УДК 618.086

В.П. КЛИМЕНКО, А.Л. ЛЯХОВ, Д.Н. ГВОЗДИК

РЕИНЖИНИРИНГ СИСТЕМЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ АНАЛИТИК

Анотація. *У даній роботі розглядається новий етап розвитку системи комп'ютерної алгебри АНАЛІТИК – реінжиніринг на основі нових прогресивних технологій. Описано процес реінжинірингу, досягнуті та очікувані результати. Показано, що в результаті реінжинірингу, окрім удосконалень, система набуває якісно нових можливостей та потенціалу для подальшого розвитку.*
Ключові слова: *АНАЛІТИК, системи комп'ютерної алгебри, реінжиніринг.*

Аннотация. *В данной работе рассматривается новый этап развития системы компьютерной алгебры АНАЛИТИК – реинжиниринг на основе новых прогрессивных технологий. Описаны процесс реинжиниринга, достигнутые и ожидаемые результаты. Показано, что в результате реинжиниринга, кроме усовершенствований, система приобретает качественно новые возможности и потенциал для дальнейшего развития.*
Ключевые слова: *АНАЛИТИК, системы компьютерной алгебры, реинжиниринг.*

Abstract. *In the article we consider a new stage of development of computer algebra system ANALITIC – reengineering based on new advanced technologies. Process of reengineering, achieved and expected results are described. It is shown that as a result of reengineering, except improvements, the system acquires the qualitatively new opportunities and the potential for further development.*
Key words: *ANALITIC, computer algebra systems, reengineering.*

1. Введение

Основанное в СССР академиком В.М. Глушковым направление теоретических и прикладных исследований по созданию языков семейства АНАЛИТИК [1] продолжает активно развиваться. В настоящее время коллективом разработчиков, состоящим из сотрудников Института проблем математических машин и систем НАН Украины (г. Киев) и Полтавского национального технического университета имени Ю.Кондратюка (г. Полтава), выполняется комплекс работ по созданию систем компьютерной алгебры (СКА) с входными языками нового поколения семейства АНАЛИТИК.

На основании анализа современных тенденций развития науки и инженерии, систем компьютерной математики как средств автоматизации этой деятельности, а также опыта решения прикладных задач, были сформулированы требования к языкам нового поколения [2–6], основные из которых следующие:

1. Одной из основных тенденций развития науки и инженерии является стремительный рост сложности и объема данных задач, что ведет к критическому падению продуктивности их интерактивного решения. Уровень языка СКА должен быть достаточным для программирования автоматического выполнения всех функций решателя, включая интеллектуальные.

2. Согласно разработанной теоретико-множественной модели задачи, данные в процессе решения представляют собой не совокупность отдельных выражений, а целостный объект с дуализмом свойств – комплекс, порожденный отношением абстрактной зависимости над множеством атомарных данных, и комплекса, порожденного отношением частичной упорядоченности над множеством операций, выполняемых в процессе решения. В частном случае комплексы могут быть иерархическими структурами, подобными используемым при описании научных и технических теорий.

3. Входной язык должен быть набором базисных средств, достаточно полным для программирования процесса распознавания свойств и преобразования таких объектов.

4. Входной язык должен содержать набор базисных средств, достаточно полный для проблемной ориентации компактного ядра и интерфейса СКА.

К настоящему времени в соответствии с этими требованиями разработаны спецификации языка АНАЛИТИК-2000 [7] и его модификация АНАЛИТИК-2007 [8].

В качестве базиса для программной реализации этих языков использовалась реализация языка АНАЛИТИК-93 [9]. В настоящее время реализация АНАЛИТИК-2007 перешла на стадию системного тестирования, а проект в целом – к стадии создания версии, ориентированной на широкого пользователя.

2. Реинжиниринг СКА АНАЛИТИК

В процессе анализа сборки и тестирования сборки оказалось, что реализация на основе АНАЛИТИК-93 (который относится к началу 1990-х годов) не может обеспечить совокупность системных свойств, в полной мере соответствующих изложенным выше требованиям.

Разработка систем АНАЛИТИК-2000, -2007 по объективным причинам выполнялась в течение достаточно долгого времени и не удовлетворяет условиям стандартизации и новациям в информационных технологиях и программировании.

