовно, починаючи з номера, що з'являється у вікні);
- помилка максимальної теселяції – це значення (виражене в міліметрах) використовується для контролю грубість теселяції (це максимальна відстань між фактичною поверхнею та його мозаїчне наближення);
- менші значення тут призведуть до більш тонких теселяцій та збільшення трикутників (значення за замовчуванням є розумним для аналізу балістичної вразливості цілі);
- значення контролю кута (це число від 0 до 1, яке забезпечує додатковий контроль над грубістю мозаїки);
- точний зв'язок між теселяцією і це значення не вказане в документації Pro / E (значення за замовчуванням 0,5 здається, працює добре).

2.4 Перетворення з / на ASCII

З реалізацією незалежного від машини формату бази даних BRL-CAD, потреба у перетворенні у формат ASCII або з нього швидко зменшилась. Тим

¹⁾ [20] Computer-aided design system Parametric Technology Corporation (PTC), Pro / E (загол. з екрану). URL: <http://www.ptc.com> (PTC, 2003) (дата звернення: 23.04.2021).

не менше, BRL-CAD все ще підтримує перетворення ASCII, і випуск 6.0 представив новий формат ASCII.

Цей формат – це просто скрипт Tcl *, який (при виконанні належним чином уповноваженого інтерпретатора Tcl) відновлює базу даних, з якої він отриманий. На відміну від цього, старий формат ASCII складався з послідовності однорядкових записів, які відповідали старим записам формату бази даних.

Утиліта asc2g використовуються для перетворення баз даних в форматі ASCII в новий двійковий формат. Ця утиліта завжди буде створювати новий двійковий формат, але він буде приймати один з двох ASCII форматів.

Використання цієї утиліти виглядає наступним чином:

```
asc2g input.asc output.g
```

У цій команді input.asc – це файл, створений раніше за допомогою g2asc, а output.g – це ім'я де повинна зберігатися отримана двійкова база даних. Зворотня для утиліти asc2g це утиліта g2asc, яка використовується для перетворення або з поточного формату бази даних або формату бази даних реліз 6.0 в формат ASCII. Бази даних у новий формат буде перетворений у новий (скрипт Tcl) формат ASCII. Бази даних у старому форматі будуть перетворені у старий формат ASCII.

Команда для цієї утиліти така:

```
g2asc input.g output.asc
```

У цій команді input.g є базою даних BRL-CAD (старий або новий формат), а output.asc - ім'я, де розміщується вихід ASCII. Як вже згадувалося

раніше, форма виводу ASCII залежить від того, який формат бази даних використовується вхідною базою даних[21] ¹⁾.

¹⁾[21]ASCII encoding table(ASCII) (загол. з екрану).URL:
<https://ru.wikipedia.org/wiki/ASCII> (дата звернення: 23.04.2021).

3 ВИДИ ГРАФІКИ ТА ГРАФІЧНОЇ МОДЕЛІ

На сьогодні існує чотири види графіки на яких базується вся сучасна робота з графічними представленнями у цілому, це:

- растрова графіка;
- векторна графіка;
- фрактальна графіка;
- тривимірна графіка.

Кожен з цих типів графіки вже давно досяг своєї популярності у певних сферах. Всі ці типи ми зустрічаємо у нашому віртуальному житті кожен день, починаючи з моменту включення персонального комп'ютеру, та закінчуючи рендерінгом різноманітних сторінок у глобальній мережі (рис. 22).

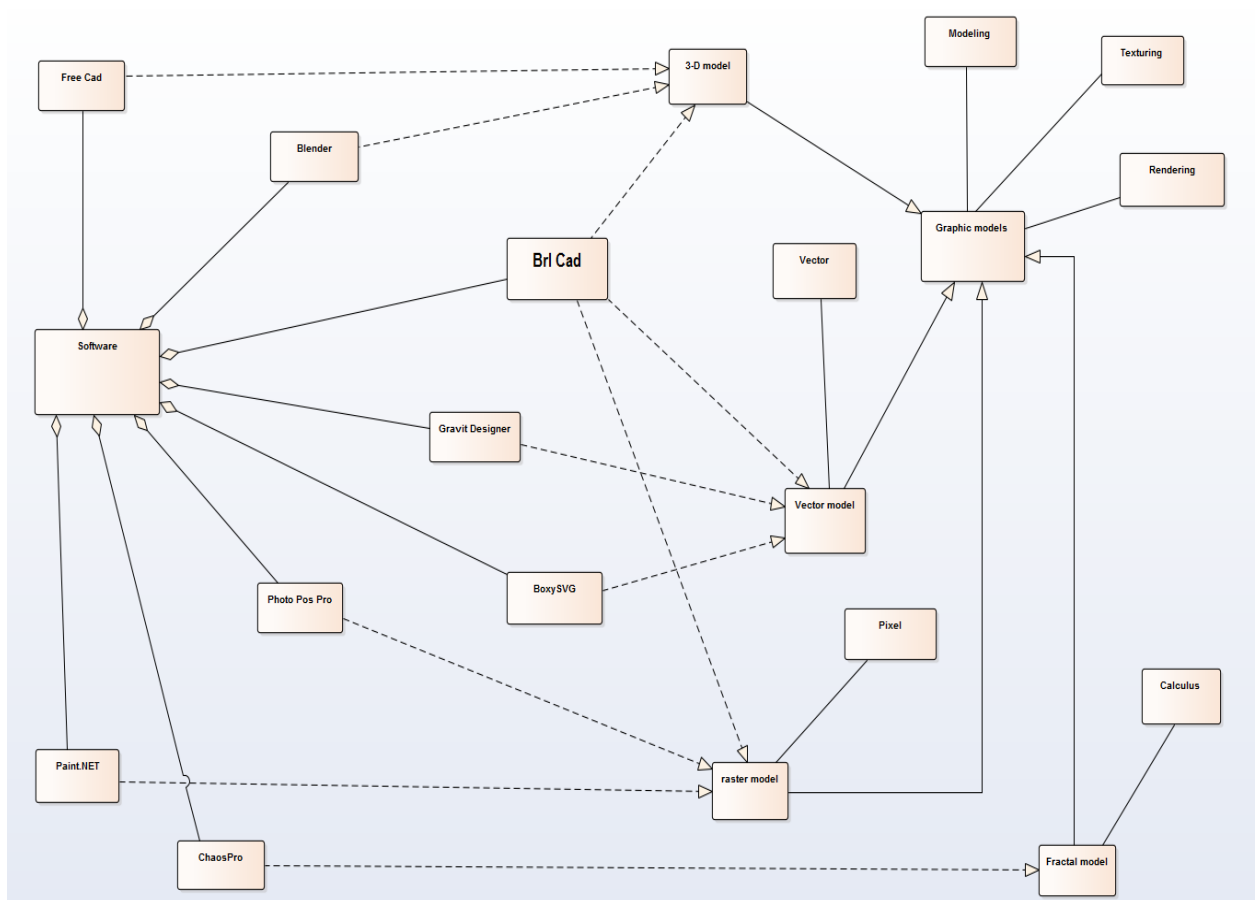


Рисунок 22 – Діаграма класів відношення графічних моделей та програмного забезпечення

Саме тому така велика їх популярність породжує неймовірний попит на створення систем для робіт з ними. Можна перерахувати безліч програмних забезпечень по роботі з тим чи іншим типом графіки, але далі буде представлена найпопулярніша вибірка засобів не розділяючи їх на типи:

- Free CAD;
- Blender;
- Auto CAD;
- BRL-CAD;
- Photo Pos Pro;
- Adobe Photoshop;
- Paint Net;
- Chaos Pro;

Слід також зазначити, що найпопулярнішим видом графіки, саме для комп'ютерного моделювання 3-D систем та автоматизації різноманітних наукових процесів, є тривимірна. Це досягається завдяки можливості з її допомогою найкраще описувати системи реального світу у цифровому форматі, а тому і проводити опити з найменшими погрішностями.

3.1 Векторні моделі

Векторна графіка, є доволі потужним інструментом, для створення зображень, тому що в її основі лежить спосіб конструювання об'єктів за допомогою різноманітних ліній (векторів, вигнутих ліній, прямих ліній та інших).

Головна перевага векторної графіки – можливість зміни розмірів зображення без втрати якості зображення [22]¹⁾. Це значно спрощує роботу з графікою та підвищує якість кінцевого результату. Ця перевага у значній мірі

¹⁾[22] Компьютерная графика. Математические основы. Версия 1.0 [Электронный ресурс] : электрон.учеб. пособие / Н. В. Соснин. – Электрон. дан. (4 Мб). – Красноярск : ИПК СФУ, 2008.

досягається завдяки підходу створення графічного уявлення за допомогою математичних формул, тому при зміні зображення, досягається постійний перерахунок значень. Саме тому файли в яких використовується векторна графіка займають досить мало місця на диску.

Недоліком даної методологіє є те, що вона не здатна передати дійсну картину світу так, як наприклад це може зробити растрова графіка. Також слід зазначити, що розробляти ПЗ для роботи з векторною графікою, це дуже складна та неблагодатна робота, все те через те, що будь-які бібліотеки фільтрів, є зовсім безкорисними при роботі з вигнутими лініями. Все це доповнюється тим фактом, що проводити будь які зміни з зображенням, можливо тільки з стандартного редактору, в якому починалась робота.

Слід зазначити, що існує певний шаблон, за яким створюються майже всі програмні засоби для роботи з векторами. Він складається з таких інструментів:

- крива Безьє (інструмент, який дозволяє створювати фігури будь якого типу, а також змінювати лінії);
- заливка (як і во всіх інших редакторах, дозволяє наповнювати фігуру будь-яким кольором);
- текст (дозволяє створювати з букв, та слів художні об'єкти);
- набір з геометричних фігур, для об'єднання створення об'єктів;
- можливість «малювання»(наприклад інструмент карандаш);

3.2 Растрові моделі

Найбільшого поширення набули растрові формати, саме з їх використанням зберігаються різні фотографії, а також інші графічні растрової графіки дозволяє створювати реалістичні фотографічні зображення, які використовуються практично в будь-якому журналі, газеті і навіть на web-сайтах в Інтернеті. І якщо вам захочеться помістити в комп'ютер свою фотографію, вона теж буде збережена в растровому форматі.

Найважливішим плюсом растрової графіки виступає те, що в цьому форматі можливо зберігати графічні зображення, які мають сходження з реальним світом. Також слід зазначити, що ця графічна модель має доволі високу деталізацію, та має можливість до передачі реалістичних градацій кольорів. Ще однією перевагою є той факт, що усі принтери дуже якісно печатають ці графічні моделі, цей ефект досягається за рахунок того що сучасні принтери працюють за принципом «точка на кожен елемент картинки». Саме цим обумовлене таке широке розповсюдження растрової графіки у світі. На сьогодні існує безліч різних редакторів по роботі з нею. Найпопулярнішим програмним забезпеченням відповідним до цього критерію є AdobePhotoshop.

За всіх своїх плюсів растрові зображення, що використовують велику бітову глибину (16 або 24 біт на піксель), займають дуже багато місця. До того ж для того, щоб редагувати високоякісні фотографії, вам знадобиться досить потужний комп'ютер з великою кількістю оперативної пам'яті, а також добрий процесор. Ще один мінус полягає в тому, що для растрового графічного файлу завжди використовується фіксована кількість пікселів, саме тому для користувача не можливо досягнути такого кейсу, де він зможе редагувати частини зображення окремо, а не увесь файл разом.

Растрова графіка має свою найвищу популярність при роботі саме з фотографіями, надав їм найвищу точність та реалістичність, вона дозволяє редагувати їх на найвищому рівні якості.

3.3 Тривимірні моделі

Тривимірна графіка – це певний вид комп'ютерної графіки, який складається з сучасних інструментів для розробки та прорисовки тривимірних моделей.

3D–зображення не складно відрізнити від двовимірного, оскільки воно включає створення геометричної проекції 3d-моделі сцени на площину за

допомогою спеціалізованих програмних продуктів. Модель, що отримується, може бути об'єктом з реальної дійсності, наприклад модель будинку, автомобіля, планети, або бути абсолютно абстрактною. Процес побудови такої тривимірної моделі отримав назву 3D моделювання і спрямований, перш за все, на створення візуального об'ємного образу об'єкта, що моделюється.

Тривимірні моделі оточують нас повсюди, а найбільшого свого поширення вони досягли у таких сферах сучасного життя, як кінематограф і мультиплікація, рекламні ролики, моделювання дизайну різного плану, наукові дослідження у цифрових сферах, та інші(рис. 23).

