УДК 355.01: 004.056

В.А. Маклаев

ПРЕДСТАВЛЕНИЕ АКТИВОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В РЕПОЗИТОРИИ БАЗЫ ОПЫТА ПРОЕКТНОЙ ОРГАНИЗАЦИИ

Маклаев Владимир Анатольевич, *кандидат технических наук, окончил радиотехнический факультет Ульяновского политехнического института. Генеральный директор ФНПЦ ОАО «НПО «Марс». Имеет статьи в области САПР. [e-mail: mars@mv.ru].*

Аннотация

Представляются средства извлечения активов из исходных кодов программных продуктов для их размещения в репозитории базы опыта проектной организации. Утилиты, обеспечивающие извлечение активов, запрограммированы на специализированном псевдо-кодовом языке. Представлена операционная обстановка доступа проектировщиков к репозиторию.

Ключевые слова: автоматизированное проектирование, активы, база опыта, репозиторий, псевдо-кодовое программирование.

Vladimir Anatolyevich Maklaev, *Candidate of Engineering; graduated from the Faculty of Radioengineering of Ulyanovsk Polytechnic Institute; Director General of Federal Research-and-Production Center Open Joint-Stock Company 'Research-and-Production Association 'Mars'; author of articles in the field of CAD. e-mail: mars@mv.ru.*

Abstract

The article presents facilities to extract assets from source codes of software products in order to locate them into a design-organization experience-base repository. Utilities ensuring the asset extraction, are programmed based on a special-purpose pseudocode language. The article also presents an operational environment for designers' access to the repository.
Key words: computer-aided design, assets, experience base, repository, pseudocode programming.

ВВЕДЕНИЕ

Для любой проектной организации, разрабатывающей «семейство автоматизированных систем», в производственную деятельность необходимо включить процессы выявления, представления, систематизации и хранения в удобной форме всего, что полезно для повторного использования в разработках очередных автоматизированных систем (АС). Единицы такого повторного использования получили название «активов».

Под «активами» (в узком смысле) принято понимать **завершенные артефакты проектирования, пригодные для самостоятельного применения, которые могут быть использованы в дальнейшей работе организации как в качестве образцов, так и в качестве готовых к повторному использованию компонентов.** Часто такое понимание расширяют, включая в него не только артефакты проектирования, но и всё то, что обеспечивает разработку АС: **инструментальные средства, технологии, кадровый и организационный потенциал, инфраструктуру и другое обеспечение процессов проектирования.** Вводя в производственную деятельность систему работы с активами, в первую очередь следует разобраться с **типологией активов** и их представлениями, эффективными при повторном применении.

Любой из активов, явно и/или неявно, имеет отношение к опыту, вложенному в этот актив в процессе предшествующих разработок. К классу явных активов относятся, например, завершённые проекты как целое, руководства по типовым бизнес-процессам, шаблоны проектных документов и библиотеки программ для полезных проектных решений. Примером актива, имеющего неявное отношение к опыту, может служить определённый инструментарий, от степени освоенности которого проектировщиками зависит результативность применения инструментария в очередном проекте.

Учитывая связь активов с опытом, для их хранения и применения принято разрабатывать «фабрики опыта» [3] и «базы опыта» [4], принципиальной составляющей которых является «репозиторий активов» [1]. В статье представляется репозиторий активов, встроенный в базу опыта [1], в построении которой используется вопросно-ответное моделирование процессов проектирования и их объектов [2, 5]. Основное внимание уделяется каталогизации активов, извлечённых из программных продуктов, и операционным средам взаимодействия пользователей с репозиторием.

ОБЩИЕ ВОПРОСЫ ТИПОЛОГИИ АКТИВОВ

Как уже было отмечено, в создании базы опыта и её репозитория в первую очередь следует определиться с типологией активов и нормативным представлением активов каждого типа. Для организаций, разрабатывающих

семейства АС, в такой работе целесообразно начинать с инвариантной типологии активов, определённой стандартом Framework for Software Product Line Practice-Version 5.0. В приложении стандарта к разработке семейств АС необходимо учитывать и другие стандарты, например, стандарт ИСО/МЭК 12207, осуществляя выбор типов активов по следующим направлениям повторного использования:

Требования. Существенная часть требований к продуктам семейства АС является общей, что значительно упрощает разработку и сопровождение требований (requirement engineering) для отдельных систем.

