

УДК 004.4'244

**СПОСІБ МУЛЬТИЛІНГВІСТИЧНОГО ПЕРЕКОДУВАННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ СКЛАДНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

Великодний С.С., Тимофєєва О.С.

*Коледж зв'язку та інформатизації Одеської національної академії зв'язку ім. О.С. Попова,
65045, Україна, м. Одеса, пр. Олександрівський, 6.
Одеський державний екологічний університет,
65016, Україна, м. Одеса, вул. Львівська, 15.
velykodniy@gmail.com*

**СПОСОБ МУЛЬТИЛІНГВИСТИЧЕСКОГО ПЕРЕКОДИРОВАНИЯ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СЛОЖНЫХ
ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ**

Великодний С.С., Тимофеева Е.С.

*Колледж связи и информатизации Одесской национальной академии связи им. А.С. Попова,
65045, Украина, г. Одесса, пр. Александровский, 6.
Одесский государственный экологический университет,
65016, Украина, г. Одесса, ул. Львовская, 15.
velykodniy@gmail.com*

**THE WAY OF MULTILINGUAL SOFTWARE TRANSCODING
FOR COMPLEX INFORMATION SYSTEMS AND TECHNOLOGIES**

Velykodniy S. S., Tymofieieva O. S.

*College of communication and informatization, O.S. Popov Odessa national academy of telecommunications,
6 Olexandrov's Av., Odessa, 65045, Ukraine.
Odessa state environmental university,
15 Lvivska St., Odessa, 65016, Ukraine.
velykodniy@gmail.com*

Анотація. У статті розглянуто новий спосіб перекодування програмного забезпечення складних інформаційних систем і технологій різного галузевого призначення. Відмінною особливістю розглянутого способу є можливість підтримки роботи понад десяти найпопулярніших мов програмування. При застосуванні способу вдається автоматизувати процес перекодування компонентів програмного забезпечення та за рахунок цього вивільнити робочий час програмістів від рутинного перепрограмування та зменшити вірогідність виникнення структурних помилок, що усладковуються від попередньої системи.

Ключові слова: інформаційні системи та технології, реінжиніринг, перекодування, компонент програмного забезпечення, CASE-засіб.

Аннотация. В статье рассмотрен новый способ перекодирования программного обеспечения сложных информационных систем и технологий, предназначенный для использования в различных отраслях. Отличительной особенностью данного способа является возможность поддержки работы более десяти самых популярных языков программирования. При применении способа удается автоматизировать процесс перекодирования компонентов программного обеспечения и за счет этого высвободить рабочее время программистов от рутинного перепрограммирования и уменьшить вероятность возникновения структурных ошибок, которые наследуются от предыдущей системы.

Ключевые слова: информационные системы и технологии, реинжиниринг, перекодирование, компонент программного обеспечения, CASE-средство.

Abstract. The article considers a new way of transcoding software for complex information systems and technologies intended for use in various industries. A distinctive feature of the presented method is the ability to support the work of more than ten most popular programming languages. When applying the method, it is possible to automate the process of transcoding software components, thereby freeing up the programmers' working time from routine reprogramming and reducing the likelihood of occurrence of structural errors that are inherited from the previous system.

Key words: information systems and technologies, reengineering, transcoding, software component, Computer-Aided Software Engineering (CASE-means).

Інформаційні системи та технології (ІСТ) застосовуються у різноманітних галузях життя й діяльності людини, причому подальший розвиток цивілізації зменшує сегмент життя людини, що не охоплено ІСТ. Експлуатація чи, звичніше сказати, використання ІСТ у кожній окремій галузі (управління виробництвом, транспортом, побудова телекомунікацій, освіта тощо) має свої принципи відмінності.