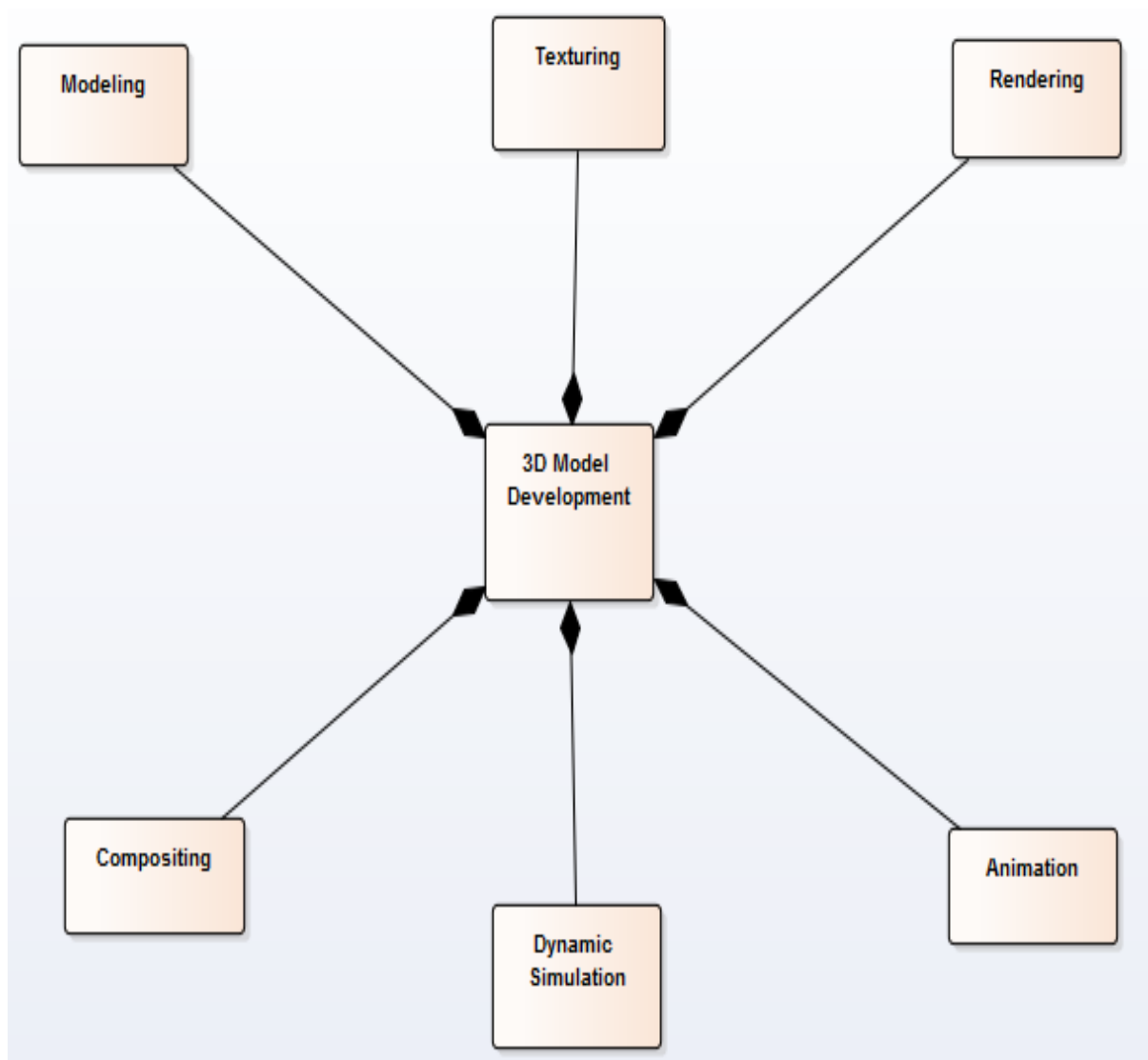


Рисунок 23—Діаграма класів для 3Dмоделювання

3D-графіка зазвичай має справу з віртуальним, уявним тривимірним простором, який відображається на плоскій двовимірній поверхні дисплея або аркуша паперу. В даний час існує кілька методів відображення тривимірної інформації в об'ємній формі, хоча більшість з них представляють об'ємні характеристики дуже умовно, оскільки працюють зі стереозображенням.

Щоб отримати тривимірне зображення на площині, необхідно виконати наступні дії:

- моделювання (створення віртуального простору, а також задачі тривимірного моделювання в залежності від мети);
- текстурування (проектування текстур на створеному віртуальному просторі в залежності від їх координат);
- освітлення (міри прийняття користувачем рішень по налаштуванню світла для моделі, все це досягається завдяки роботі з тінями);
- анімація (перенесення динамічних властивостей об'єктів реального світу на об'єкти моделювання, з метою аналізу їх поведінки);
- динамічна симуляція (основна логіка по розрахунку усіх сил, які впливають на цільову модель, за допомогою фізичних та математичних формул);
- рендерінг (побудова проекції відповідно до обраної фізичної моделі);
- композитінг (добробка тривимірної моделі);

4ПРИНЦИП ОРГАНІЗАЦІЇ РЕНДЕРИНГУ У САПР BRL-CAD

В цьому дипломному проєкті розглядається виключно моделювання тривимірних систем, за допомогою САПР, а якщо точніше відкритої кросплатформеної САПР BRL-CAD.

На жаль, єдиного загально визнаного методу створення тривимірних моделей у САД системах не існує. Насправді, як відомо, професійні архітектори використовують безліч унікальних методів для досягнення рівноцінних результатів. Тим не менш, існує кілька основних етапів або процедур, які зазвичай використовуються більшістю моделей для створення точних, реалістичних та корисних геометричних зображень вчасно та ефективно.

Ці кроки можна описати різними способами, але для зручності їх можна узагальнити на наступні п'ять категорій і представити аббревіатурою M-O-D-E-L:

- M – полегшення (або збір / перетворення) даних;
- O – Організація структури;
- D – розвиваюча (або будівельна) геометрія;
- E – Електронне оцінювання (або перевірка) геометрії на правильність;
- L – Logging (або створення) документації.

Процес моделювання можна розглядати як колесо-візок з п'ятьма спицями. Кожна спиця виходить за межі внутрішньої маточини – місія моделі і однаково важлива для надання колесу міцності та функціональності.

Крім того, хоча загальноприйнятим є розгляд кроків у тому порядку, в якому вони перераховані (тобто M, то O, то D, то E, то L), процес моделювання є динамічним, і не є незвичайним, що певна фаза відбувається в іншому порядку, повторюється або взагалі пропускається в процесі розвитку проєкту.

Наприклад, фаза організації часто є першим кроком у великих або складних модельних проєктах, оскільки вона допомагає архітектору створити

деревоподібну структуру, яка допоможе йому у зборі / вимірюванні правильної (або потрібної кількості) даних.

Крім того, модельєр часто виявляє відсутні або неточні дані на етапі розробки геометрії, що вимагає повернення до фази вимірювання. Нарешті, у випадках, що стосуються перетворення геометрії з іншого джерела, вимірювання та – етапи розробки можуть бути не застосовними, і модельєр може перейти безпосередньо до фази оцінки.

Незалежно від того, чи призначена модель для балістичного аналізу, радіолокаційних досліджень або чогось ще, ціль моделі повинна бути основою для визначення того, як повинні проводитися всі частини процесу моделювання.

Це включає в себе рівень деталізації, якого повинен досягти розробник моделей: деревоподібну структуру моделі, кількість часу моделювання, яке повинно бути виділено, типи валідації та верифікації, які повинна мати модель, і навіть спосіб створення документації і зареєстрований. Цей момент може здатися очевидним, але невизнання цілі може призводити до втрати часу і ресурсів і, в кінцевому підсумку, до неефективної моделі.

Якщо архітектор не створює концептуально нову геометрію, він повинен працювати з різними джерелами інформації для побудови моделі. У деяких випадках існують креслення або механічні креслення. В інших випадках поїздка на полігон потрібна для фізичного виміру об'єктів і орієнтації.

Геометрія існує в іншому форматі САПР і повинна бути перетворена в формат в потрібний архітектору формат.

Кожен тип джерела вимірювань має свій набір переваг і недоліків в залежності від точки зору розробника і / або користувача.

Витрата часу на складання деревної структури об'єкта перед його побудовою – ще один важливий крок у процесі моделювання, особливо якщо об'єкт, що будується, є складним, якщо час моделювання та ресурси

обмежені, якщо робота виконується в команді, або якщо модель буде передана комусь іншому пізніше (рис. 24).

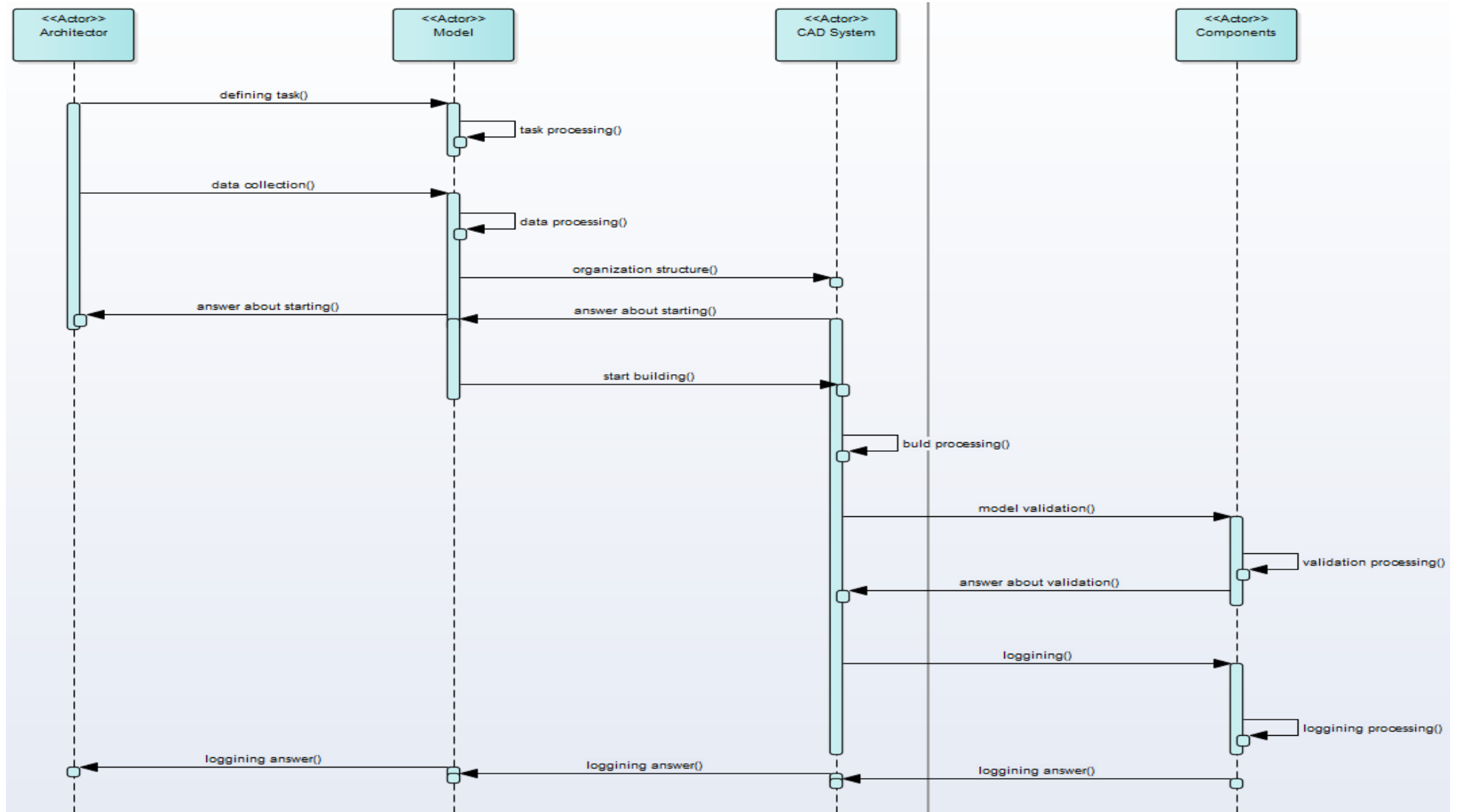


Рисунок 24 – Діаграма послідовності методів створення 3D-структур

4.1 Розробка геометрії

Зрештою, основою процесу моделювання є власне побудова геометрії. Усі найкращі вимірювання, організація, оцінка та документація були б неефективними, якщо геометричні фігури, які складають модель, не побудовані або були побудовані неправильно.

По суті, є два кроки до розробки геометрії: (1) створення геометрії та (2) позиціонування геометрії [23]¹⁾. Звичайно, як і на всіх інших етапах процесу моделювання, є різні варіації думки щодо того, як ці кроки мають бути виконані, і кожен метод має свій власний набір переваг та недоліків.

Фактори, які необхідно враховувати, вирішуючи, які методи використовувати, включають у себе зручність розташування тривимірної моделі та спосіб (наприклад, геометрія будівлі в початковій точці або в порядку, який використовує попередньо визначені вимірювання або математичні розрахунки), також кількість реплікацій об'єктів, які знадобляться в моделі, легкість редагування однієї або кількох реплікацій, місце для зберігання, час підготовки/рендерингу і так далі.

Нижче наведено кілька загальних порад щодо ефективного розвитку геометрії в BRLCAD:

- створення основної структури моделі, рекомендується створити побудову з «головної» моделі цільового об'єкта— це може бути сама велика частина (наприклад: модель регіону, деталь, найбільш центральна частина щодо інших частин моделі, або деталь розташування якої представляє собою виступаючий кут або точку, так само, як і на конвеєрі виробника: створення основної структури моделі спочатку забезпечує загальну систему координат моделі для решти менших другорядних частин для посилення);

¹⁾ [23] Чмырь И. А., Ус М. Ф. Моделирование систем в среде UML (Unified Modeling Language): конспект лекций. Одесса: ОГАХ, 2001. 70 с.

