

В.В. Спиринцев, О.Є. Баланенко

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ API ДЛЯ РОЗШИРЕННЯ БІБЛІОТЕК В СИСТЕМАХ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ

Анотація. Запропоновано універсальне програмне забезпечення на базі мови програмування C# з використанням API для розширення бібліотек в найбільш поширених системах автоматизованого проектування: КОМПАС, Autodesk Inventor, SolidWorks.

Ключові слова: автоматизоване проектування, API, C#, параметризація.

Постановка проблеми. Сучасний ринок програмного забезпечення автоматизації підготовки виробництва насичений найрізноманітнішими універсальними САПР, що здатні суттєво полегшити роботу проектувальника. Разом з тим, незважаючи на величезну кількість такого виду інструментальних засобів автоматизації інженерної діяльності, універсальні системи часто недостатньо ефективні для вирішення конкретного завдання користувача, за часту інформаційно несумісні.

Більшість застосовуваних у промисловості тривимірних САПР можуть бути використані як основа для побудови спеціалізованих САПР, які вирішують завдання розрахунку і проектування конкретного класу виробів. При цьому необхідно об'єднати розрахунковий модуль, який визначає розмірні та інші параметри проектного об'єкта з уже наявним в САПР тривимірним геометричним ядром. Цей процес здійснюється за допомогою API - інтерфейс прикладної програми. Набір таких інтерфейсів забезпечує взаємозв'язок між зовнішніми модулями прикладної програми і низькорівневими функціями ядра, а так само між компонентами ядра – різними бібліотеками, що істотно підвищує потенційні можливості застосування універсальних систем в специфічних предметних областях.

Аналіз останніх досліджень. Попередні дослідження [1, 2], були орієнтовані на розробку програмного забезпечення для конкретної CAD (Computer-Aided Design) системи із застосуванням мови програ-

мування Delphi. Цей факт істотно звужував область використання запропонованого програмного забезпечення. На більшості підприємств за часту встановлені різні CAD-системи, враховуючи специфіку виробництва та вирішуваних ними завдань. Для більш ефективної роботи проєктувальників та з метою скорочення часу на проєктування було б доцільно розробити універсальне програмне забезпечення на базі мови програмування C# з використанням API для розширення бібліотек в найбільш поширених системах автоматизованого проєктування: КОМПАС, Autodesk Inventor, SolidWorks.

Метою даної роботи є узагальнення та систематизація результатів досліджень по розробці програмного забезпечення для розширення бібліотек нових виробів в найбільш поширених системах автоматизованого проєктування при організації сучасного виробництва.

Основна частина. На рисунку 1 наведено загальну схему параметризації моделі.



Рисунок 1 - Загальна схема зміни конфігурації 3D моделі

Робота з процесами CAD-систем здійснюється за допомогою COM (Component Object Model - об'єктна модель компонентів) об'єктів - це технологічний стандарт від компанії Microsoft, призначений для створення програмного забезпечення на основі взаємодіючих компонентів, кожен з яких може використовуватися в багатьох програмах одночасно. Основним поняттям, яким оперує стандарт COM, є COM-компонент. Кожен COM-компонент має унікальний ідентифікатор (GUID) і може одночасно використовуватися багатьма програмами. Компонент взаємодіє з іншими програмами через COM-інтерфейси - набори абстрактних функцій і властивостей. Кожен

COM-компонент повинен, як мінімум, підтримувати стандартний інтерфейс «IUnknown», який надає базові засоби для роботи з компонентом. Інтерфейс «IUnknown» включає в себе три методи: QueryInterface, AddRef, Release.

Windows API надає базові функції, що дозволяють використовувати COM-компоненти. Бібліотеки MFC і, особливо, ATL / WTL надають більш гнучкі та зручні засоби для роботи з COM.