Спільної рисою для усіх ІСТ залишається те, що під впливом часу та інших невід'ємних факторів інформатизації (оновлення: операційних систем, мов програмування, принципів дії розподілених систем обробки даних, що особливо важливо у сфері телекомунікацій) відбувається еволюційне старіння програмного забезпечення (ПЗ) ІСТ. Така тенденція призводить до погіршення швидкісних, інформаційно-комунікаційних, графічних, часових та інших характеристик аж до повної відмови ІСТ.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими або практичними задачами. Створення складних ІСТ (СІСТ) – складна і трудомістка робота, виконання якої під силу тільки великому висококваліфікованому і злагодженому колективу розробників, оскільки тільки сам процес створення СІСТ вже містить у собі понад десяток стадій: передпроектні дослідження, технічне завдання, технічна пропозиція, ескізний проект, технічний проект, робочий проект, виготовлення, налагодження, випробування, введення в дію та здебільшого подальший технічний супровід та підтримка.

Розробка СІСТ являє собою величезну науково-технічну проблему, а її впровадження вимагає значних капіталовкладень. Усі СІСТ, які створено, та ті, що створюються за допомогою сформованої методології містять у собі результати багаторічних досліджень тисяч науковців, інженерів, конструкторів та програмістів, які брали участь у розробці проектних рішень.

За сучасними світовими тенденціями проектування СІСТ повинні бути такими, що розвиваються. Існує, принаймні, дві вагомні причини, за якими СІСТ повинні бути змінюваними за часом системами. По-перше: розробка такого складного об'єкта, як СІСТ, займає тривалий час і економічно вигідно вводити до експлуатації частини системи в міру їх готовності (введений в експлуатацію базовий варіант – надалі розширюється). По-друге: постійний прогрес об'єктів проектування, технологій виготовлення, обчислювальної техніки та обчислювальної математики призводить до появи нових, більш досконалих математичних моделей і методів, які повинні замінювати старі менш вдалі аналоги.

У зв'язку з цим, СІСТ повинні мати властивість зручності використання та можливості розширення за допомогою підключення розроблених та / або удосконалених видів забезпечення СІСТ. Ось тут постає питання подальшого вирішення цієї проблеми – це може бути нова розробка або реінжиніринг (reengineering).

Аналіз останніх розробок та публікацій, в яких розпочато вирішення зазначеної проблеми. Реінжиніринг містить у собі процеси реорганізації та реструктуризації СІСТ, переведення окремих компонентів системи в іншу, сучаснішу мову програмування, а також процеси модифікації або модернізації структури та системи даних [1]. При цьому архітектура системи може залишатися незмінною.

З комерційної точки зору, реінжиніринг часто вважають єдиним засобом збереження успадкованих модулів в експлуатації СІСТ. На думку провідних фахівців [2, 3], нову

розробку такої системи не рекомендується розглядати не тільки з точки зору дефіциту часу, що вже було затрачено на первинну розробку, та як слід – збільшення економічних витрат, а й з точки зору ризику виникнення структурних помилок.

У цей самий час реінжиніринг дає змогу виконати еволюціонування СІСТ шляхом позитивних змін видів її забезпечення з метою підвищення зручності її експлуатації та супроводу [4].

Також слід зупинитись на одній із найголовніших проблем створення СІСТ, а саме: уніфікація або універсальність [5]. Цю проблему, можна віднести до «проблеми початку», під якою слід розуміти, що всі сучасні СІСТ (різного галузевого призначення), на превеликий жаль, сучасними не є – це пов'язано, перш за все, з тим, що створювались вони на тих мовах, які були актуальні на самому початку їх розробки. Більшість із них через 1,5 – 2 роки не витримує підвищених вимог щодо швидкості роботи з відтвореним графічним зображенням та його обчислювальним відновленням (рендерінгом), а трансформація вихідного коду з однієї мови в іншу, виходячи з того, що сучасні СІСТ можуть складатися з декількох мільйонів рядків коду, може займати місяці та навіть роки.

Проблему реінжинірингу СІСТ різного галузевого призначення було детально розглянуто у [6, 7]. Методологічні засади реінжинірингу, які можна віднести й до СІСТ, було закладено у [8].