Реализация также не удовлетворяет ни одному из существующих стандартов программного обеспечения (ПО) – не содержит потенциала для развития и адаптации к быстро изменяющимся современным внешним и внутренним условиям применения программных продуктов и, в частности, систем компьютерной математики.

Исходя из изложенных соображений, было принято решение о реинжиниринге [10] системы на основе современных технологий и с учетом современных тенденций в развитии программного обеспечения в области компьютерной математики.

2.1. Реверс-инжиниринг СКА АНАЛИТИК

Существующая реализация АНАЛИТИК-2007 создавалась в течение длительного времени и разными программистами, со многими из которых связь утеряна, следствием чего является существенная нехватка информации о структуре и способе создания системы. Представляется целесообразным в качестве первого этапа реинжиниринга выполнить реверс-инжиниринг, т.е. исследование системы и имеющейся документации с целью понять принципы ее работы. Конечная цель этого этапа – выявить все ключевые подсистемы, обеспечивающие работу СКА в целом, механизмы их взаимодействия и их воспроизведение на основе новых технологий, но с аналогичными функциями.

Исследования показали, что в имеющейся реализации внимание уделялось в основном свойствам языка, его сигнатуре, типам данных, алгоритмам, операторам и функциям языка для их обработки, которые тщательно тестировались и являются вполне надежными. Соответствующую часть программной реализации было решено оставить без особых изменений.

Вместе с тем в процессе испытаний был выявлен целый ряд существенных недостатков. Представляется необходимым в первую очередь устранить такие:

1. Следует полностью изменить подсистему, обрабатывающую ошибки. Ранее использовался механизм АВОСТов – аварийных остановок работы ядра программы при возникновении ошибок. При работе с системой в режиме «командной строки» такая реализация особых проблем не создавала. Однако при работе с системой, обладающей GUI, такая реализация делает ее неработоспособной – при АВОСТе работа или аварийно завершается, или оставшийся «мусор» не позволяет корректно продолжать работу.

Предлагается подсистема, позволяющая автоматически и корректно обрабатывать возникшие ошибки с помощью специальных языковых конструкций.

2. АНАЛИТИК обладает механизмом проблемной ориентации ядра, имеющей определенные преимущества над аналогичными [8]. В настоящее время для этого используется единая библиотека функций, которую можно изменять, добавляя или удаляя пользовательские функции. Однако при таком подходе функции разной проблематики размещаются в одной и той же библиотеке и не сгруппированы по областям применения, что создает определенные трудности.

Представляется необходимым, сохраняя для пользователя возможность проблемно ориентировать ядро, полностью изменить механизм проблемной ориентации ядра во внутреннем представлении системы, в частности, механизмы создания и подключения пользовательских функций. Вместо библиотеки функций предложен механизм, дающий возможность как добавлять отдельные пользовательские функции и пользовательские операторы, так и пакеты функций и операторов. Под пакетом понимается некий набор пользовательских функций и операторов, объединенных согласно некоторому принципу, который целесообразен для пользователя.

Благодаря предложенному механизму, обеспечивается возможность накопления и каталогизации пользовательских функций из разных областей знаний, обеспечивающих решение задач соответствующей проблематики. В частности, в разных пакетах могут содержаться функции с одинаковыми именами.

2.2. Прямой реинжиниринг СКА АНАЛИТИК

Прямой реинжиниринг программного обеспечения – это процесс создания новой функциональности или устранения ошибок путём кардинального изменения разработанного программного обеспечения [10].

Архитектура текущей реализации продиктована парадигмой структурного программирования, которая являлась господствующей при проектировании программных систем и организации труда программистов в конце 90-х годов, и поэтому она требует кардинального изменения.

В основу разработки новой архитектуры положен объектно-ориентированный подход. Ядро реализуется как динамически подключаемая библиотека с набором классов, достаточно полным для выполнения системы требований к АНАЛИТИКу.

В качестве платформы для реализации была выбрана .NET 2.0, что позволяет сразу естественным образом решить одну из наиболее серьезных проблем текущей реализации – проблему распределения памяти и сборку «мусора». Как показали эксперименты, существенного уменьшения производительности системы такой подход не вызывает по сравнению с используемым в СКА Maple, Mathematica, у которых эти функции выполняет ядро.

