УДК 004.9+004.451.84
© Т. Лозинський[1], 2012

СИСТЕМНИЙ ІНТЕРФЕЙС ТРАНСЛЯТОРА СТРУКТУР ДАНИХ

Наведено структуру генератора програмного коду об'єктно-орієнтованої моделі для компілювання завдання технологічного процесу.

The structure of the generator code of object-oriented model for compiling task process have been described.

1. ВСТУП

Постановка завдання для автоматизованого технологічного процесу, зокрема в поліграфії, потребує виконання значної кількості операцій і затрат робочого часу фахівця. На сьогодні існує значна кількість середовищ проектування об'єктно-орієнтованих моделей, робота яких підтримується певними алгоритмами, що реалізують певний інформаційний процес, наприклад, MATLAB/Simulink, AutoCAD, Visio, AbstractAL, TermEdit, і т.д. [1-6].

Ці системи автоматизованих проектувальних робіт (САПР) забезпечують побудову та дослідження моделей.

2. ПОСТАНОВКА ЗАДАЧІ

Однак в описаних середовищах немає засобів генерування коду завдання виробничого процесу, реалізованого підготовленою моделлю. Тому з метою генерування коду завдання, доцільно розробити генератор коду, який би забезпечив створення лістінгу алгоритму виконання завдання мовою програмування високого рівня.

3. ВИБІР СЕРЕДОВИЩА ПРОГРАМУВАННЯ

C# це потужна об'єктно-орієнтована мова програмування, спроектована спеціально для застосування разом з середовищем .NET.

Основні можливості C# – повна підтримка класів і об'єктно-орієнтованого програмування, включаючи успадкування реалізації та інтерфейсів, віртуальні функції та перевантаження операцій.

Дизайн відповідає сучасним інструментам розробки. Погоджений набір базових типів. Повна підтримка бібліотек базових класів. NET [7-9].

[1]Українська академія друкарства

4. СИСТЕМНИЙ ІНТЕРФЕЙС ТРАНСЛЯТОРА СТРУКТУР ДАНИХ

Завдання, подане розширеною мовою розмітки, яке описує побудовану у середовищі автоматизованого проектування модель технологічного процесу, подається на опрацювання транслятору структур даних (ТСД), який сладається з аналізатора, модуля виявлення синтаксичних помилок (parser) та транслятора.

Аналізатор здійснює перевірку завдання на дійсність (присутність у каталозі заданого системою усіх файлів завдання) яке складається з файлів опису завдання та класів. Якщо файл не знайдено у каталозі, здійснюється з'єднання з базою даних (база програм) і пошук відбувається у базі програм [10]. Зчитані дані завдання (крім даних отриманих з бази програм) передаються модулю-parser для синтаксичної перевірки. Знайдені помилки автоматично поповнюють список помилок (ErrorList). Якщо аналіз завдання завершений, а список помилок заповнений, відображається діалогове вікно з детальним списком помилок в іншому випадку транслятор з перевірених даних завдання, генерує код об'єктно-орієнтованою мовою програмування C#, який записуються у файлову системо з розширенням (.cs). Також є можливість запису цього коду в базу програм.

За допомогою утиліти MSBuild відбувається побудова проекту відповідно до завдання. Проект і згенеровані файли (.cs) компілятором c# компілюються у збірку (проміжний код IL). Під час виконання у загальному середовищі виконання мов (CLR), оперативним (JIT) компілятором код IL компілюється у машинний код [2].

Схема зв'язків системного інтерфейса транслятора структур даних представлена (рис. 1).

5. ІЄРАРХІЯ КЛАСІВ СИСТЕМИ ГЕНЕРУВАННЯ КОДУ

Для даної системи було спроектовано класи об’єктно-орієнтованою мовою програмування C#. Класи аналізу завдання, опису системи(класів, методів, подій і т.д.), синтаксичного аналізу, словників, винятків, з'єднання з базою даних, трансляції, генерування проектів та компіляції за допомогою зовнішніх модулів.