Актуальність статті виходить з перспектив розробки способу реінжинірингу СІСТ, як засобу отримання нового компонента шляхом виконання послідовності операцій внесення змін, модернізації або модифікації, а також перепрограмування окремих компонентів програмних систем. Цей спосіб повинен містити у собі процеси реорганізації та реструктуризації системи, переведення окремих компонентів системи в іншу, сучаснішу мову програмування, а також процеси модифікації або модернізації структури і системи даних.

До причин, що перешкоджають найскорішому використанню способу, можна віднести відсутність опису шляхів відновлення та наслідування програмної архітектури системи, що перепрограмується.

Формулювання мети статті. Високої ефективності СІСТ, яка виражається, перш за все, через мінімізацією часових, а, відповідно, й матеріальних витрат при експлуатації СІСТ, можна домогтися за рахунок удосконалення її ПЗ.

В основу статті покладено задачу розробки способу реінжинірингу СІСТ шляхом еволюції позитивних змін її ПЗ, що забезпечить підвищення зручності експлуатації, супроводу та використання, надасть можливості розширення за допомогою підключення розроблених та / або удосконалених видів забезпечення СІСТ.

Таким чином, **метою статті** є розробка, перевірка та практична реалізація способу мультилінгвістичного перекодування ПЗ, що дозволять на промисловій основі удосконалювати складні ІСТ.

Формулювання способу. Названа мета статті досягається тим, що спосіб мультилінгвістичного перекодування програмного забезпечення СІСТ можна звести до визначеної наведеної послідовності, що включає виконання нижченаведених сформульованих пунктів.

1 Обрання кінцевої мови програмування. Виконання цього пункту залежить від технічного завдання або сучасних вимог ринку, зведених у так званий топ-список найпопулярніших мов програмування, до яких належать: C; C++; C#; Java; PHP; Delphi; Python; Visual Basic тощо.

2 Вибір оптимального CASE-засобу. Вибір CASE-засобу залежить від уподобань користувача. На думку авторів, найоптимальнішим CASE-засобом, що підтримує імпортування та генерацію коду, написаного мовами, які наведено вище, є Enterprise Architect (EA). Саме EA (версія 13.5) будемо розглядати як ефективний інструмент перекодування.

3 Ідентифікація компонентів. Далі необхідно визначитись який саме компонент СІСТ треба піддати перекодуванню. Для прикладу розглянемо обрання будь-якого відкритого компонента, що написаний мовою С (рис. 1). Нехай це буде компонент «type.h» (див. рис. 1).

