

ПРИКЛАД ВИБОРУ АРХІТЕКТУРНИХ РІШЕНЬ ПІДСИСТЕМИ РОБОТИ З ВІЗУАЛЬНИМИ МОДЕЛЯМИ

К.О. Холод, С.В. Кузнецов

Національний авіаційний університет, Інститут комп'ютерних технологій

У статті розглянуті підходи до вибору архітектурних рішень підсистеми роботи з візуальними моделями на прикладі генератору каналу комп'ютерного генератору зображення імітатора візуальної обстановки комплексного авіаційного тренажеру.

Для забезпечення більшої безпеки польотів, а також перекваліфікації пілотів, ІКАО було передбачено поряд з виробництвом повітряного судна розробляти і його комплексний авіаційний тренажер, який забезпечує можливість об'єктивної оцінки якості підготовки льотчиків і екіпажа у всіх режимах, у тому числі попередньому польотові, а також у режимах позаштатних ситуацій і при виконанні бойових задач. Він являє собою відтворений у реальному масштабі макет кабіни повітряного судна конкретного типу або конструкції, конкретної моделі або серії. Включає: комплект обладнання та програмне забезпечення, необхідне для відтворення умов експлуатації повітряного судна на землі та в повітрі; систему візуалізації, що відображає позакабінний простір; систему імітації акселераційних впливів.

Комп'ютерний генератор зображення (КГЗ) – компонент імітатора візуальної обстановки, що призначений для побудови та відображення на пристрій індикації візуальної обстановки поза кабіною літака у реальному часі динаміки польоту. Сучасний КГЗ складається з імітатору та серверу повітряного руху, серверу баз даних середовища (СБДО відповідає за моделювання та передачу генераторам каналів об'єктивної інформації про стан атмосфери, земної поверхні, аеродромах та інших літаків в повітрі на землі) та одного чи декількох генераторів каналів, що відображують на пристрої індикації певну частину простору поза кабіною літака. Усі імітатори тренажеру обмінюються інформацією та командами через виделену.

В багатоканальних КГЗ задачі окремого генератору каналу (ГК) істотно відрізняються від одно канальних КГЗ в першу чергу обмеженням спектру задач, які вони вирішують. До цих задач відносяться: отримання та інтерпретація потоку керуючих команд та даних опису тривимірних просторових моделей, відповідна

модифікація та підготовка передачі до графічного конвеєру графу сцени, підсистема виводу до графічного конвеєра.

Сформульовані наступні загальні вимоги до архітектури:

- мінімальна складність та простота супроводу;
- слабке сполучення, що забезпечує можливість підміни реалізацій інтерпретаторів графа моделі;
- можливість повторного використання;
- кроссплатформенність;
- стратифікація.

Також виділені наступні специфічні вимоги: параметризація керування генерацією й інтерпретацією моделі (це необхідно, наприклад, для моделей небесної сфери й лінії обрїю); єдине програмне ядро для генерації моделі і її наступної інтерпретації.

Проектування архітектури здійснюється на основі сформульованої системи вимог з використанням системного підходу, дозволяє вирішити цілий ряд завдань, головним з яких є [1]:

- декомпозиція на агрегати й модулі;
- виявлення рівнів ієрархії;
- дослідження й виявлення системоутворюючих властивостей компонентів та їх взаємні зв'язки на різних рівнях;
- вибір принципів побудови різних конфігурацій залежно від цільової функції.

Реалізовані рішення етапу проектування:

- декомпозиція на основі сформульованої системи вимог;
- виявлення на основі формальних вимог, забезпечуваного якості моделювання й ступені формалізації опису існуючих моделей даних й алгоритмів які доцільно повторно використати, а не розробляти заново;
- вибір методу програмної реалізації модуля (об'єктно-орієнтований підхід, структурне програмування) на основі ступеня формалізації відповідної частини предметної області;
- застосування паттернів проектування для поділу логіки керування й інтерпретації моделі.

Умовно усі компоненти за функціональністю можна розподілити між 6 рівнями: сервісу операційної системи; системний; моделей; інтерпретації моделей; програмний; комунікаційний.

На кожному з рівнів компоненти за призначенням розподілені по таких ланках: ядро, модулі та допоміжні.

На рис. 1. представлена схема розподілу КГЗ на компоненти та зв'язки між ними.