- для проектів, у яких кілька моделей одночасно працюють над окремими деталями, починаючи з основної конструкції, інші елементи можна встановлювати на місце або встановлювати відразу після завершення (ця практика є більш ефективною, оскільки вона виключає наявність зайвих частин, що плавають навколо, що очікують на розміщення, а також забезпечує кращу картину завершення моделі протягом усього проекту);
- розуміння чотирьох рівнів моделювання та їх відмінностей (всі моделі, створені в BRL-CAD, будуються в межах чотирьох його рівнів моделювання: (1) примітивний рівень, (2) рівень комбінації, (3) рівень регіону та (4) рівень складання. Знання характеристик цих рівнів моделювання є одним із перших ключів до розробки ефективною геометрії);
- знання основних примітивів та їх обмежень (одним ключем до хорошої побудови моделі є розуміння необхідних введених даних, параметрів редагування, геометричних характеристик та відносних переваг/недоліків основних «будівельних блоків» пакета – примітивних форм. Хоча в даний час пакет містить понад 20 первинних примітивів (також ще десяток примітивів для розробки/спеціального використання), лише деякі з цих примітивів використовуються на регулярній основі);
- використання найкращих команд для створення примітивів (на додаток до розуміння основних будівельних блоків і рівнів моделювання пакета, важливо розуміти поведінку та переваги/недоліки його основних «інструментів» для створення);
- використання правильної команди побудови в потрібний час може максимізувати ефективність моделювання (в деяких випадках використовуючи дані з попередньо створеної геометрії та зберігаючи вимірювання та/або час введення);

- створення об'єктів в найзручнішому місці: хоча системи координат різняться залежно від типу ситуації (наприклад, перетворена геометрія або групове моделювання, де встановлено певну орієнтацію, моделі BRL-CAD, як правило, центруються в початку координат ($xuz = 0\ 0\ 0$), де вісь $+X$ знаходиться спереду, вісь $+Y$ ліворуч, а вісь $+Z$ – вгору);
- для об'єктів, які є симетричними за своєю природою, ця практика може скористатися перевагами операцій дзеркального відображення BRL-CAD і надати простіші контрольні номери для об'єктів, які є більш складними за композицією та/або орієнтацією (у деяких випадках, однак, розробник моделі вважає, що доцільніше будувати об'єкти на місці в моделі, сюди входять випадки, коли раніше створені об'єкти пропонують зручні контрольні номери для розташування/орієнтації об'єкта, а також випадки, коли дотики та інші необхідні розрахунки було б важче отримати з об'єктом у початку координат);
- створення кількох входжень об'єктів найбільш вигідним способом (іноді розробнику моделей доведеться зробити кілька входжень об'єкта, існують дві основні методики створення таких колекцій: перший включає фактичне відтворення геометрії, а другий включає посилання на спільну геометрію.
- незалежно від використовуваної техніки, розробник моделі зазвичай починає зі створення прототипу об'єкта (у першій техніці розробник моделювання створює повні копії об'єкта, який потрібно відтворити: кожна копія потім розміщується всередині моделі, у другій техніці створюється комбінація «посилання», яка містить лише прототип, потім ця комбінація розміщується в моделі);
- зберігання обмежувальних примітивів, якомога меншими та компактними (можна використовувати великі примітиви для досягнення перетинаних форм у BRL-CAD, використовуючи

обмеження або примітиви віднімання, які значно виходять за межі позитивного об'єму області, зазвичай не рекомендується, оскільки це сповільнює програми трасування променів і може ускладнити перегляд геометрії каркаса, особливо в складна база даних);

- моделям потрібно вміти імітувати рух частин і персоналу (рис. 25) або показувати унікальність переглядів для презентаційних цілей (на жаль, розробник моделі (або навіть користувач) не завжди може передбачити всі можливі варіанти використання на початку проекту, тому розумно, особливо в організаціях, які використовують багато різних типів додатків моделей, намагатися розробити та побудувати моделі з думкою про те, що в якийсь момент їх, можливо, потрібно буде сформулювати, анімувати або представити в різних конфігураціях).

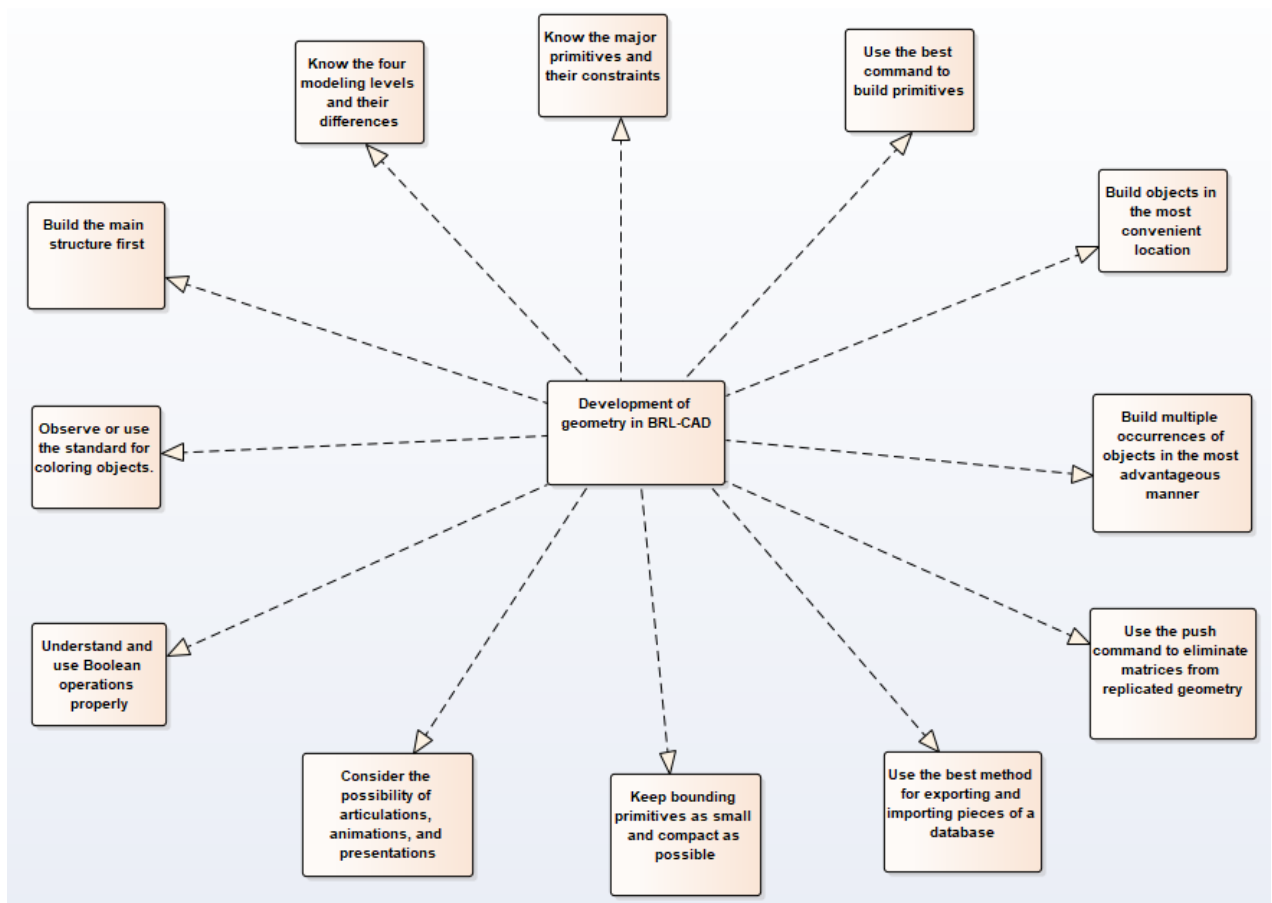


Рисунок 25 – Діаграма класів розробки геометрії в BRL-CAD

4.2 Оцінювання геометрії

Оцінювання правильності геометрії є важливим супутником побудови реальних моделей. Фактично, без перевірки правильності розташування та композиції геометрії процес моделювання фактично не був завершений. Оцінка виконується в наступні два моменти процесу моделювання: (1) після того, як окремі об'єкти побудовані та організовані в області та комбінації, і (2) після того, як модель повністю побудована. В обох випадках головною метою оцінки є виявлення будь-яких помилок у вимірюванні, логіки чи введенні даних, які б зробили модель недійсною або нереалістичною. Однією з поширених помилок, яку виявляє процес оцінки, є перекриття геометрії. Перекриття – це фізичне порушення, яке виникає, коли два або більше об'єктів (областей) займають однаковий об'єм у просторі. Хоча ця умова іноді прийнятна (наприклад, при моделюванні об'ємів повітря), вона створює неточності, коли згодом аналізується геометрія.

Існує кілька способів оцінки геометрії BRL-CAD. Вони включають в себе (1) відтворення зображення за допомогою `rt`, (2) перевірку на перекриття за допомогою `rtcheck` (або менш відомого/використовуваного інструменту перекриття `g_lint` або `MGED`) і (3) перевірку неправильного складу матеріалу (наприклад, щільності) за допомогою `rtweight`. Загалом, як тільки область буде завершена, хороша практика трасування променів за допомогою `rt`. Це дозволяє користувачеві візуально перевірити, чи вся булева логіка правильна, а геометрія не має очевидні проблеми. Якщо є щось сумнівне, то це можна детальніше вивчити за допомогою трасування променів, що виділяє цю конкретну область. Зауважте, що візуалізація також є хорошим способом порівняння геометрії з кресленнями, ескізами, фотографіями чи зображеннями з інших систем САПР. Оскільки підкомпоненти організовуються в збірки, а складність геометрії збільшується, доцільно використовувати `rtcheck` (або `g_lint` або інструмент перекриття), щоб

допомогти знайти будь-які помилки в геометрії та ізолювати будь-які проблеми.

4.3 Розміщення фігур у тривимірному просторі в BRL-CAD

Створіть arb8 з графічного інтерфейсу. Назвіть форму arb8.s. Перейдіть до Перегляд і виберіть Спереду. Перейдіть до пункту Редагувати на панелі меню. Arb8 потрібно збільшити, тому в меню Редагувати виберіть Масштабувати. Помістіть вказівник миші у верхню половину екрана, щоб зробити arb8 збільшити і натискати середню кнопку миші, доки сторони arb8 не торкнуться кожної сторони екрана. За допомогою клавіші SHIFT та лівої кнопки миші перетягніть регулятор у потрібне положення. У меню «Редагування» виберіть «Перемістити лицьову частину», а потім «Перемістити лицьову частину 4378». Помістіть вказівник миші в нижню половину екрана та клацніть середню кнопку миші, доки arb8 не буде приблизно товщиною стільниці. Поверніться до Редагувати та прийняти зміни, а потім використовуйте SHIFT та будь-яку клавішу миші, щоб розташувати стільницю так, щоб вона виглядала приблизно так (рис.26).

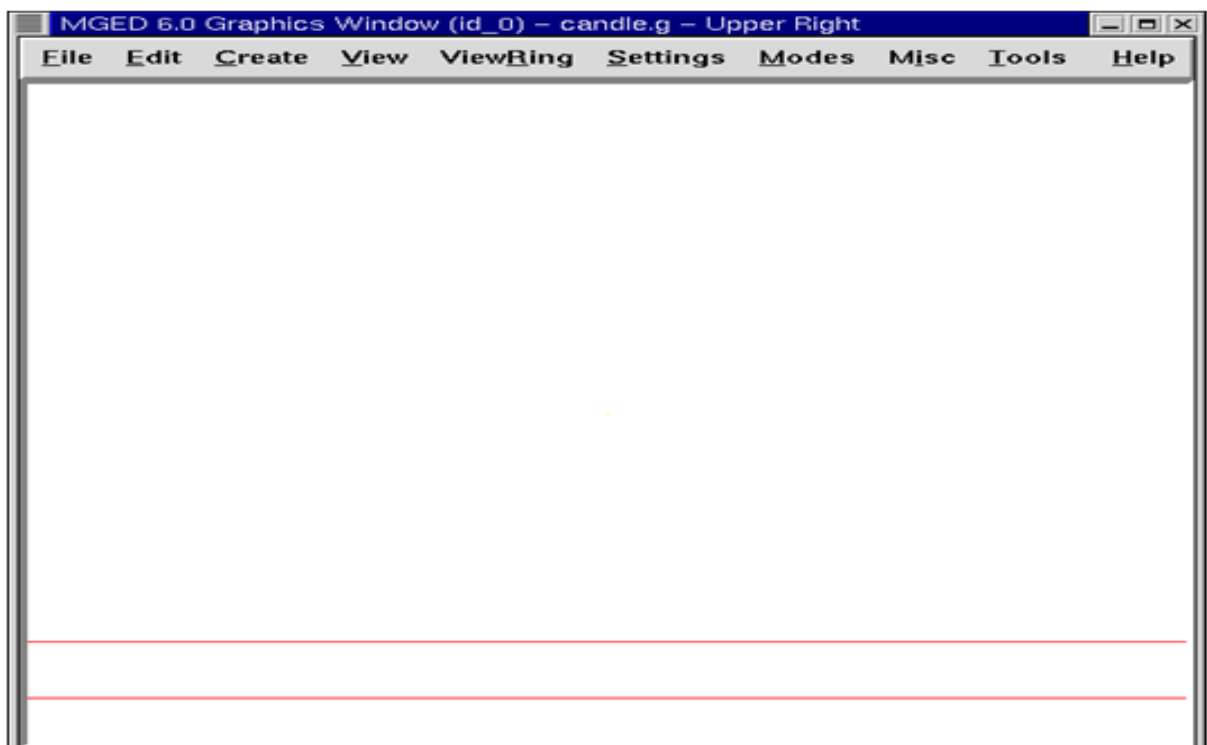


Рисунок 26 – Каркасне зображення стільниці з вигляду спереду
Створіть область стільниці, набравши в командному вікні:

```
rtable1.r uarb8.s<ENTER>
```

Створіть свічку і назвіть її eto1.s.Щоб створити нижню частину основи свічки, вам потрібно буде перевернути свічку на 180° (рис. 27).Введіть у вікні командного вікна:

```
rot 0 180 0 <ENTER>
```

Це повідомляє MGEDпро те, що потрібно обертати фігуру на 180° вздовж осі у.