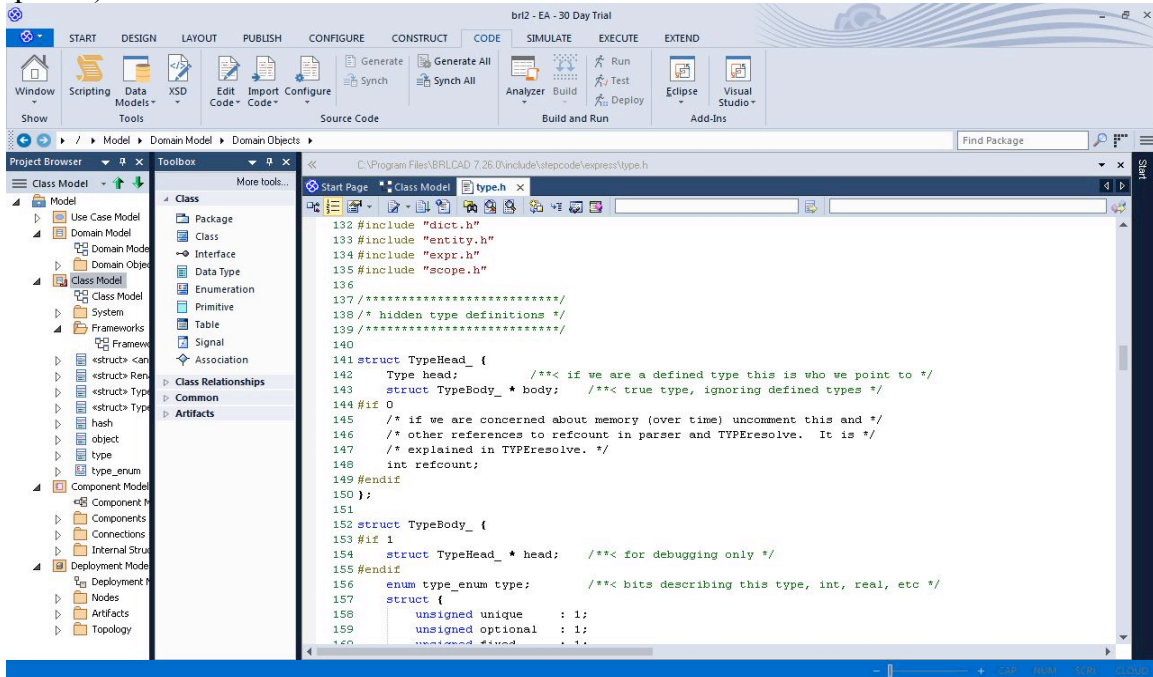


Рисунок 1 – Представлення компонента за допомогою вбудованого до ЕА редактора коду

4 Імпорт компонентів. Імпорт компонентів вже виконаємо за допомогою ЕА (рис. 2), де послідовно після встановлення необхідних параметрів, оберемо:

Code → Source Code → Import Code → C Files...

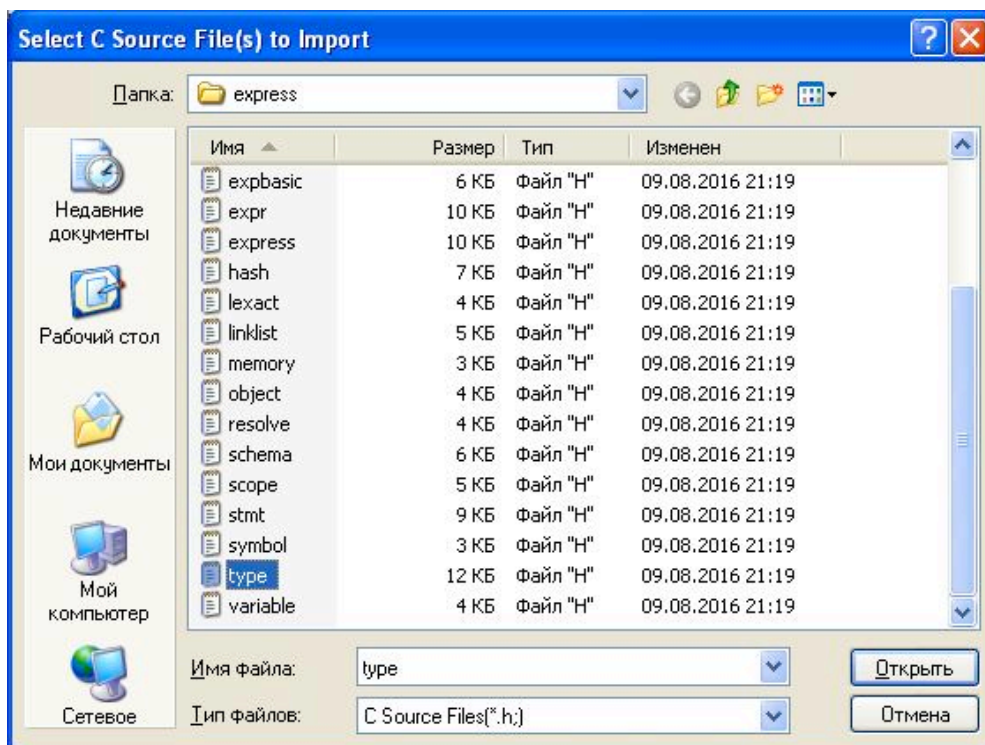


Рисунок 2 – Імпорт одного з компонентів, що написаний мовою С

Так само можна виконати імпорт будь-якого компонента, написаного будь-якою мовою із переліку мов, що випадає, які включено до ЕА.