Архитектура. В рамках семейства продуктов разрабатывается архитектура типовой системы. Архитектура конкретной системы создается на ее основе, и тем самым существенно экономятся ресурсы на проектирование.

Компоненты. Общая для всех представителей семейства функциональность реализуется в виде повторно используемых компонентов, с учётом особого внимания к программным компонентам, из-за чего разработка систем значительно упрощается.

Различные модели анализа, проектирования, производительности и т. д. также могут повторно использоваться при создании продуктов семейства.

Средства разработки. В рамках семейства продуктов формируется общая для разных продуктов технологическая среда: средства разработки (IDEs - Integrated Development Environments), СУБД, средства поддержки планирования, тестирования, конфигурационного управления и т. д. Кроме того, могут создаваться специальные программные средства, «заточенные» под специфику данного семейства (например, кодогенераторы по визуальным моделям).

Процессы. Здесь речь идет о повторном использовании приемов работы с заказчиком, методов управления проектом, способов работы с планами и о других процедурах процесса разработки.

Квалификация сотрудников. Поскольку системы, разрабатываемые в рамках семейства, похожи, а также существует общая инфраструктура разработки: технологии и программные средства, процедуры процесса, знания в предметной области, на которую ориентировано семейство, и т. д., то в такой ситуации легко «передвигать» работников из одного проекта в другой (например, при необходимости «усилить» какой-то проект), либо после окончания одного проекта подключать разработчиков к новому проекту, либо вводить в уже существующий.

Так как для АС особо важны активы, связанные с разработкой их программного обеспечения, детализацию типов и их представление в репозитории проведём только для программных комплексов. Такой акцент упрощает и выбор инвариантных типов, в составе которых выделим активы, извлечённые из исходных кодов, и активы, представляющие структуру программного обеспечения. Независимо от ограничений, в рамках которых в статье будут представлены структура и содержание репозитория, решения по операционным обстановкам взаимодействия с активами и реализации операционных обстановок являются общими.

УНИФИКАЦИЯ ПРЕДСТАВЛЕНИЯ АКТИВОВ

Построение базы опыта и её репозитория ориентировано на вопросно-ответное моделирование процессов проектирования и их результатов в инструментальной среде WIQA (Working In Questions and Answers), детально описанной публикациях [2] и [5]. Такая ориентация позволяет адаптировать средства этого инструментария к работе с активами, унифицировав их представление.

В такой унификации, причем только для программного обеспечения, раскроем два уровня: обобщённый уровень и детализацию обобщённого уровня в его приложении к программному комплексу. Обобщённый уровень базируется на соотнесении артефактов программного обеспечения и вопросно-ответной моделирующей среды, представленных на рисунке 1.

На обобщённом уровне с любой структурной единицей программного комплекса связано её представление «задачей» с вопросно-ответной моделью (QA-моделью), вопросная структура которой имеет следующий вид:

Z: <Имя актива>
Q1. Информация об активе?
Q2. Из какого проекта актив извлечён?
Q3. Статистика использования актива?
Q3.1. Число просмотров?
Q3.2. Число заимствований?
Q4. Ключи для поиска актива?
Q5. Дополнительная информация об активе?
Q6. QA-модель актива?
Q7. Ссылки на представление экземпляра актива в других типах активов?