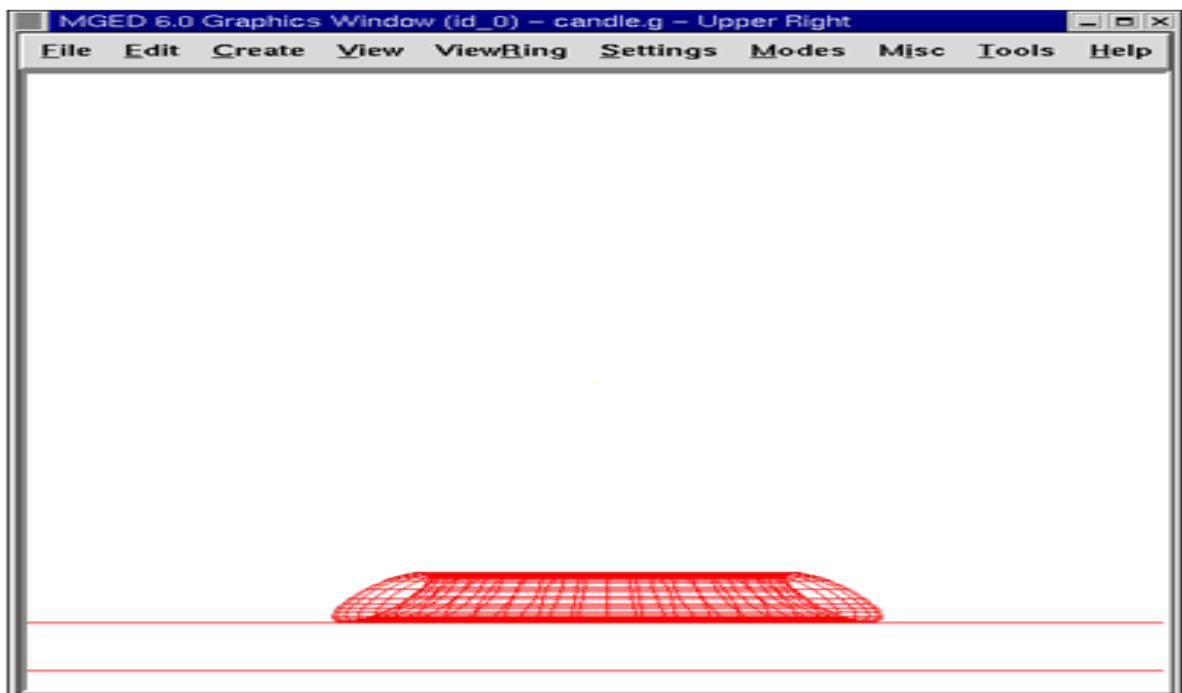


Рисунок 27– Каркасне представлення столу і свічки

Далі виберіть Scale і зробіть свічу трохи меншою за розміром за замовчуванням. Помістіть свічу на стільницю за допомогою клавіші SHIFT та лівої кнопки миші, щоб перетягнути основу в потрібне положення. Перегляньте свій дизайн з різних сторін, щоб переконатися, що свічка розташовується врівень у центрі стільниці. Натисніть прийняти, коли вас будуть влаштовувати її розмір та розміщення. Ваша база повинна бути схожа на ту, яку показано нижче.

Наступним кроком створення основи свічки є створення правильного круглого циліндра (rcc). Назвіть фігуру rcc1.s.

Зайдіть в Edit. На додток до стандартних команд вам буде представлено менше тринадцяти способів редагування цієї форми для конкретних фігур (рис.28).

Set H	Set A,B
Set H (Move V)	Set C,D
Set A	Set A,B,C,D
Set B	Rotate H
Set c	Rotate <u>AxB</u>
Set d	Move End H(<u>rt</u>)
	Move End H

Рисунок 28— Тринадцять специфічних для фігури способів редагування

Масштабуйте фігуру, доки вона не стане трохи більшою в діаметрі, ніж верхня частина eto1.s (це можна перевірити, переключившись на вид зверху).

Поверніться до Редагувати та виберіть Встановити H. Зменшуйте висоту фігури, доки rcc приблизно вдвічі перевищує висоту eto1.s. Розташуйте циліндр на підсвічнику. Перевірте розміщення rcc зверху,

ліворуч і спереду, щоб переконатися, що він відцентрований в ето. Переконайтеся, що низ gss не зовсім торкається стільниці. Прийміть зміни. Завершивши, ваш дизайн повинен виглядати наступним чином. Останній крок у виготовленні основи для свічок – це створення ще однієї свічі. Назвіть її eto2.s. Відредагуйте цю фігуру так само, як і попередню свічку, і розмістіть її поверх gss, як показано на наступному рисунку (рис. 29).

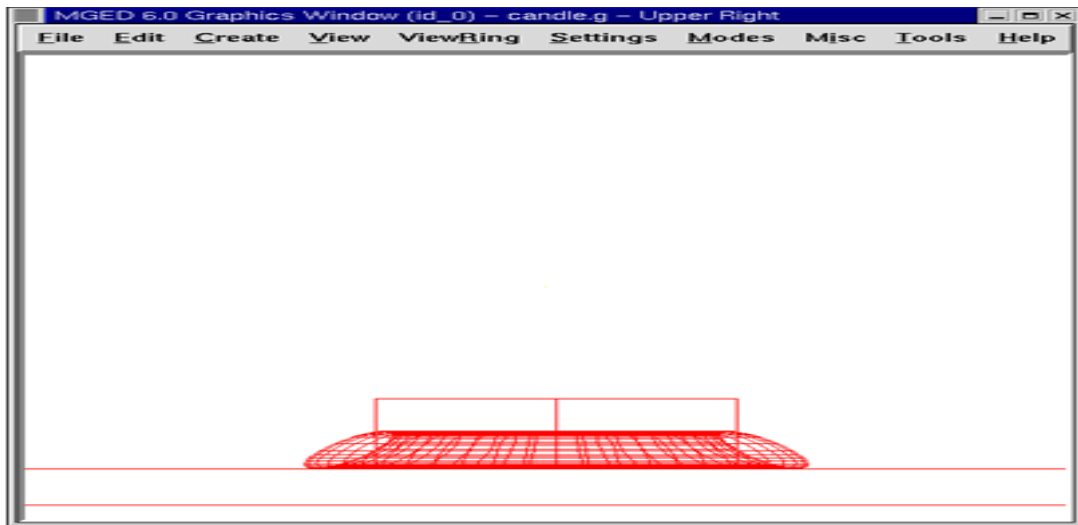


Рисунок 29 – Каркасне представлення стільниці, першої ето та першого gss

Прийміть зміни, коли закінчите. Тепер ваша основа для свічок повинна виглядати так (рис.30).

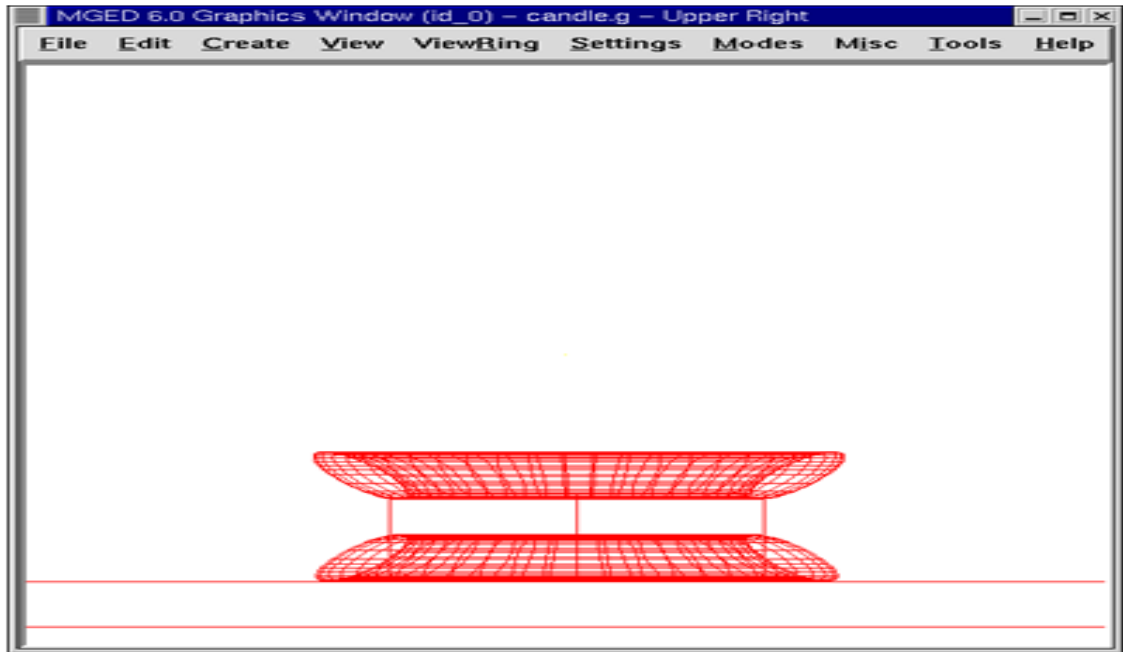


Рисунок 30 – Каркасне представлення стільниці та основи для свічок

Зробіть область з трьох фігур основи. Назвіть це `base1.r`. Ваш булевий вираз повинен читати:

```
r base1.r u eto1.s u rcc1.s u eto2.s
```

Зверніть увагу, що ми могли ще написати:

```
r base1.r u eto1.s u eto2.s u rcc1.s
```

Перший вираз бажано узгоджувати з порядком пізнішого прикладу. За мить ми захочемо видалити деякі матеріали, які нам надає `eto2.s`. Поставивши `eto2.s` останнім у списку, ми можемо легко виконати це видалення (рис. 31).

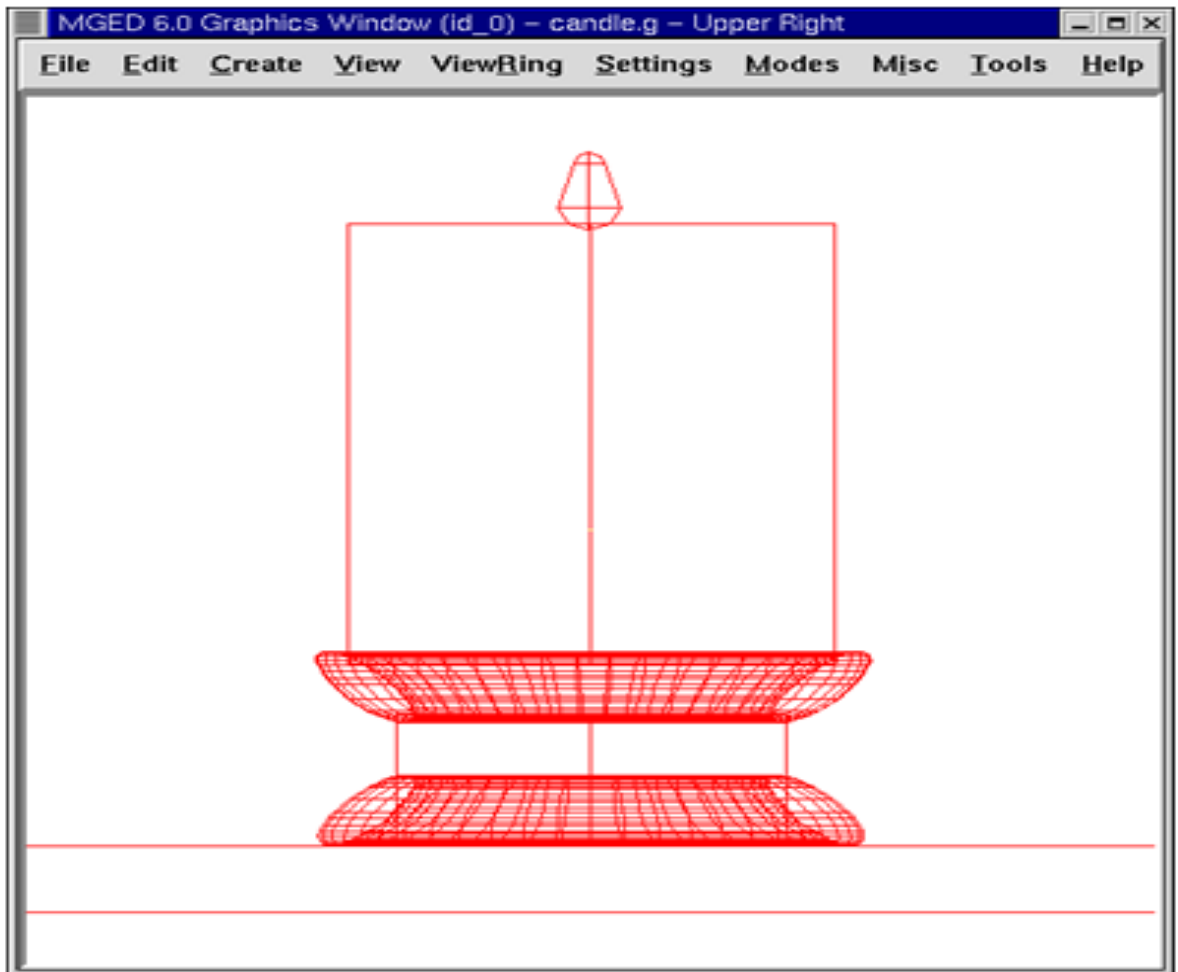


Рисунок 31 – Каркасне подання завершеного дизайну свічок

Створіть rcc і назвіть його rcc2.s. Відредагуйте фігуру так, як це було зроблено в попередньому rcc. Коли ви закінчите, це повинно виглядати приблизно так, як показано на наступному рисунку. (Примітка: Після того, як ви прийняли зміни, ви зможете отримати всю свою стільницю та свічку у вікні графіки, використовуючи клавішу SHIFT та ліву кнопку миші, щоб перемістити ваш погляд на дизайн).