Существующая реализация .NET Framework для Unix-подобных систем (проект Mono) дает возможность запускать программы, написанные с использованием .NET, практически на всех операционных системах, что решает вопросы, связанные с кроссплатформенностью, актуальные на следующих этапах жизненного цикла СКА АНАЛИТИК.

Подход на основе .NET также обеспечивает дополнительные возможности:

• классы могут быть использованы не только интерфейсом, но и любой программой, написанной на .NET, что соответствует одной из современных тенденций в разработке программных продуктов – возможность интеграции различных программных систем;

• возможность использования технологии ASP.NET для создания Web-приложений на основе функциональности языка АНАЛИТИК без изменений ядра системы;

• может быть создана подсистема для тестирования СКА удаленным пользователем, что даст возможность более продуктивно и в короткие сроки провести как альфа, так и бета-тестирование;

• возможность реальных применений новых представлений о данных, как о комплексе над множеством связанных выражений, одной из парадигм языка АНАЛИТИК, для

организации параллельных вычислений как локальных, так и, например, в качестве ядра GRID-вычислений.

Существенным является и то, что в результате реинжиниринга набор классов ядра СКА АНАЛИТИК иерархически упорядочен. Это упрощает понимание работы ядра для конечного пользователя и обеспечивает выполнение требования п. 3 (см. выше). Для повышения производительности системы развит принцип повторного использования кода, написанного на C++ и Assembler и реализующего часто применяемые и трудоемкие методы.

2.3. Изменения и развитие СКА АНАЛИТИК

В результате реинжиниринга СКА АНАЛИТИК приобретает целый ряд новых возможностей. Предложенный механизм обработки ошибок направлен на развитие доминанты систем этого семейства. Он ориентирован на разработку автоматических программ. Более того, благодаря этому механизму, будет предоставлена возможность создавать в теле программы обработчики ошибок с элементами интеллектуального поведения, в соответствии с заданными или выработанными в процессе решения критериями распознавать существенные и несущественные ошибки и, соответственно, прекращать или не прекращать процесс автоматического решения.

Новая подсистема для работы с ошибками позволяет генерировать сообщения об ошибках на разных языках. В свете развития проекта в направлении создания версии СКА для широкого пользователя это означает, что приобретается возможность создавать локализации ядра без его изменения.

Замена механизма библиотеки функций на механизм пакетов пользователей также является не просто заменой, а развитием СКА АНАЛИТИК в направлении создания версии для широкого пользователя (бета-версии) и предпосылкой к следующему этапу – созданию коммерческой версии.

Механизм пакетов расширения качественно усиливает возможности по расширению функциональности и проблемной ориентации ядра. Само ядро при этом остается компактным и содержит только базисный набор функций, что соответствует требованиям к системе.

Языковый механизм для работы с пакетами будет сохранен (как и ранее, для подключения и исключения функций, пакетов, операторов будут использоваться функции ИСКЛ и ВКЛ). Вызов функции из пакета может происходить как по имени функции, если оно уникально, так и путем указания перед функцией имени пакета (например, Пакет.Функция1()). Подобный механизм используется в большинстве современных СКА, однако в результате реинжиниринга закладывается большой потенциал для развития АНАЛИТИКа. В частности, в новой реализации в пакете могут содержаться не только сами функции, но и справочная информация по ним, примеры вызова и использования и др.

В соответствии с требованиями АНАЛИТИК ориентирован на решение сложных задач с большими объемами данных. В настоящее время в качестве хранилищ больших объемов данных используются преимущественно базы данных. Логичным развитием указанного свойства является введение в язык нового типа данных «Таблица». Такое развитие языка и архитектура на основе .NET дают возможность СКА АНАЛИТИК работать практически со всеми современными базами данных и знаний.

Введение типа «Таблица» существенно расширит область применимости АНАЛИТИКа. На основе этого типа возможно внедрение в язык АНАЛИТИК SQL-подобных конструкций. В перспективе такие конструкции будут работать не только с типом «Таблица», но и с векторами. Например, появляется возможность выбирать все координаты из вектора, зависящие от заданного выражения, и др. Представляет интерес и возможность с по-

мощью типа «Таблица» сохранять результаты вычислений в базах данных, подготовленных как АНАЛИТИКом, так и другими программными системами.