Якщо імпорт компонентів було виконано вірно, то після зворотного інжинірингу імпортованого коду з'явиться відновлена структура первісного компонента (рис. 3).

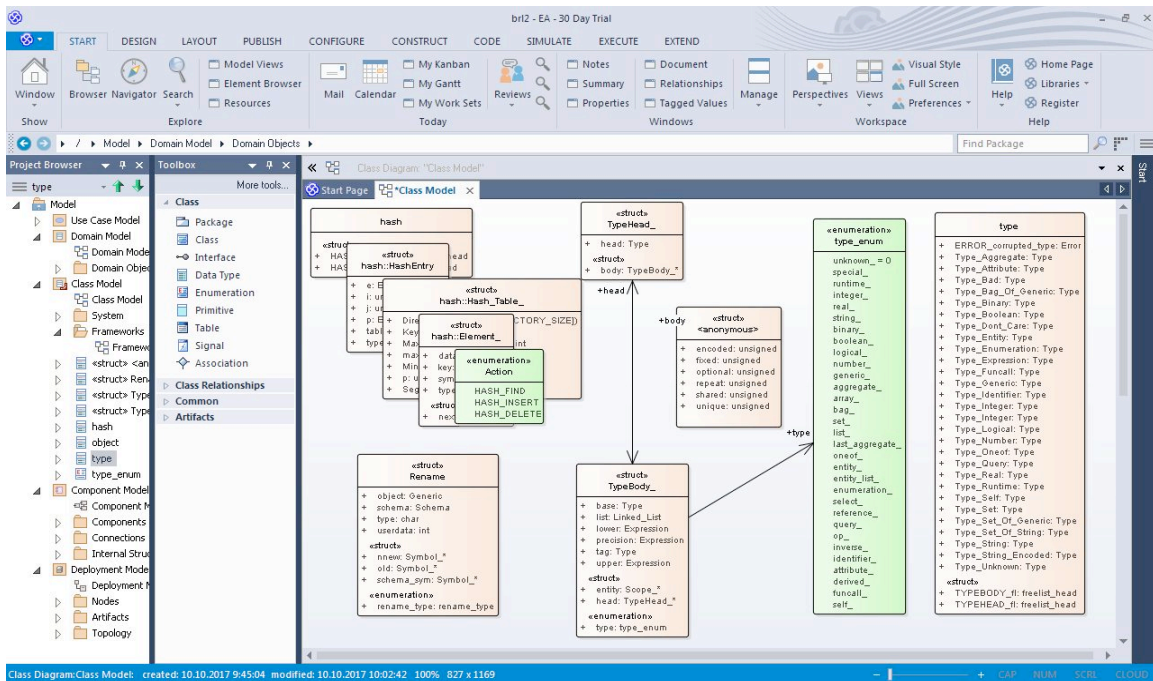


Рисунок 3 – Відновлена структура імпортованого компонента

5 Генерація коду нової структури. Для виконання генерації коду після встановлення необхідних параметрів ЕА, до яких входить виділення потрібних структурних елементів, необхідно виконати наступні дії зі сформованою структурою:

Source Code Engineering → *Generate Current Element...*

Після чого з'явиться запит – «Генерація коду» (рис. 4), у якому треба вказати місце для збереження нового компонента, перекодованого вже новою мовою.