Зробіть область свічки. Ваше логічне висловлювання має читати:

```
r eto1.r u rcc2.s
```

Тепер ми створюємо циліндричний виріз в основі, щоб свічка сиділа. Для цього ми можемо використовувати форму свічки таким чином:

```
r base1.r - rcc2.s
```

5 АНАЛІЗ ТА ПОБУДОВА ТРИВИМІРНОЇ ГРАФІЧНОЇ МОДЕЛІ РЕЛЬЄФУКАРПАТСЬКОГО РЕГІОНУ

5.1 UML-діаграма послідовності побудови графічної моделі

Зазвичай ДП використовується для того, щоб розробнику було легше продемонструвати візуально, як саме працює його система. Це досягається за рахунок чіткого поняття, останнім, усієї послідовності кроків для повноцінної реалізації мети [24] ¹⁾.

У ДП, яка була реалізована в цієї магістерської кваліфікаційної роботі , було відображено всі об'єкти(рис.32), від яких на пряму залежала розробка тривимірної графічної моделі регіону, а саме від:

- розробник (архітектор);
- модель регіону;
- база знань;
- компонента Mged;
- компонента Archer;
- компонента RtWizar;

У діаграмі чітко позначено лінії життя кожного об'єкта. Слід зазначити що між цими основними сутностями [25] ²⁾, встановлено зв'язок за допомогою ліній з'єднання, які називаються повідомленнями та асинхроннимивикликами. Ці зв'язки мають такі назви:

- повідомлення про проектування мети;
- процес проектування мети;
- повідомлення про збір даних;

¹⁾[24] Леоненков А.В. Самоучитель UML. Санкт-Петербург: БХВ-Петербург, 2002. 304 с.

²⁾[25] Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник. Санкт-Петербург: Питер, 2002. 656 с.

- процес збору даних;
- повідомлення про організацію структури моделі;
- процес організації структури моделі;
- повідомлення про старт будівництва моделі;
- процес будівництва моделі;
- повідомлення про додавання орієнтації у просторі;
- процес додавання орієнтації у просторі;
- повідомлення про запуск врахування;
- процес запуску врахування;
- повідомлення про додання світла та тіней;
- процес додання світла та тіней;
- повідомлення про старт валідації;
- процес валідації;
- повідомлення про створення документації;

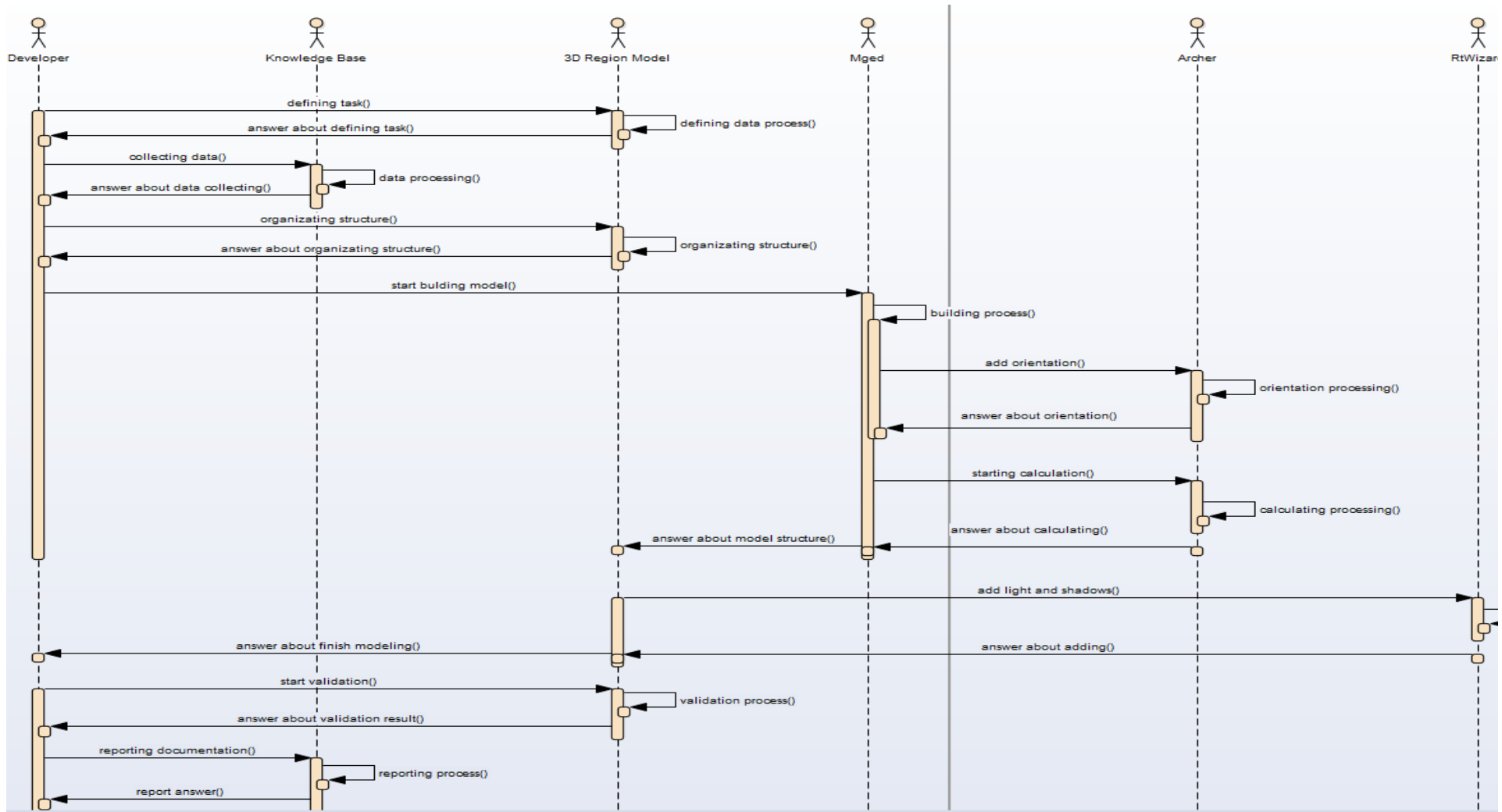


Рисунок 32 – Діаграма послідовностей для позначення кроків розробки тривимірної моделі регіону

5.2 UML-діаграма класів моделі

Діаграми класів [26] ¹⁾використовуються для того, щоб наглядно продемонструвати основні компоненти системи та їх зв'язки (рис. 33). В ролі основних сутностей [27] ²⁾ у даній магістерській кваліфікаційній роботі виступають:

- Mged
- RtWizard
- Archer
- результуюча модель
- зібрані дані
- аналіз даних
- валідація
- документування
- розрахунки координат
- розрахунки впливу зовнішніх чинників
- тіні
- світло
- візуалізація
- прості об'єкти
- оптимізація

Всі сутності зв'язані один з одним за допомоги таких залежностей, як:

- асоціація;
- композиція
- агрегація

¹⁾ [26] Фаулер М., Скотт К. UML. Основы. Санкт-Петербург: Символ-Плюс, 2002. 192 с.

²⁾ [27] Буч Г., Рамбод., Джекобсон А. Язык UML. Руководство пользователя (Сер. программистов). Москва: ДМК, 2000. 432 с.

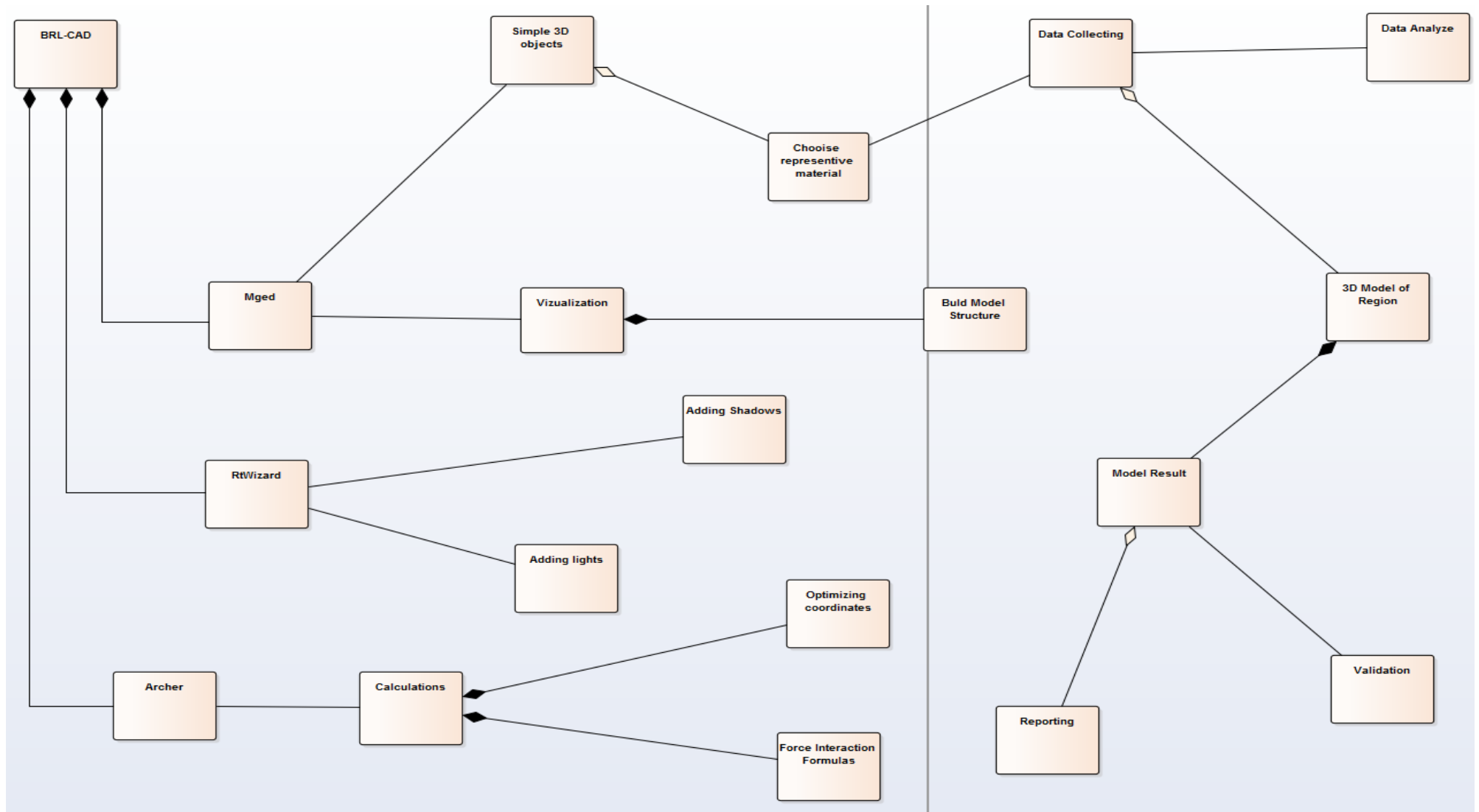


Рисунок 33 – Діаграма класів повністю описуюча структуру робочого проекту по моделюванню 3-D моделі

5.3 Основні команди і формули використані при моделюванні тривимірної графічної моделі карпатського регіону

В процесі створення тривимірної графічної моделі карпатського регіону, для даної магістерської кваліфікаційної роботи, було спроектовано цілий план дій, який допоміг мені, як розробнику реалізувати кінцеву модель.

На першому етапі, дипломним керівником було обрано тему приведену вище. Ця тема має велику актуальність на сьогодні, та набула досить великого поширення серед схожих наукових робіт, однак у дипломному проекті було принесено велику кількість наукової новизни, яка полягає у відкритті, науковому товариству, дуже якісної та потужної, а головне безкоштовної та кросплатформової САПР з підтримкою цілої купи моделей для моделювання. Також крім створення документації по використанню САПР BRL-CAD, було проведено багато додаткових та другорядних аналізів, виведено купу завдань з проведених наукових експериментів.

Далі після того, як отримано мету розробки, потрібно провести збір та аналіз потрібних для моделювання графічної моделі даних. Слід зазначити, що при зборі даних було знайдено багато формул, які в наступних кроках, допомогли в проектуванні, а головне в моделюванні цільової моделі.

Проектування структури тривимірної графічної системи у BRL-CAD включає у себе оцінку області, яка моделюється, завдяки цьому досягається найбільша точність при зіставленні з реальною областю. Наприклад при проектуванні було прийнято до уваги, такий фактор, як розділення Зовнішніх і Внутрішніх карпатських гір за допомоги Утьосової зони. Слід зазначити, що виникнення цієї області розділення, приходить на період, який проходив на нашій планеті сотні мільйонів років тому. Карпатській регіон являється дуже цікавим для аналізу і вчення. Більша частина його території заповнена горами та висотами. Завдяки тому, що багато хребтів розміщено неподалеку один від одного, регіон і славиться своєю потужною сейсмічною активністю. Коли модель, повністю спроектована і готова до моделювання,

починається етап її розробки. На цьому етапі, перш за все добавляються координати для об'єктів регіону, щоб повністю передати картину реального світу, потім починають проводитись всі необхідні розрахунки, по уникненню різного роду розсіювань. Розрахунки напряду задаються за допомоги фізичних формул і обраховуються безпосередньо в системі BRL-CAD (рис. 34).