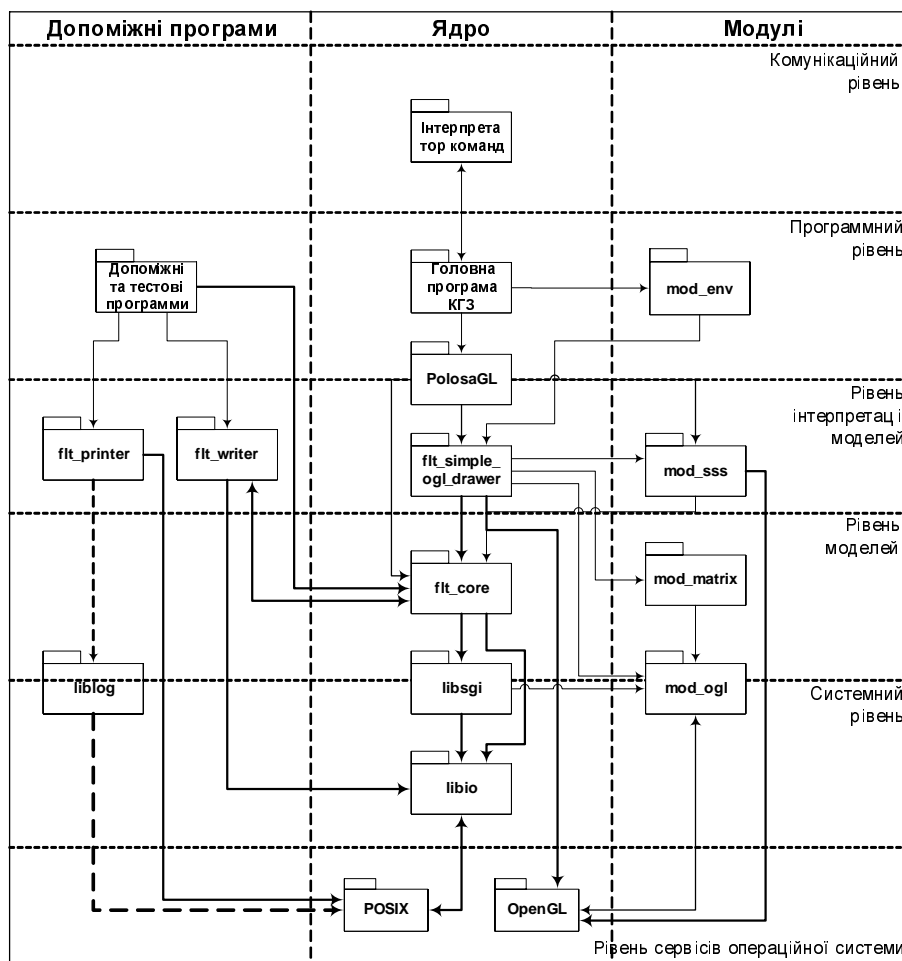


Рис. 1. Архітектура каналу КГЗ

До компонентів рівня сервісу операційної системи компоненти, що відповідають за взаємодію КГЗ з апаратними пристроями через інтерфейси операційної системи. Компоненти системного рівня використовують можливості компонентів нижнього рівня для надання вищим рівням абстрактних інтерфейсів.

mod_OGL – модуль, що накопичує частину параметрів графічного конвеєру, та забезпечує ефективну роботу з ними. **libIO** – бібліотека вводу/виводу, забезпечує платформи-незалежну роботу з бінарними файлами. **libSGI** – бібліотека роботи з текстурами у форматі Silicon Graphics. Формат підтримує стиснення даних та можливість організації багаторівневих текстур. **libLog** – допоміжна бібліотека для організації гнучкого ведення протоколу роботи програм.

Компоненти представлення моделей. Основним форматом для представлення полігональних моделей обраний формат OpenFlight (<http://multigen.com>), що є індустріальним стандартом для промислових систем візуалізації. OpenFlight підтримує: змінні рівні деталізації, обмеження ступенів свободи, звуки, посилання (внутрішні та

зовнішні), реплікацію, анімацію, обмежуючи об'єми, освітлення, вогні та стрічки вогнів.

flt_core – бібліотека, що містить класи які представляють типи вузлів бази даних OpenFlight. Для представлення моделей у форматі OpenFlight використовується паттерн Композит[3]. До складу бібліотеки входить пул-синглетон, який накопичує завантажені моделі та дозволяє керувати ними.

mod_matrix – модуль, що забезпечує роботу з матрицями перетворень. Виконує наступні функції: завантажує початкові матриці з графічного конвеєра, накопичує стеки видових, текстурних, та проєкційних матриць перетворень, виконує над матрицями операції переносу, повороту, масштабу та множення, перераховує координати для заданої точки в контексті поточного стану стеку матриць перетворень.