Загальний алгоритм роботи:

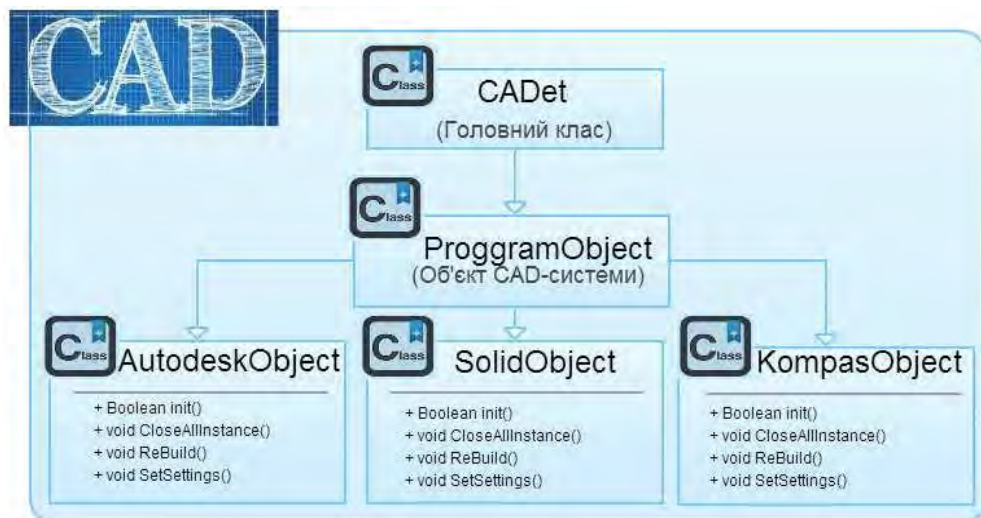
1. Створення COM об'єкта, використовуючи системний метод CreateInstance (), параметр - тип CAD системи, об'єкт якої створюється.

2. Відкриття файлу моделі, яка має бути піддана модифікації.

3. За допомогою API функцій відповідної CAD-системи, отримуємо список параметрів (записуємо їх у користувальницький інтерфейс, що виводить поточні значення цих параметрів).

4. Після зміни користувачем параметрів 3D моделі і натискання кнопки «Перебудувати», модель деталі перебудовується.

Ієрархія класів реалізованого програмного модуля «CADet» показана на рисунку 2.



Рисунку 2 - Ієрархія класів програмного модуля "CADet"

Для прикладу, розглянемо більш докладно процес розробки програмного забезпечення на базі API Autodesk Inventor.

Для отримання можливості працювати з API Autodesk Inventor, в проект необхідно додати посилання на бібліотеку - Interop.dll (за замовчуванням для 64-розрядної операційної системи перебуває в кореневій папці

«C:\Windows\Microsoft.NET\assembly\GAC_MSIL\Autodesk.Inventor.Interop\v4.0_19.0.0.0_d84147f8b4276564\autodesk.inventor.interop.dll», також її можна завантажити з офіційного сайту).

Перед тим як створити новий процес Autodesk Inventor, необхідно закрити всі поточні процеси Autodesk Inventor (з метою оптимізації роботи):

```
Process[] processes =
Process.GetProcessesByName("INVENTOR");
foreach (Process proces in processes)
{
proces.CloseMainWindow();
proces.Kill();
}
```

Далі створюємо новий процес Autodesk Inventor і наводимо тип COM об'єкта до типу Inventor.Application:

```
Type t = Type.GetTypeFromProgID("INVENTOR.Application");
v_invApp =
(Inventor.Application)Activator.CreateInstance(t);
```

Inventor.Application - посилання на об'єкт Autodesk Inventor, точка входу для використання API Autodesk Inventor.

Створюємо новий 3D документ типу Inventor_Document і відкриваємо 3D модель використовуючи метод Open об'єкта Inventor_Document.Documents:

```
Inventor._Document v_file;
v_file =
v_invApp.Documents.Open(@"F:\Детали\Корпус\Деталь.ipt");
```

При натисканні на кнопку «Перебудувати», викликається метод ReBuild (), який отримує посилання на активний документ - 3D документ типу:

```
dynamic otherdoc = v_invApp.ActiveDocument;
```

З об'єкта otherdoc отримуємо посилання на список змінних 3D документа типу oParameters і посилання на список параметрів поточної моделі:

```
Parameters oParameters =
otherdoc.ComponentDefinition.Parameters();
ModelParameters ModelParams =
oParameters.ModelParameters;
```

Далі, проходячи циклом з цієї колекції, змінюємо значення параметрів. Після зміни значень параметрів, необхідно перебудувати 3D документ:

```
v_file.Update();
```

Таким чином, за допомогою API Autodesk Inventor, програмний модуль змінює значення параметрів 3D моделі. При необхідності, можна зберегти поточний результат або ще раз змінити значення параметрів і знову перебудувати модель.