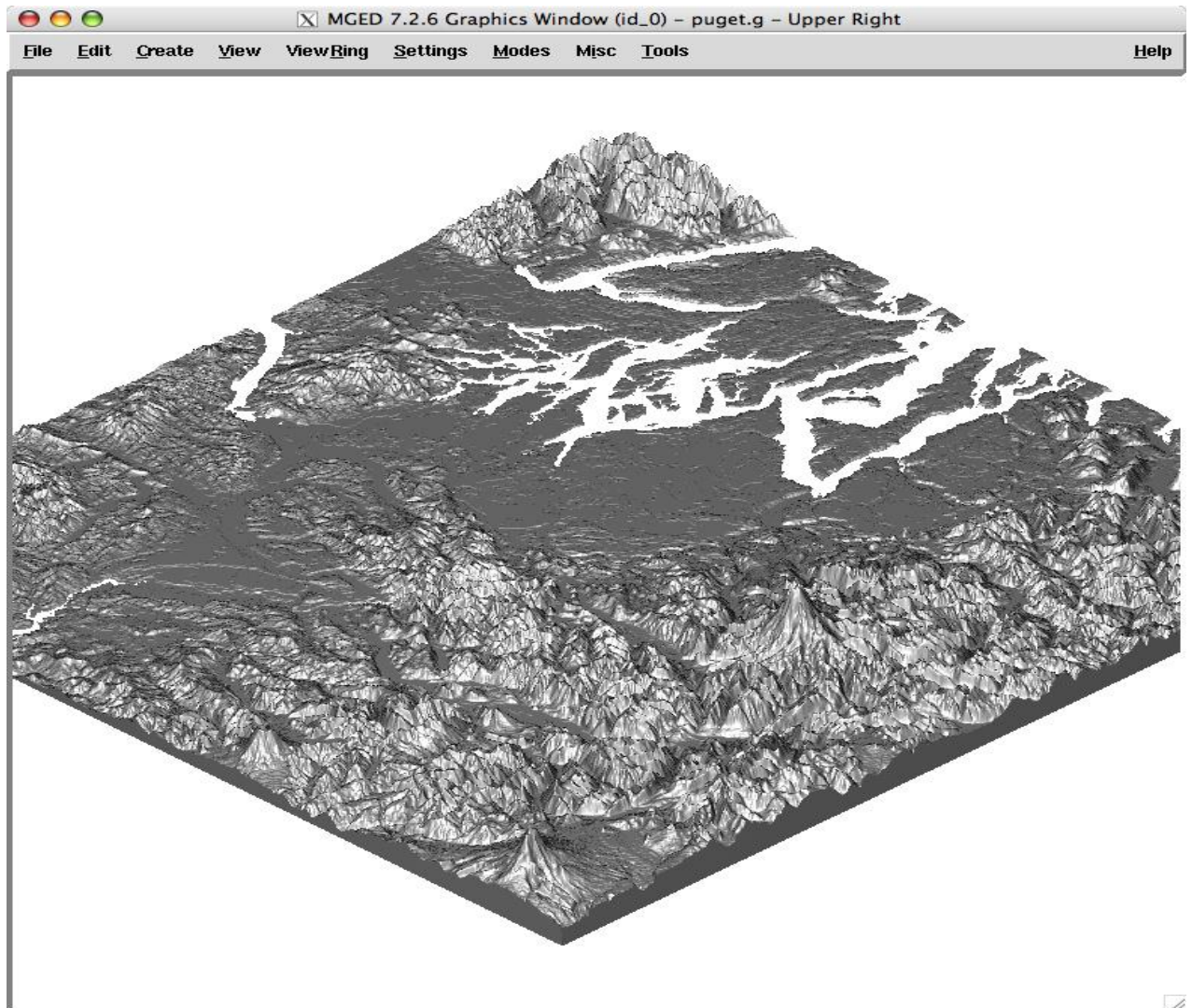


Рисунок 34– Частина тривимірної моделі карпатського регіону, змодельована у САПР BRL-CAD

Після цього йде добавлення світла і тіней. Коли створюється 3Dграфіка, освітлення є одним із найважливіших інструментів у розпорядженні

архітектора. Те, чи зможете він успішно відтворити свою модель, багато в чому залежить від вашої здатності керувати освітленням на своїй моделі та керувати ним. Погана настройка освітлення може зіпсувати чудові зусилля моделювання, а правильне освітлення може підкреслити найкращі частини сцени. Функція, яка надзвичайно корисна під час використання світла в BRL-CAD

–це можливість брати цільове світло та вимикати та повертати помічника цілі, щоб розробник міг мати повну свободу для позиціонування та обертання вогнів у своїй сцені. Він не може вмикати та вимикати цілі на вкладці «Створити», але ви можете зробити це на вкладці «Змінити». Під час роботи з вогнями важливою інформацією, про яку розробник повинен знати, є кут падіння. Кут падіння розраховується за допомогою вектора від світлового об'єкта та грані, нормалі до геометрії, на яку впливає світло. Якщо кут падіння дорівнює нулю градусів, що вказує на те, що світлові промені перпендикулярні до геометричної поверхні, геометрія буде освітлена повною інтенсивністю світла. Коли значення кута падіння підвищиться вище нуля градусів, на поверхню буде все менше і менше впливати світло. Ще одна функція освітлення, яку було використано при розробці графічної моделі карпатського регіону – це навколишнє освітлення (також називається глобальне освітлення). Навколишнє освітлення найкраще можна описати як вторинне, непряме світло, яке не надходить безпосередньо від певного джерела. Ви можете думати про навколишнє світло як про світло, яке ви отримуєте в похмурий день, коли світло падає з усіх боків. Ви можете використовувати навколишнє освітлення, щоб виділити приховані деталі або, навпаки, придушити деталі. Правильне використання навколишнього світла значною мірою сприяє фотореалізму. Найкраще збільшити навколишнє освітлення в сцені з великою обережністю. Насправді в сцену не включено додаткове освітлення, але кольори поверхні об'єктів сцени освітлені,

Було враховано такі важливі фактори, як розсіювання хмарами та опадами, та взято у розробку моделювання моделі карпатського регіону,

формулу (1.1) по оцінюванню площини через вирахування коефіцієнту розсіювання у зворотній методології.

$$\sigma_2 = \frac{\pi^5 |K|^2 D^6}{\lambda^4} \quad (1.1)$$

Також у комбінуванні з формулою (1.1), для мого магістерського, кваліфікаційного проекту, дуже важливим було урахування поняття, яке напряду відповідає за виявлення потужності радіолокатору для проведення експерименту по розрахункам якості інтенсивного розсіювання (1.2).

$$z = \frac{C \bar{P}_r r^2}{|K|^2} \quad (1.2)$$

Приведені вище формули було напряду використано при розрахунках у задачі по моделюванню обробленої моделі на основі супутникового знімку при врахуванні накладених хребтів гір на зображення (рис.35).

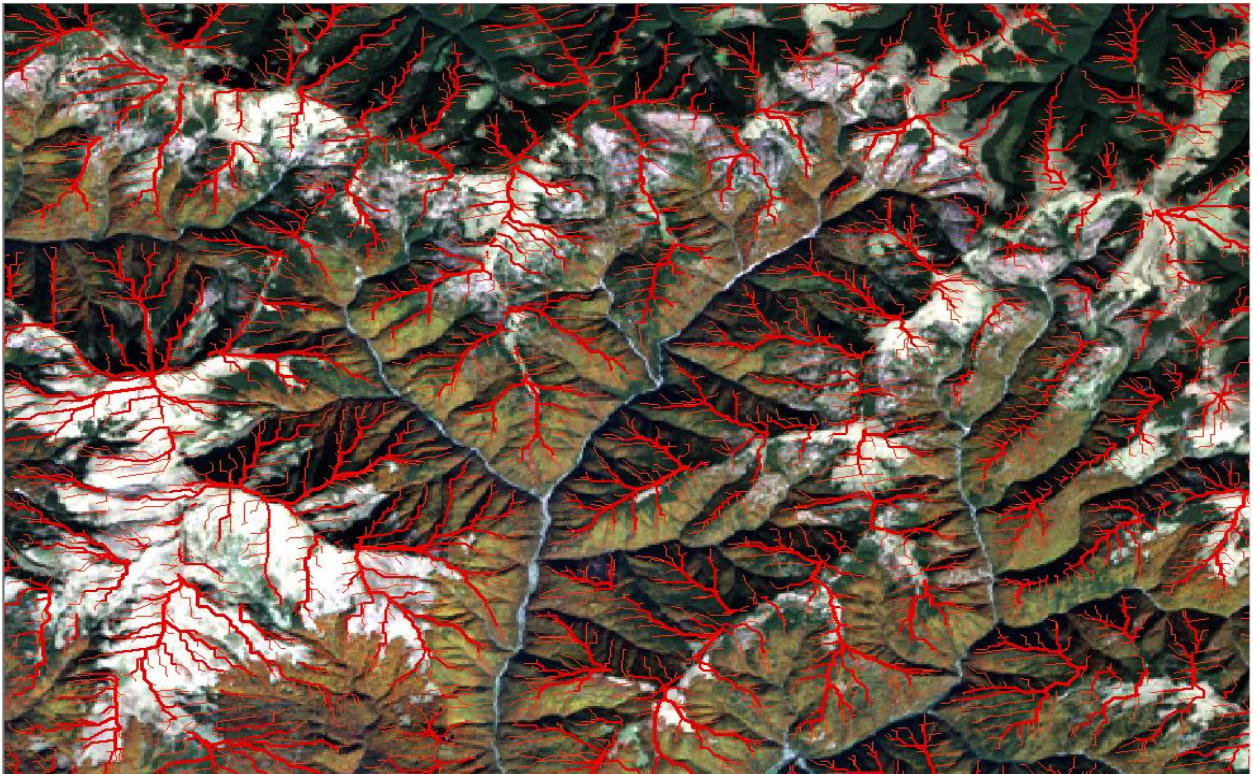


Рисунок 35– Модель карпатського регіону на основі супутникового зображення з накладенням хребтів

Розташування метеолокатора – це завдання, в якому потрібно пам'ятати багато різних нюансів, що впливають на результат. Огляд метеолокатора встановлюється обласною морфометрією. Коли середовище низинне, необхідно знайти таке становище, у якому немає недоступних секторів. У горбистих областях такі місця знаходяться нагорі пагорбів. Огляд навколишнього простору – це лише частина роботи, що розпізнає, де знаходиться метеолокатор.

Важливим рівнем хорошої станції радіолокації вважається “карта затінення”, яка показує можливу межу або найменшу потрібну висоту в залежності від можливої межі. Висота введення антени (над площиною) теж вважається одним з основних критеріїв: вона задає і шар радіолокації, і характер оцінок опадів. У межах невеликої висоти радіолокаційні промені будуть швидше за все частково, а може й повністю недоступні морфометрією або проблемами, створеними штучно(рис.36).

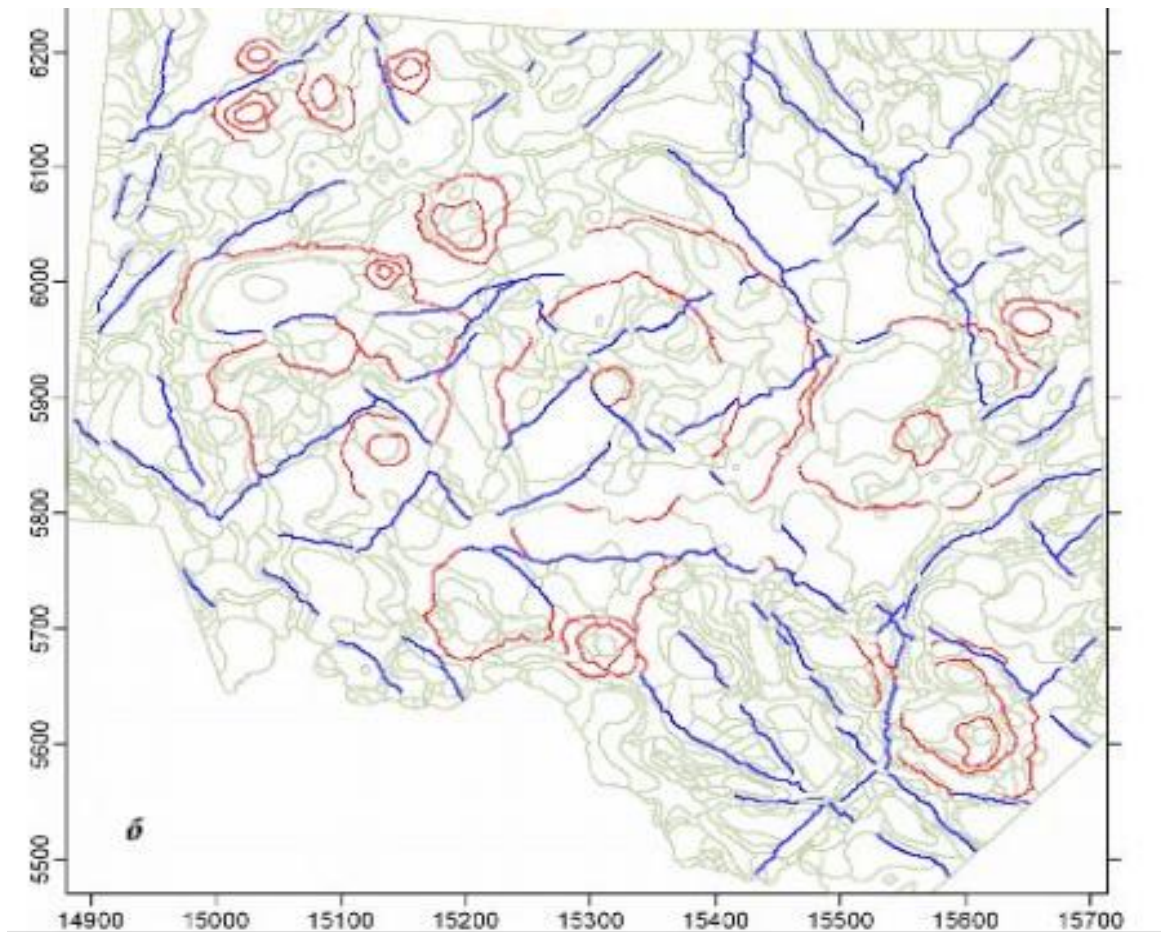


Рисунок 36 – Приклад виділених лінеаментів

Недоступність, а це теж саме що і блокування, якщо її не брати до уваги, вона може створювати деякі похибки в аналізі відбивного потенціалу локатора. Метеолокатор у Карпатській області обходиться геологічним побудовою області. Гори можуть бути навіть понад 2000 метрів на висоту над поверхнею моря, де радіолокаційний промінь з певною ймовірністю потрапляє прямо в гори, що надалі важає знайти метеорологічні цілі. Гори непроглядні для метеолокатора, з цього випливає, що йому не вдається «виявити» метеорологічні цілі у високогірних областях (рис.37).