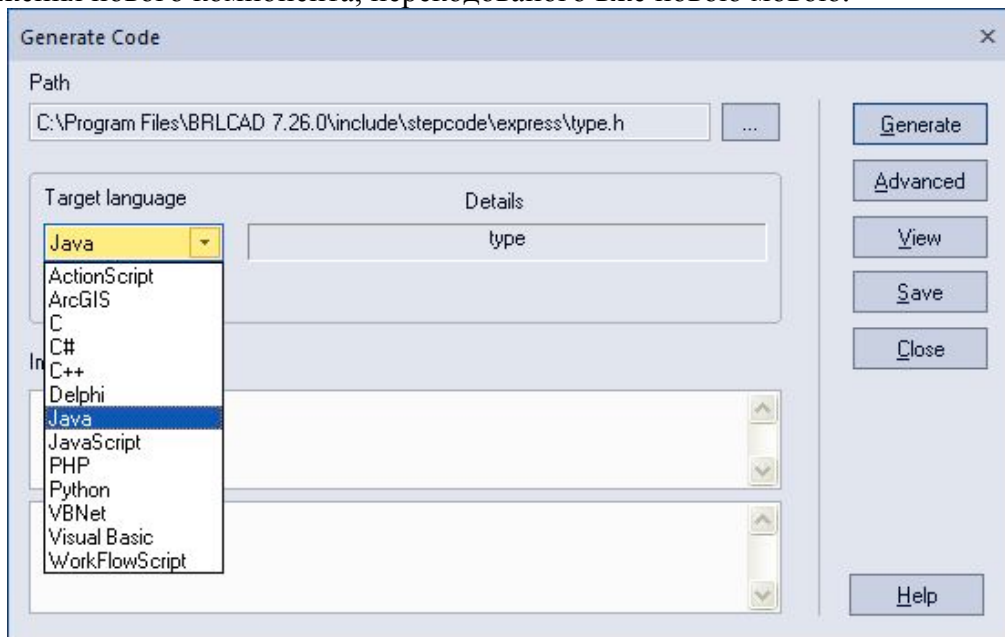


Рисунок 4 – Приклад запиту «Генерація коду»

Далі вже обираємо мову, у яку буде перекодований компонент, наприклад: Java (рис. 4) та натискаємо кнопку «Generate».

Після цього виконується процес перекодування, за яким можна спостерігати за допомогою повідомлень декількох системних вікон (рис. 5).