Внедрение в АНАЛИТИК SQL-подобных конструкций может стать основой качественно новых технологий обработки аналитических выражений. Более того, тип данных «Таблица» может стать основой принципиально нового подхода к реализации представлений о структуре данных (п. 2), которые являются еще одной доминантой языков семейства АНАЛИТИК.

В настоящее время АНАЛИТИК ориентирован, в основном, на аналитические преобразования и численные расчеты. Возможности для работы с графикой весьма скромные.

В процессе реинжиниринга планируется реализация базовых функций для работы с 2D и 3D графическими примитивами, графиками функций и др. С этой целью вводятся новый класс «Графический объект» и базовые методы для работы с экземплярами этого класса. На основе этого класса становится возможным построение сколь угодно сложных графических объектов, а в целом АНАЛИТИК становится полноценным инструментом решения и визуализации решения сложных прикладных задач численно-аналитическими методами.

3. Выводы

В результате реинжиниринга АНАЛИТИК будет отличаться от текущей реализации следующим образом.

• Новая реализация осуществляется с использованием перспективных технологий на основе платформы .NET.

• Такой подход позволяет сохранить все существенные функциональные возможности предыдущих реализаций АНАЛИТИКа.

• Благодаря введению SQL-подобных конструкций, язык АНАЛИТИК приобретает качественно новые изобразительные возможности, что можно рассматривать как повышение уровня интеллекта языка.

• Вместе с тем, благодаря качественным изменениям в архитектуре, новая реализация в значительно большей мере соответствует требованиям к конечной цели проекта.

Перечисленные выше сделанные и планируемые изменения и нововведения на основании реинжиниринга СКА АНАЛИТИК в целом дают возможность рассматривать АНАЛИТИК как язык четвертого поколения и позволяют говорить о новом этапе развития системы.

СПИСОК ЛИТЕРАТУРЫ

1. Клименко В.П. Развитие и реализация идей ЭВМ МИР / В.П. Клименко // Третья международная научно-техническая конференция «Компьютерная математика в науке, инженерии и образовании» (CMSEE-2009), (Полтава, 1–31 октября 2009 г.). – Киев: Изд-во НАН Украины, 2009. – С. 5 – 11.
2. Фишман Ю.С. Основные особенности создания и применения средств реализации численно-аналитических методов / Ю.С. Фишман // Математические машины и системы. – 1998. – № 2. – С. 9 – 18.
3. Клименко В.П. Основные тенденции развития языков систем компьютерной алгебры / В.П. Клименко, А.Л. Ляхов, Ю.С. Фишман // Математичні машини і системи. – 2002. – № 2. – С. 29 – 64.
4. Ляхов О.Л. Деякі сучасні проблеми застосування чисельно-аналітичних методів / О.Л. Ляхов // Математичні машини і системи. – 2003. – № 2. – С. 54 – 65.
5. Клименко В.П. Прикладная математическая задача как объект компьютерной алгебры / В.П. Клименко, А.Л. Ляхов // Математичні машини і системи. – 2003. – № 3–4. – С. 103 – 123.
6. Клименко В.П. Інтелектуалізація розв'язування складних прикладних задач методами комп'ютерної алгебри / В.П. Клименко, О.Л. Ляхов. – К.: Логос, 2009. – 293 с.
7. АНАЛИТИК-2000 / А.А. Морозов, В.П. Клименко, А.Л. Ляхов [и др.] // Математичні машини і системи. – 2001. – № 1–2. – С. 66 – 99.

8. АНАЛИТИК-2007 / А.А. Морозов, В.П. Клименко, Ю.С. Фишман [и др.] // Математичні машини і системи. – 2007. – № 3–4. – С. 8 – 52.
9. АНАЛИТИК-93 / А.А. Морозов, В.П. Клименко, Ю.С. Фишман [и др.] // Кибернетика и системный анализ. – 1995. – № 5. – С. 127 – 157.
10. Chikofsky E. Reverse Engineering and Design Recovery: A Taxonomy / E. Chikofsky, J. Cross // IEEE Software. – 1990. – N 7 (1). – P. 13 – 18.

Стаття надійшла до редакції 23.06.2010