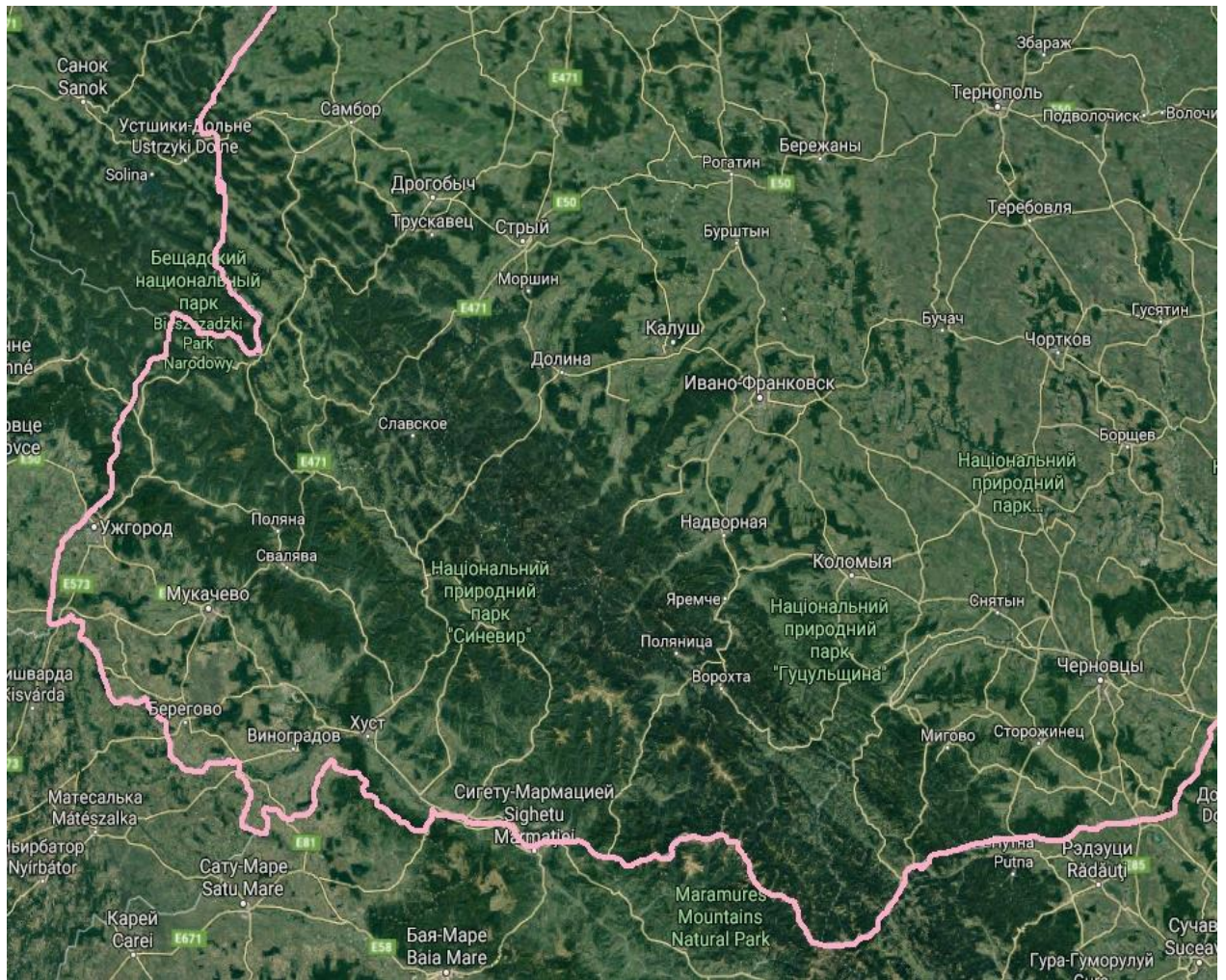


Рисунок 37 – Супутниковий знімок Карпатського регіону України

У горбистій ділянці мало добрих місць для розташування метеорадіолокаційної станції. Екрановане злиття гір на справу МРЛ - це величезні кути закриття положень і велика кількість відображень від найближчих об'єктів на екранах МРЛ, це знижує можливість їх виявити. Труднощі з розташуванням МРЛ стають явними, коли кількість доріг знижується, відбувається це через робіт, що відбуваються там, які загрожують обвалами, завалами, а іноді навіть сніговими лавинами. Коли вибирається розташування, потрібно скоротити погане вплив гір і погорбів, також може допомогти вживання різних гірських вершин або плато, щоб розташувати метеорологічний радіолокатор. Завданням вирішення сформованих труднощів - це розташування кількох метеорологічних радіолокаторів, тобто. організація радіолокаційної системи спостереження

середовищем оточення. Труднощі, що створюється при експлуатації МРЛ для знаходження метеоцелі згодом найважча. Складність полягає у високих горах, які закривають собою огляд, цим закривають і промінь радара та метеоцілі. Знаходження вигідного розташування радарів має враховувати кілька важливих вимог, й у перебування найкращого розташування метеорологічних радарів практикується нелегкий порядок дій. Методологічний підхід із застосуванням ГІС застосовується для знаходження вигідного розташування для метеорологічних радарів, які в свою чергу застосовуються для знаходження гідрометеорологічних цілей.

За прорисовку і рендерінг тривимірної моделі у САПР BRL-CAD, напряду відповідає компонента Mged. Вона є основною ланкою при моделюванні 3Dсистем. Наприклад Mgedзабезпечує таку важливу функцію, як трасування промінів, тобто наповнення фігури об'ємом і кольором. Цей процес досягається завдяки тому, що є можливість навчити САПР, використовувати різноманітні формули для трасування, виходячи із цілі(1.3).

$$P = O + t(V * 3 - o) \quad (1.3)$$

Перш за все потрібно розібратис основнепараметричне рівняння. В цій формулі P—це будь-яка точка промінню, а O—об'єкт через який проходить промінь.

Наступне рівняння напряду пов'язано з попереднім і відповідає на питання на яких саме координатах, промінь проходить скрізь об'єкт моделювання (1.4).

$$\{x, y\} = \frac{-k\sqrt{k^2 - 4cb}}{k} \quad (1.4)$$

Де x,y—координати пересічення промінню з об'єктом, а k, c, b— це другорядні параметри, необхідні для отримання координат. Виходячи з

останньої формули, було написано інтегровану частину коду, на строго-типізованій мові програмування Java, яка за допомоги методу `traceRayExecutor` отримує точки пересічення промінню з кожним об'єктом на своєму путі, і в залежності від результату повертає колір промінню:

```
public static String traceRayExecutor
(double x, double y, double tMin, double tMax) {
double cT;
Sphere cSphere = null;
for (Sphere sphere : spheres) {
Coordinate coords = sphereRayTracing(x, y, sphere);
if (tMin <= coords.coord1 && coords.coord1 <= tMax &&
coords.coord1 < tMin) {
cT = coords.coord1;
cSphere = sphere;
}
if (tMin <= coords.coord2 && coords.coord2 <= tMax &&
coords.coord2 < tMin) {
cT = coords.coord2;
cSphere = sphere;
}
}
if (cSphere == null) {
return BACKGROUND_COLOR;
} else {
return cSphere.color;
}
}
```

ВИСНОВКИ

Існує безліч переваг, пов'язаних із використанням комерційних пакетів у дослідженнях вразливості. Центр аналізу інформації про вразливість, визначив деякі найпоширеніші переваги у своєму сучасному звіті щодо геометричного моделювання.

Скорочення часу та зусиль на моделювання –виробники часто витрачають сотні годин на побудову детальних моделей САПР для впорядкування процесів проектування, виробництва та складання (наприклад, за допомогою комп'ютерного цифрового обладнання управління).

Збільшення фінансування та підтримки –оскільки комерційні пакети повинні підтримувати більшу базу користувачів, щоб залишатися конкурентоспроможними на відкритому ринку, найпопулярніші пакети зазвичай мають достатнє фінансування та персонал для постійного їх розвитку, підтримки та вдосконалення.

Сумісність зі стандартизованими форматами –більшість комерційних пакетів мають пряму або непряму (через сторонніх постачальників) можливість перетворення на стандартні або посередницькі формати геометрії, роблячи пакети сумісними (принаймні, до певної міри) з широким спектром інших форматів САПР.

Підтримка сторонніх надбудов –великі комерційні пакети, як правило, пропонують різноманітні плагіни для інших пакетів / утиліт.

Процес моделювання можна сприймати приблизно так . Перш ніж щось будувати, модельєр повинен взяти на себе роль архітектора та викласти логічний спосіб розбити потенційно складний об'єкт на менші, більш керовані частини. Крім того, цей крок часто може виявити логічний порядок будівництва (наприклад, гіпсокартон не встановлюється, поки не прокладена електропроводка всередині стін), а також виявити важливі взаємозв'язки між

частинами. У наведеному нижче списку подано кілька порад щодо досягнення хорошої структуризації моделей:

По-перше, використовуйте підхід зверху вниз. Це гарна ідея проектувати структуру, використовуючи підхід зверху вниз, починаючи з найбільших, найбільш охоплюючих або найбільш функціонально значущих частин / систем і працюючи звідти. Знову ж таки, місія моделі тут дуже важлива, якщо броньований танк моделюється для балістичного аналізу, модель мабуть, має бути побудована таким чином, щоб усі частини, підключені до башти, були згруповані разом і, отже, можуть рухатися разом при обертанні башти;

По-друге, скористайтеся встановленими організаційними конвенціями: розумно дотримуватися будь-яких традиційних або широко використовуваних конвенцій, які можуть бути доступні, скажімо, в посібнику власника або оператора, якщо механік або користувач зазвичай очікує від конкретного компонента бути частиною системи підвіски, тоді модельєру доцільно структурувати свою модель відповідно, якщо немає вагомих причин робити інакше;

По-третє, використовуйте хороші практики іменування тісно пов'язана з ідеєю використання хороших організаційних конвенцій для структурування геометрії - ідея використання хороших конвенцій іменування для іменування геометрії, хоча іменування може здатися тривіальною справою, справа в тому, що не завжди легко встановити заголовки та схеми, які інтуїтивно зрозумілі, надійні та корисні, щоб допомогти кінцевому користувачеві дізнатися, де він перебуває в потенційно складній моделі.

В процесі виконання магістерської роботи, було вирішено низку практичних задач, що ставилися безпосередньо перед виконанням роботи:

- проведено повний аналіз САПР BRL-CAD, та здобуто навички по роботі з усіма модулями програмного забезпечення для подальшого моделювання регіону;

- побудовано UML діаграми, які описують архітектуру дизайну розробки графічної моделі регіону;
- Створено тривимірну графічну модель рельєфу за допомогою САПР BRL-CAD, для облегшення подальших робіт по удосконаленню способів і засобів одержання даних про атмосферу радіофізичними методами;

Практичне значення отриманих результатів полягає в тому, що було виявлено нову, якісну, потужну САПР систему, за допомогою якої можливо моделювати великі моделі, а також масштабувати їх, на прикладі створення моделі рельєфу регіону

Також вдосконалено процес розробки цільової моделі, за допомоги створених інструкцій по міграціям між САПР системами у BRL-CAD. Оскільки більшість комерційних систем САПР мають можливість створювати мозаїчні наближення своїх об'єктів BREP, деякі перетворювачі в BRL-CAD використовують ці можливості. Крім того, згаданий раніше формат STL, який представляє тверді об'єкти повністю із трикутниками, пропонує спільноті універсальний спосіб перетворення геометрії BRL-CAD у майже будь-яку комерційну систему САПР.

Розроблено інструкції із застосування необхідного програмного забезпечення, що доповнені коментарями, відносно роботи спроектованого рельєфу регіону засобами відкритої кросплатформеної САПР BRL-CAD.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Великодний С. С., Тимофєєва 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.
2. Великодний С. С. Методологические основы реинжиниринга систем автоматизированного проектирования. *Междун. науч. журн. «Управляющие системы и машины»*. 2014. № 2. С. 39–43.
3. BRL-CAD Tutorial Series: Volume II – Introduction to MGED (загол. з екрану). URL:
https://brlcad.org/w/images/c/cf/Introduction_to_MGED.pdf (дата звернення: 11.10.2020).
4. Velykodniy S. S. Analysis and synthesis of the results of complex experimental research on reengineering of open CAD systems. *Applied Aspects of Information Technology*. 2019. Vol. 2. No 3. P. 186–205. (кат. «Б») DOI: 10.15276/aait.03.2019.2.
5. Velykodniy S. Reengineering of open software system of 3D modeling BRL-CAD. *Innovative Technologies and Scientific Solutions for Industries*. 2019. No 3(9), P. 62–71. (кат. «Б») <https://doi.org/10.30837/2522-9818.2019.9.062>.
6. Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання структурної частини. *Вісник Кременчуцького національного університету ім. Михайла Остроградського*. 2019. Вип. 3 (116). С. 130–139. (кат. «Б») DOI:10.30929/1995-0519.2019.3.130-139
7. Грунина Е.В., Калачёв О.Н. Методика проектирования в CAD/CAM *Cimatron УП для гравирования профиля на юбилейной медали*. Ярославль: ЯГТУ, 2008. 58 с.