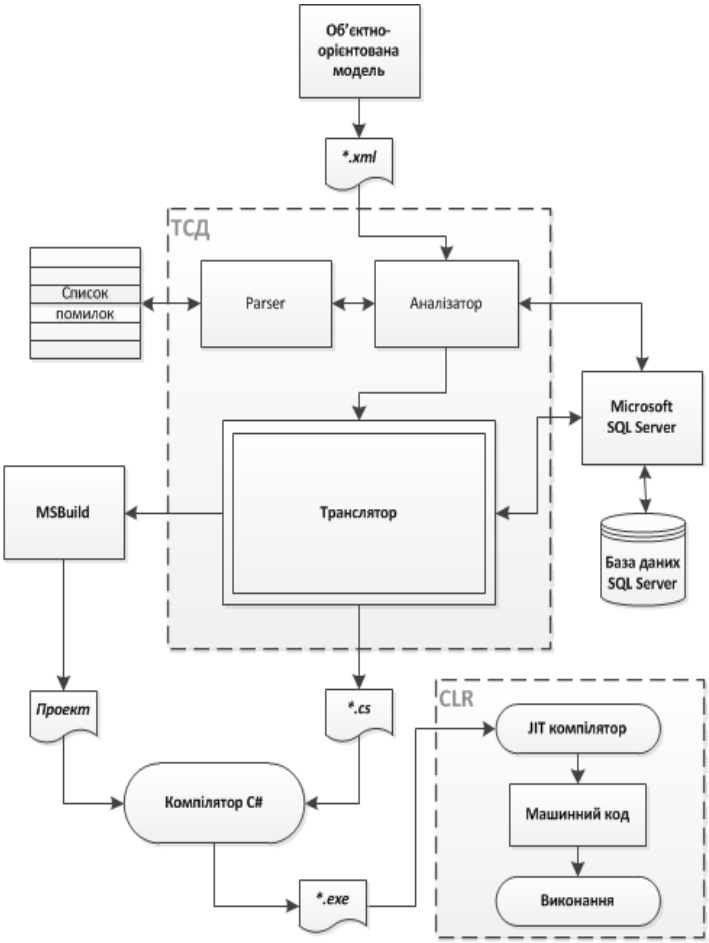

Рис. 1. Схема зв'язків системного інтерфейса транслятора структур даних

6. ВИСНОВОК

Таким чином, побудовавна модель для компілювання завдання може бути адаптована під довільний автоматизований технологічний процес, що дозволить уникнути помилок, притаманних людському чиннику та суттєво прискорить і здешевить отримання кінцевого результату.

1. www.mathworks.com 2. www.microsoft.com 3. autodesk.com 4. Овсяк В., Василюк А. Принцип побудови підсистеми редагування формул абстрактних алгоритмів //Комп'ютерні технології друкарства –Львів: УАД, – 2004. – №12. – с. 137-146. 5. Овсяк В., Овсяк О., Козелко М., Кулик С., Лозинський Т. Модель і програма комп'ютерної системи генерування програм з формул алгоритмів (ГЕПФА).: Львів, 2011. 6. Овсяк В.К., Лозинський Т.М. Декомпозиція системи аналізу XML-опису формул алгебри алгоритмів // Тези доповідей звітної науково-технічної конференції УАД за 2010 р.: Львів, 2011. 7. Нейгел, Кристиан, Ивьен, Билл, Глинн, Джей, Уотсон, Карли. C# 4.0 и платформа .NET 4 для профессионалов. : Пер. с англ. – М. : ООО "И.Д. Вильямс", 2011. – 1440 с. 8. Троелсен, Эндрю. Язык программирования C# 2010 и платформа .NET 4.0, 5-е изд. : Пер. с англ. – М. : ООО "И.Д. Вильямс", 2011. – 1392 с. 9. Т. Нортроп, М. Стэкер, С. Стэйн Разработка клиентских Windows-приложений на платформе Microsoft.Net Framework: Пер. с англ. – М.: Диалектика, 2011. – 624 с. 10. Роберт Виейра. Программирование баз данных Microsoft SQL Server 2008 для профессионалов.: Пер. с англ. – М.: ООО "И.Д. Вильямс", 2010. – 1072 с