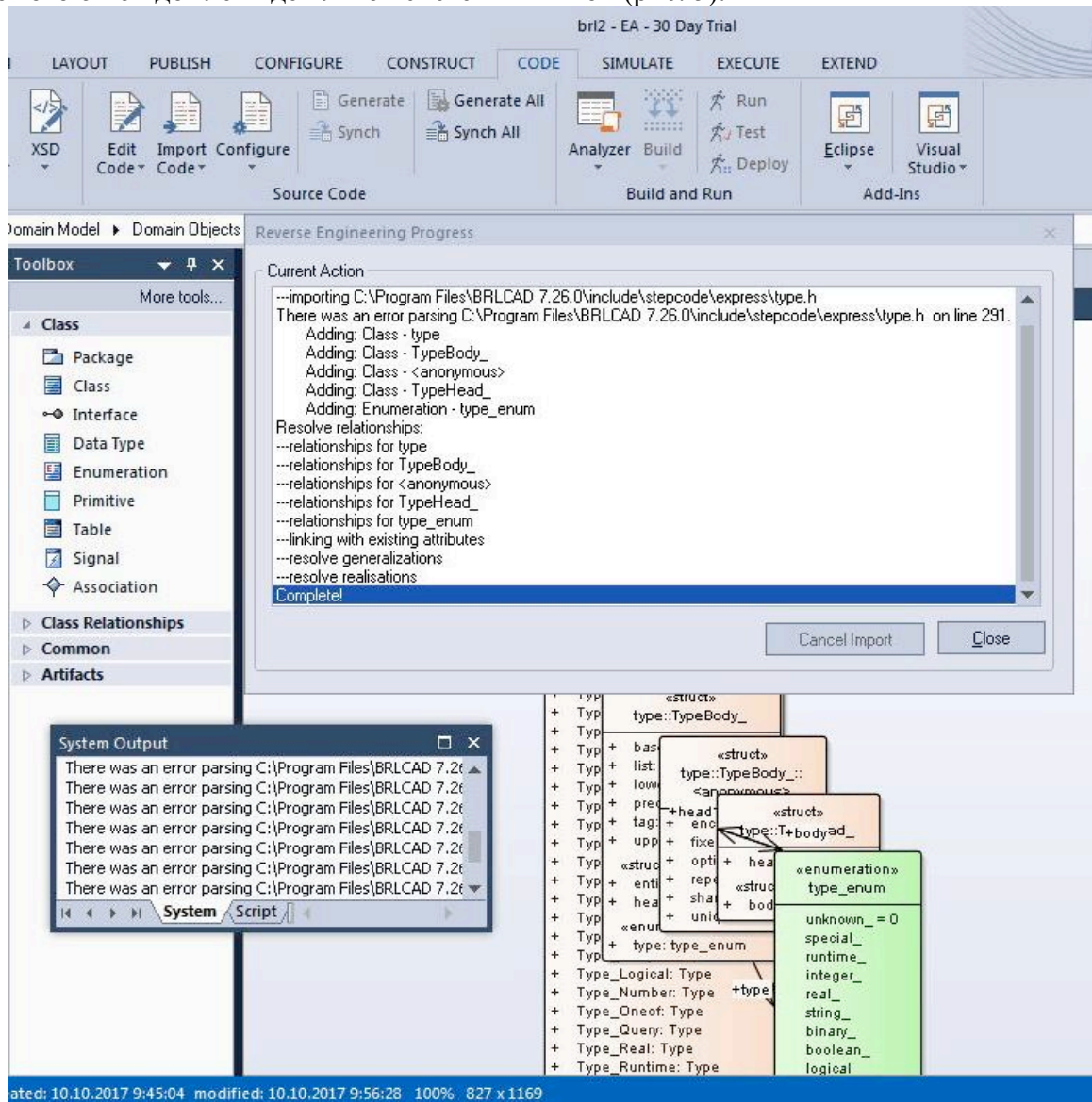


Рисунок 5 – Процес перекодування за допомогою системних повідомлень

Якщо все виконано вірно, то указаному місці з'явиться перекодований необхідною мовою компонент СІСТ.

6 Редагування перекодованого компоненту. Цей компонент СІСТ може бути доступний для подальшого редагування або підключення до нової СІСТ як за допомогою ЕА, так і за допомогою будь-якого редактора, орієнтованого на вже нову перекодовану мову.

Висновки та перспективи подальшого розвитку способу. Отже, у поданій статті описано новий спосіб мультилінгвістичного перекодування програмного забезпечення СІСТ. Цей спосіб складається із чітко визначених етапів, об'єднаних в описану послідовність.

Вірно виконане перекодування ПЗ СІСТ обов'язково повинно характеризуватися досягненням таких результатів:

- зниження ризику виникнення помилок при майбутньому оновленні СІСТ;

– зниження собівартості продукту за рахунок повторного використання компонентів ПЗ при розробці нової СІСТ;

– скорочення працездатності створення СІСТ за рахунок майже повного виключення рутинних операцій перепрограмування багатьох вже ідентифікованих компонентів.

Планується, що використання способу надасть значне підвищення ефективності застосування ІСТ у таких галузях їх використання: сфера телекомунікацій, управління виробництвом та транспортом, освіта тощо.

Розроблений спосіб стане корисним системним архітекторам та інженерам-програмістам, які задіяні у перепроєктуванні програмного забезпечення ІСТ, що вже знаходяться у багаторічній експлуатації.

Перспективи розвитку наведених досліджень полягають у створенні моделей реінжинірингу для кожного з інших видів забезпечення ІСТ, що будуть перепроєктовані. Реінжиніринг ІСТ дозволить подолати протиріччя між темпами розвитку науки і техніки та процесів проектування, підвищити ефективність технічного супроводу ІСТ, скоротити експлуатаційні витрати.