8. Великодний С. С. Метод представлення оцінки реінжинірингу програмних систем за допомогою проектних коефіцієнтів. Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 1 (7). С.34–42. (кат. «Б») DOI: <https://doi.org/10.30837/2522-9818.2019.7.034>.
9. Великодний С. С. Ідеалізовані моделі реінжинірингу програмних систем. Радіоелектроніка, інформатика, управління. 2019. № 1. С. 150–156. DOI: 10.15588/1607-3274-2019-1-14.
10. Трухнин Н.М., Филиппович К.В. Макрорасширения в GrafCAM – путь к наращиванию функциональности. Пермь: ООО «Евразия Лимитед», 2006. 72 с.
11. Филиппович К.В., Попович И. ToolStore – среда для ведения библиотеки инструментов в верификаторе CNC-Verify системы САПР-ЧПУ/2005. Пермь: ООО «Евразия Лимитед», 2005. 36 с
12. Дискретная математика. Часть II. Математическая логика, : [Учеб. пособие.]: Э.Р. Зарипова, М.Г. Кокотчикова, Л.А. Севастьянов. М.: РУДН, 2013. 116 с.
13. Редькин Н.П. Дискретная математика. М.: ФИЗМАТЛИТ, 009. 264 с. ISBN 978-5-9221-1093-8..
14. Богге У., Богге М., UML. Rational Rose: Секреты эффективного проектирования сопровод. объектно-ориентиров. прилож. Москва: «Лори», 2000. 570 с.
15. A.J.Medland, P.Burnett CAD/CAM in practice: a manager's guide to understanding and using CAD/CAM systems. Kogan Page, 2003. 673 p.
16. Мацяшек Л. А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML. Москва: Вильямс, 2002. 432 с.
17. Калачев О. Н. Диалоговая система KON7 для расчета размерных цепей при механообработке. Ярославль: ЯГТУ, 2005. 93 с.

18. Филиппович К.В. Некоторые аспекты настройки пользовательских предпочтений в САПР-ЧПУ/2005. Пермь: ООО «Евразия Лимитед», 2005. 42 с.
19. Schultz D.F. What's Wrong With Postprocessors? Numerical Control Compute Sciences. 1998. №1. P. 11–15.
20. Computer-aided design system Parametric Technology Corporation (PTC), Pro / E (загол. з екрану). URL: <http://www.ptc.com> (PTC, 2003) (дата звернення: 23.04.2021).
21. ASCII encoding table (ASCII) (загол. з екрану). URL: <https://ru.wikipedia.org/wiki/ASCII> (дата звернення: 23.04.2021).
22. Компьютерная графика. Математические основы. Версия 1.0 [Электронный ресурс] : электрон. учеб. пособие / Н. В. Соснин. – Электрон. дан. (4 Мб). Красноярск : ИПК СФУ, 2008.
23. Чмырь И. А., Ус М. Ф. Моделирование систем в среде UML (Unified Modeling Language): конспект лекций. Одесса: ОГАХ, 2001. 70 с.
24. Леоненков А.В. Самоучитель UML. Санкт-Петербург: БХВ-Петербург, 2002. 304 с.
25. Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник. Санкт-Петербург: Питер, 2002. 656 с.
26. Фаулер М., Скотт К. UML. Основы. Санкт-Петербург: Символ-Плюс, 2002. 192 с.
27. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя (Сер. программистов). Москва: ДМК, 2000. 432 с.

ДОДАТОК А. СКРИПТ З КОМПОНЕНТИ MGED ПРИЗНАЧЕНИЙ ДЛЯ МОДЕЛЮВАННЯ ТРИВИМІРНОЇ МОДЕЛІ ГРАФІЧНОЇ ОБЛАСТІ.

```

import java.io.FileNotFoundException;
import java.io.RandomAccessFile;
import java.util.Map;
import java.util.HashMap;
import java.io.IOException;
import java.io.EOFException;
import numerics.Tolerance;
import java.util.BitSet;

public class BrlcadDb
{
/** Name of the BRL-CAD db file */
private String dbFileName;
/** The title of this BRL-CAD db */
private String title;
/** The RandomAccessFile object associated with this BRL-CAD
db */
private RandomAccessFile dbInput;
/** Directory of BRL-CAD object names and their offset into
the BRL-CAD db */
private Map < String, Long > directory;
/** Magic number (byte) at the start of the BRL-CAD db file */
public static final byte DB5HDR_MAGIC1 = 'v';
/** Magic number (byte) at the end of each object in the BRL-
CAD database */
public static final byte DB5HDR_MAGIC2 = '5';
/** default tolerance */
public static final double DEFAULT_TOL_DIST = 0.005;
public static final double DEFAULT_TOL_PERP = 1.0e-6;
public static Tolerance tolerance = new Tolerance(
DEFAULT_TOL_DIST, DEFAULT_TOL_PERP );
/** minimum number of faces to trigger use of pieces */
public static int BOT_MINFACES = 32;
/** number of triangles per Bot piece */
public static int BOT_FACES_PER_PIECE = 4;
/**
* Constructor
*
* @param dbFileName Name of the BRL-CAD DB file
*
* @exception FileNotFoundException
* @exception IOException
* @exception DbException
*
*/
public BrlcadDb ( String dbFileName ) throws FileNotFoundEx-
ception, IOException, DbException
{
this.dbFileName = dbFileName;
this.dbInput = new RandomAccessFile( this.dbFileName, "r" );
this.scan();
}
/**
* Sets Tolerance

```

```

*
* @param    Tolerance    a Tolerance
*/
public static void setTolerance(Tolerance tolerance)
{
    BrlcadDb.tolerance = tolerance;
}
/**
* Returns Tolerance
*
* @return    a Tolerance
*/
public static Tolerance getTolerance()
{
    return tolerance;
}
/**
* Get the RandomAccessFile object asociated with this BRL-CAD
database
*
* @return    a RandomAccessFile
*
*/
public RandomAccessFile getRAM()
{
    return this.dbInput;
}
/**
* Scan the BRL-CAD database, constructing the directory and
extracting the title
*
* @exception    IOException
* @exception    DbException
*
*/
private void scan() throws IOException, DbException
{
    // this will hold the 8 byte header at the start of the file
    byte fileHeader[] = new byte[8];
    // create a new directory
    this.directory = new HashMap < String, Long > ();
    try
    {
        // make sure we start at the start
        this.dbInput.seek( 0L );
        // read the file header
        this.dbInput.readFully( fileHeader );
        // verify the file header
        if( !this.fileHeaderIsValid( fileHeader ) )
        {
            throw new IOException( "Invalid file header" );
        }
        // scan the rest of the file by importing DbExternal objects
        Long offset;
        while( true )
        {
            offset = this.dbInput.getFilePointer();
            DbExternal dbExt = new DbExternal( this, offset );
            // if this object has a name, add it to the directory
            String name = dbExt.getName();

```

```

if( name != null )
{
if( this.directory.get( name ) != null )
{
System.err.println( "Duplicate name (" + name + ") ignored" );
}
else
{
System.out.println( name + ": " + offset );
this.directory.put( name, offset );
}
}
}
catch( EOFException e )
{
// we reached the end of the file
// get the title from this database
try
{
DBObject global = this.getInternal( "_GLOBAL" );
this.title = global.getAttribute( "title" );
}
catch( Exception e1 )
{
throw new DbException( "Db has no GLOBAL object " + e1);
}
System.out.println( "Completed scan, " +
this.directory.size() +
" objects" +
", title = " + this.title);
return;
}
}
/**
 * Get the DbObject that has the specified name
 *
 * @param name The name of the desired object
 *
 * @return a DbObject
 *
 * @exception IOException
 * @exception DbException
 * @exception DbNameNotFoundException
 *
 */
public DbObject getInternal( String name ) throws IOException,
DbException, DbNameNotFoundException
{
// Lookup this name in the directory
Long offset = this.directory.get( name );
if( offset == null )
{
// no such object
throw new DbNameNotFoundException( "Error: " + name + " not
found" );
}
// Read the object in external form
DbExternal dbExt = new DbExternal( this, offset );
// Import this object into its internal form (this will usual-

```

```

ly be a subclass of DbObject)
DbObject dbObj = this.importObj( dbExt );
return dbObj;
}
/**
 * Import a DbExternal object into its internal form
 * @param    dbExt          a DbExternal object
 * @return   a DbObject or, more likely, a subclass of DbObject
 */
private DbObject importObj(DbExternal dbExt) throws DbException
{
switch( dbExt.getMajorType() )
{
case 0:
throw new DbException( "Illegal major type number (0)" );
case 1:
switch( dbExt.getMinorType() )
{
case Ellipsoid.minorType:
return new Ellipsoid( dbExt );
case Arb8.minorType:
return new Arb8( dbExt );
case Bot.minorType:
return new Bot( dbExt );
case Combination.minorType:
return new Combination( dbExt );
default:
throw new DbException( "Unrecognized minor type (" +
dbExt.getMinorType() + ")" );
}
case 2:
return new DbObject( dbExt );
default:
throw new DbException( "Unrecognized major type (" +
dbExt.getMajorType() + ")" );
}
}
}
/**
 * Get the description of an object with the specified name
 * @param    name          a String
 * @return   a description of the specified object
 * @exception DbException
 * @exception IOException
 * @exception DbNameNotFoundException
 */
public String describe( String name ) throws DbException, IOException, DbNameNotFoundException
{
DbObject obj = this.getInternal( name );
return obj.toString();
}
}
/**

```

```

* Read an unsigned integral number from the database using the
specified number of bytes
*
* @param    numBytes          the number of bytes to read
(must be 1, 2, 4, or 8)
*
* @return   a long
*
* @exception IOException
*
*/
public long getUnsignedLong( int numBytes ) throws IOException
{
    long longNum;
    switch( numBytes )
    {
    case 1:
        return (long)this.dbInput.readUnsignedByte();
    case 2:
        return (long)this.dbInput.readUnsignedShort();
    case 4:
        byte[] bytes = new byte[4];
        this.dbInput.readFully( bytes );
        longNum = (long)(bytes[0] & 0xff);
        for( int i=1 ; i<4 ; i++ )
        {
            longNum = longNum << 8;
            longNum |= (long)(bytes[i] & 0xff);
        }
        return longNum;
    case 8:
        longNum = this.dbInput.readLong();
        if( longNum < 0 )
        {
            throw new IOException( "Database contains an unsigned long
that we cannot read in Java!!");
        }
        return longNum;
    default:
        throw new IOException( "Br1cadDb.getLong(): Illegal length ("
+ numBytes +
")" );
    }
}
/**
* Read a signed integral number from the database using the
specified number of bytes
*
* @param    numBytes          The number of bytes to read
*
* @return   a long
*
* @exception IOException
*
*/
public long getLong( int numBytes ) throws IOException
{
    switch( numBytes )
    {
    case 1:

```

```

return (long)this.dbInput.readByte();
case 2:
return (long)this.dbInput.readShort();
case 4:
return (long)this.dbInput.readInt();
case 8:
return this.dbInput.readLong();
default:
throw new IOException( "BrlcadDb.getLong(): Illegal length ("
+ numBytes +
)" " );
}
}
/**
 * Extract a long value using bytes from the provided input ar-
ray
 *
 * @param bytes The array containing the bytes
 * @param pointer The location in the above array to start
converting
 * @param length The number of bytes to convert
 * @return A long
 */
public static long getLong( byte[] bytes, int pointer, int
length )
{
long longBytes = (long)(bytes[pointer] & 0xff);
for( int i=1 ; i<length ; i++ )
{
longBytes = longBytes << 8;
longBytes |= ((long)bytes[pointer+i] & 0xff);
}
return longBytes;
}
/**
 * Extract a double value using 8 bytes from the provided input
array starting with the
 * array element at index "pointer"
 *
 * @param bytes a byte[]
 * @param pointer the starting index in the
above array
 *
 * @return a double
 *
 */
public static double getDouble( byte[] bytes, int pointer )
{
long longBytes = (long)(bytes[pointer] & 0xff);
for( int i=1 ; i<8 ; i++ )
{
longBytes = longBytes << 8;
longBytes |= ((long)bytes[pointer+i] & 0xff);
}
return Double.longBitsToDouble( longBytes );
}
/**
 * Extract a BitSet from an input array of bytes. The end of
the BitSet is
 * the first 0 value byte after the starting point.

```

```

* @param bytes The array of bytes
* @param pointer The starting point of the BitSet in the
above array
* @return A BitSet
*/
public static BitSet getBitSet( byte[] bytes, int pointer )
{
    BitSet bitset;
    // find start of bit vector
    while( Character.iswhitespace(bytes[pointer]) )
    {
        pointer++;
    }
    // count number of hex digits in bit vector
    int index = pointer;
    while( bytes[index] != 0 )
    {
        index++;
    }
    int length = index - pointer;
    if( length < 2 || (length % 2) != 0 )
    {
        return null;
    }
    bitset = new BitSet( length * 4 );
    // set the bits
    int bitNum = length*4 - 1;
    while( bytes[pointer] != 0 )
    {
        byte mask = (byte)0x8;
        byte b = Byte.decode( "0x" + (char)bytes[pointer] );
        for( int i=0 ; i<4 ; i++ )
        {
            if( (b & mask) != 0 )
            {
                bitset.set( bitNum, true );
            }
            mask = (byte)(mask >> 1);
            bitNum--;
        }
        pointer++;
    }
    return bitset;
}
/**
* Method fileHeaderIsValid
*
* @param fileHeader a byte[]
*
* @return a boolean
*/
}

```