Інструкція користувача:

1. Створення параметризованого об'єкта.
2. Підключення до програмного середовища, в якому необхідно змінити об'єкт (Autodesk Inventor, SolidWorks, Компас 3D). Інтерфейс «підкаже» користувачу, коли з'єднання відбулося.
3. Завантаження об'єкта в програмний модуль, у вікно програмного модуля динамічно завантажаться параметри моделі.

Тепер користувач має змогу переглянути модель (рисунок 3).

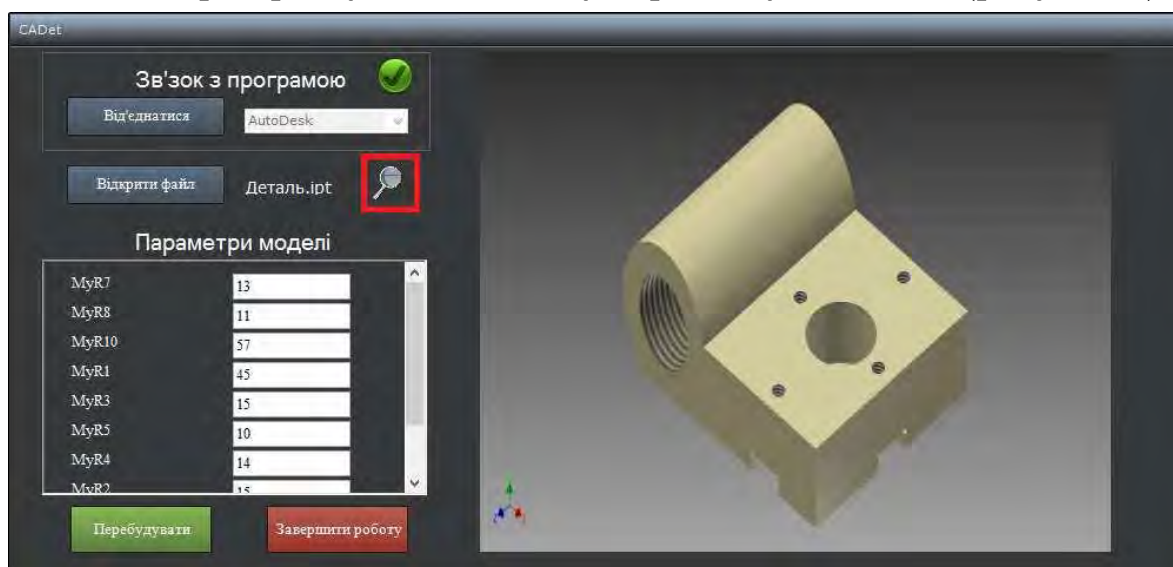


Рисунок 3 - "CADet" - користувацький інтерфейс модуля САПР

4. Зміна параметрів креслення.
5. Перебудова об'єкта.

Висновки. В роботі запропоновано універсальне програмне забезпечення на базі мови програмування C# з використанням API для розширення бібліотек в найбільш поширених системах автоматизованого проектування: КОМПАС, Autodesk Inventor, SolidWorks. Розроблене програмне забезпечення враховує потреби сучасного виробництва.

Подальший розвиток запропонованих досліджень можливо проводити в напрямку розширення функціоналу розробленого модулю та пошуку нових шляхів формування складних поверхонь виробів.

ЛІТЕРАТУРА

1. Спірінцев В.В. Використання АРІ- інтерфейсу КОМПАС в автоматизованому проектуванні / В.В.Спірінцев, С.В.Троян// Актуальні проблеми дисциплін природничо- наукової і професійно-практичної підготовки сучасних інженерів // Матеріали Всеукраїнської студентської науково-практичної конференції – Херсон, ХДАУ,2011.- С.61-64.
2. Спірінцев В.В. Розробка спеціалізованого програмного модуля для проектування типових деталей/ В.В.Спірінцев// Сучасні проблеми геометричного моделювання // Збірник праць XIV Міжнародної науково - практичної конференції. – Мелітополь, ТДАТУ 2012. – с. 103-107.