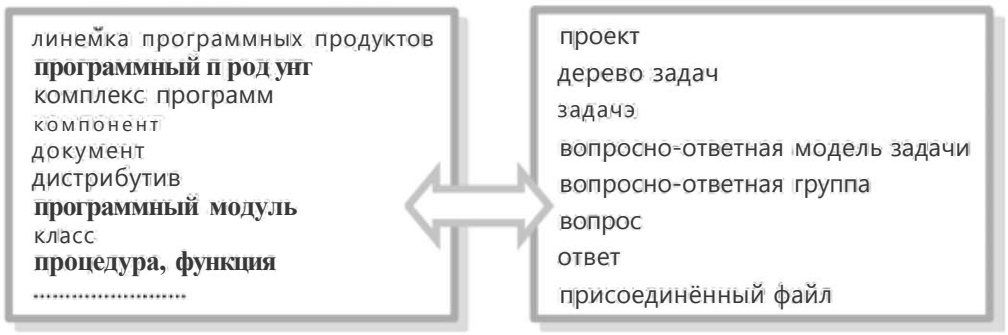

Рис. 1. Сопоставление активов со средствами их представления

Обобщённый уровень связан с детализацией актива через позицию Q6, выполняющую роль ссылки на представление актива с помощью соответствующей подчинённой задачи, которая включена в задачу библиотечной каталогизации программного продукта. Типовое библиотечное представление программного продукта, включающего программные комплексы и их составные единицы, приведено на рисунке 2. Если структурная единица относится к программным кодам, то её библиотечное представление доступно пользователям через позиции каталога Qx2 и Ax.k: Версия от <DATE> (см. рис. 2).

Библиотечное описание, представленное на рисунке 2, формируется шаг за шагом, каждый из которых фиксирует определённое состояние результатов работ по созданию программного продукта. В такой работе необходимы помощники, в первую очередь те, которые обеспечивают автоматическое извлечение структурных единиц из исходных программных кодов.

ПОДГОТОВКА «ИСХОДНЫХ КОДОВ» К ИЗВЛЕЧЕНИЮ АКТИВОВ

Для того, чтобы активы, извлекаемые из исходных кодов, можно было удобно использовать, необходимо обеспечить установку значений основных атрибутов этих активов. Соответственно, исходные коды необходимо предварительно прокомментировать, используя следующие предлагаемые форматы комментариев. Так как для построения исходных кодов применяют различные алгоритмические языки, их комментирование должно быть согласовано с ключевыми словами используемых языков. Задача извлечения активов из «исходных кодов» решена для группы языков, каждый из которых представляют следующие совокупности ключевых слов и «расширений» файлов:

1. Группы проектов (*.sln, *.bpg,...)

<SOLUTION>
<NAME>название группы проектов_</NAME>
<DESCRIPTION>_общее описание группы проектов_</DESCRIPTION>
^EYWORDS^Kra^, ключ_2, ключ_3, </KEYWORDS>
</SOLUTION>

Структура-комментарий помещается в начало файла, при этом каждая строка описывается в виде однострочного комментария (//). Для sLn-файлов комментарии вносятся в специальный файл с расширением comm, поскольку среда разработки не допускает никаких дополнительных текстов в таких файлах.

2. Проекты (*.pro, *.csproj, *.bpr,...)

<PROJECT>
<NAME>_название проекта_</NAME>
<DESCRIPTION>_описание проекта_</DESCRIPTION>
<KEYWORDS>MIO4_1, ключ_2, ключ_3, </KEYWORDS>
</PROJECT>

Структура-комментарий помещается в начало файла, при этом каждая строка описывается в виде однострочного комментария (// либо #). Для csproj-файлов комментарии вносятся в специальный файл с расширением comm, поскольку среда разработки не допускает никаких дополнительных текстов в таких файлах.

3. Программные модули (*.h, *.cpp, *.cs, *.pas,...)

<CSFILE>
<DESCRIPTION>_описание программного модуля_
</DESCRIPTION>
<KEYWORDS>MIO4_1, ключ_2, itnio4_3,...</KEYWORDS>
</CSFILE>

Структура-комментарий помещается в начало файла, при этом каждая строка описывается в виде однострочного комментария (//).

4. Пространства имен (*.cs,...)

<NAMESPACE>
<DESCRIPTION>_описание пространства имен_
</DESCRIPTION>
<KEYWORDS>K™4_1, ключ_2, ключ_3, </KEYWORDS>
</NAMESPACE>

Структура-комментарий помещается непосредственно перед первым упоминанием пространства имён, при этом каждая строка описывается в виде однострочного комментария (//).