ЛІТЕРАТУРА:

1. Лаврищева Е. М. Сборочное программирование. Основы индустрии программных продуктов / Е. М. Лаврищева, В. Н. Грищенко. – [2-е изд. доп. и перераб.]. – К.: Наук. думка, 2009. – 372 с.
2. Норенков И. П. Основы автоматизированного проектирования: учебник [для вузов] / Норенков И. П. – [4-е изд., перераб. и доп.]. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. – 430 с.
3. Тимченко А. А. Основы системного проектирования та системного аналізу складних об'єктів. Кн. 1. Основы САПР та системного проектирования / Тимченко А. А. – К.: Либідь, 2000. – 272 с.
4. Пантелеймонов А. А. Аспекты реинженерии приложений с графическим интерфейсом пользователя / А. А. Пантелеймонов // Проблемы программирования. – 2001. – № 1–2. – С. 53–62.
5. Фаулер М. Рефакторинг: улучшение соответствующего кода / Фаулер М. – СПб.: Символ-Плюс, 2003. – 432 с.
6. Великодний С. С. Проблема реинжиниринга видов обеспечения систем автоматизированного проектирования / С. С. Великодний // Междун. науч. журн. «Управляющие системы и машины». – 2014. – № 1. – С. 57–61, 76.
7. Великодний С. С. Реинжиниринг систем мониторингу та дистанційного управління судновими енергетичними установками / С. С. Великодний // Матер. XXII міжн. конф. з автот. управл. [«Автоматика 2015»], (Одеса, 10 – 11 верес. 2015 р.). – Одеса, 2015. – С. 133–134.
8. Великодний С. С. Методологические основы реинжиниринга систем автоматизированного проектирования / С. С. Великодний // Междун. науч. журн. «Управляющие системы и машины». – 2014. – № 2. – С. 39 – 43.

REFERENCES:

1. Lavrisheva E. M., Grischenko V. N. Sborochnoe programmirovaniye. Osnovyi industrii programnykh produktov / E. M. Lavrisheva, V. N. Grischenko. [2- izd., dop. i pererab.]. – K.: Nauk. dumka, 2009. – 372 p.
2. Norenkov I. P. Osnovy avtomatizirovannogo proektirovaniya: uchebnyk [dlya vuzov] / Norenkov I. P. – [4- izd., pererab. i dop.]. – M.: Izd-vo MG TU im. N. E. Bauman, 2009. – 430 p.
3. Timchenko A. A. Osnovy sistemnogo proektuvannya ta sistemnogo analizu skladnykh ob'ektlv. Kn. 1. Osnovy SAPR ta sistemnogo proektuvannya / Timchenko A. A. – K.: Libld, 2000. – 272 p.
4. Panteleymonov A. A. Aspekty reinzhenerii prilozheniy s graficheskim interfeysom polzovatelya / A. A. Panteleymonov // Problemy programmirovaniya. – 2001. – № 1 – 2. – P. 53 – 62.
5. Fauler M. Refaktoring: uluchshenie sootvetstvuyushchego koda / Fauler M. – SPb.: Simvol-Plyus, 2003. – 432 p.
6. Velykodniy S. S. Problema reinzhiniringa vidov obespecheniya sistem avtomatizirovannogo proektirovaniya / S. S. Velykodniy // Mezhdun. nauch. zhurn. «Upravlyayuschie sistemy i mashiny». – 2014. – № 1. – P. 57 – 61, 76.
7. Velykodniy S. S. Reinzhiniring sistem monitoringu ta distantsiynogo upravlinnya sudnovimi energetichnimi ustanovkami / S. S. Velykodniy // mater. XXII mizhn. konf. z avtom. upravl. [«Avtomatika 2015»], (Odesa, 10 – 11 sep 2015.). – Odesa, 2015. – P. 133 – 134.
8. Velykodniy S. S. Metodologicheskie osnovy reinzhiniringa sistem avtomatizirovannogo proektirovaniya / S. S. Velykodniy // Mezhdun. nauch. zhurn. «Upravlyayuschie sistemy i mashiny». – 2014. – № 2. – P. 39 – 43.