Компоненти рівня інтерпретації моделей. Для обробки графів моделей використовується паттерн Відвідувач [3]. Класи-відвідувачі дозволяють відділити операції на моделями та самих моделей. Виділені два типи класів-відвідувачів: основні та допоміжні. Основні класи-відвідувачі відповідають за відображення моделей. Допоміжні – за виконання над графами-моделями різноманітних операцій по збору та аналізу інформації.

flt_simple_ogl_drawer – бібліотека, що містить основний клас-відвідувач, який за допомогою бібліотеки OpenGL відображує модель.

Знаючи за допомогою інформації з модуля mod_env поточне положення у просторі, відсікає відображення невидимих ділянок моделі та обирає необхідний рівень деталізації для частин графу моделі, які мають альтернативні представлення в залежності від відстані до них. Клас-відвідувач збирає і накопичує інформацію про вогні для подальшого їх відображення у mod_sss. Про кожний з вогнів зберігається: координати у світовій системі координат, видові координати, координати в одиничному кубі, колір на передньому плані, колір на задньому плані. Для розрахунку координат вогнів у видовій системі координат та в одиничному кубі використовується інформація, що накопичується в модулі mod_matrix.

flt_writer – бібліотека, що містить клас-відвідувач, який зберігає задану FLT-модель на диску.

mod_sss – модуль, що відповідає за відображення вогнів (наприклад, вогнів світлосигнальних систем).

Компоненти програмного рівня.

PolosaGL – бібліотека, що містить фасад[3] для доступу до моделей OpenFlight та їх відображенням за допомогою класу-

відвідувача з бібліотеки `flt_simple_ogl_drawer`. Використовується головною програмою КГЗ.

Головна програма КГЗ встановлює зв'язок з графічним прискорювачем, завантажує та починає роботу модулів, організовує середовище для зв'язку с інтерпретатором команд, організує відображення сцени.

Допоміжні програми – програми підтримки процесу створення сцен. Прикладом таких програм є генератори ландшафтів та місцевості, параметричних аеродромів, параметричних світлосигнальних систем, рівнів деталізації моделей.

`mod_env` – модуль, що накопичує інформацію про стан середовища моделювання. Мінімальний набір складає інформація про положення глядача в просторі та поточний час. Інформація оновлюється з головної програми КГЗ.

Компоненти комунікаційного рівня. Інтерпретатор команд – компонент, що приймає із мережі команди та виконує їх над внутрішніми об'єктами КГЗ. Також на ньому можуть бути програми-адаптери для протоколів Глорія, DIS, HLA та інших. Це дозволяє адаптувати КГЗ для сумісної роботи з різноманітними моделюючими комплексами та використовувати його в різних прикладних областях.

Для реалізації вимог що поставлені до системи візуалізації (в основному до пілотажних тренажерів) необхідні дослідження та розробка моделей в таких областях як моделювання поверхні землі, атмосферних явищ, лінії горизонту, вогнів та вогнів світлосигнальних систем, методах корекції перспективних викривлень.

Кожна з приведених областей погано підлягає формалізації. Для того щоб дослідники мали змогу ефективно проводити експерименти з моделями в цих областях цикл апробації моделей повинен бути зменшений до мінімуму. Відкрита архітектура КГЗ сприяє цьому найкращим чином.

Список літератури

1. Моржов В.И. Научно-технические методы построения моделирующей среды, обеспечивающей потребные эксплуатационно-технические характеристики авиационного тренажера. Диссертация на соискание ученой степени доктора технических наук. К: КИИГА, 1990 год, 352 стр.

2. Alfred T. Lee, Flight Simulation: Virtual Environments In Aviation. Ashgate Publishing, 2005, 134 pages.

3. Гамма Э., Хелм Р., Джонсон Р., Влссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.: ил. (Серия «Библиотека программиста»)

Дата надходження до редколегії: 15 травня 2007 року