5. Классы (*.h, *.cpp, *.cs, *.pas,...)

<CLASS>
<DESCRIPTION>_описание oacca_</DESCRIPTION>
<KEYWORDS>K™4_1, ключ_2, ключ_3, </KEYWORDS>
</CLASS>

Структура-комментарий помещается непосредственно перед описанием класса, при этом каждая строка описывается в виде однострочного комментария (//).

6. Свойства, процедуры, функции (*.cpp, *.cs, *.pas,...)

<METHOD>
<DESCRIPTION>_описание свойства, процедуры или функции_</DESCRIPTION>
<KEYWORDS>K™4_1, ключ_2, ключ_3, </KEYWORDS>
</METHOD>

Структура-комментарий помещается непосредственно перед реализацией метода, при этом каждая строка описывается в виде однострочного комментария (//).

ПСЕВДО-КОДОВОЕ ПРОГРАММИРОВАНИЕ В ИЗВЛЕЧЕНИИ АКТИВОВ

Для извлечения активов из исходных кодов на специализированном псевдо-кодовом языке [5], встроенном в реализацию базы опыта, разработана совокупность утилит, каждая из которых настроена на извлечение активов из исходных кодов на определённом алгоритмическом языке C++.

Специфику псевдо-кодовой обработки исходных кодов представим фрагментом утилиты для языка.

```
Z 10.3.5.1.2.1 Извлечение активов из cpp-файла
  Q 10.3.5.1.2.1.1 &cur_par& := &Current_
Parents;
  Q 10.3.5.1.2.1.2 &methodflag& := 0;
  Q 10.3.5.1.2.1.3 &qtflag& := 0;
  Q 10.3.5.1.2.1.4 LABEL &CYCLE1&;
  Q 10.3.5.1.2.1.5 &parid& := QA_GetParent
```

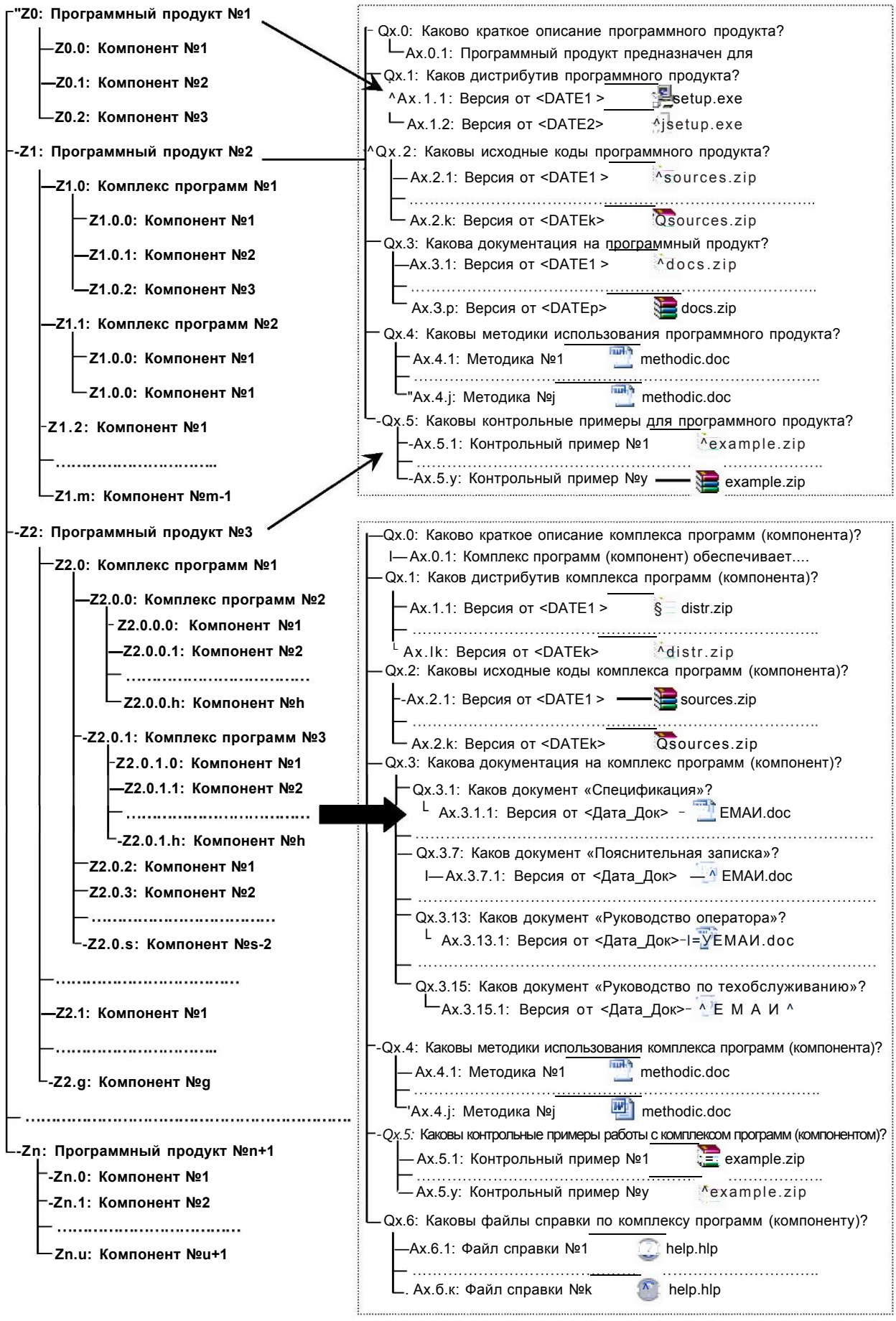

Рис. 2. Схема вопросно-ответной каталогизации разработанных программных продуктов в репозитории базы опыта

```
(&Current_Project&,&cur_par&);
   Q 10.3.5.1.2.1.6 IF &parid&==0 THEN GOTO
&CHECK& ELSE GOTO &GOUP&;
   Q 10.3.5.1.2.1.7 LABEL &GOUP&;
   Q 10.3.5.1.2.1.8 &partxt& := QA_GetQAText
(&parid&);
   Q 10.3.5.1.2.1.9 &clc& :=
STRCMP(&partxt&,"Свойства, процедуры, функции");
   Q 10.3.5.1.2.1.10 IF &clc&==0 THEN
&methodflag& := 1;
   Q 10.3.5.1.2.1.11 &qtc& :=
STRCMP(&partxt&,"QT  4");
   Q 10.3.5.1.2.1.12 IF &qtc&==0 THEN
&qtflag& := 1;
   Q 10.3.5.1.2.1.13 &cur_par& := &parid&;
   Q 10.3.5.1.2.1.14 GOTO &CYCLE1&;
   Q 10.3.5.1.2.1.15 LABEL &CHECK&;
   Q 10.3.5.1.2.1.16 &cur_txt& := QA_
GetQAText(&cur_par&);
   ..........................................
   Q 10.3.5.1.2.1.75 GOTO &FIN&;
   Q 10.3.5.1.2.1.76 LABEL &ERROR&;
   Q 10.3.5.1.2.1.77 ERRBOX("Извлечение
активов","Методика запущена в неверном задачном
контексте!");
   Q 10.3.5.1.2.1.78 LABEL &FIN&;
   Q 10.3.5.1.2.1.79 FINISH;
```

Отметим, что индексные имена операторов формируются автоматически.

ИНСТРУМЕНТАРИЙ ПОЛЬЗОВАТЕЛЦ АРХИВА АКТИВОВ

Доступ к активам, извлечённым из программных продуктов, открыт (в рамках предоставленных полномочий) с любого рабочего места корпоративной сети проектирования. К числу основных функций пользователя архива активов проектной организации относятся:

- доступ к архиву активов в соответствии с назначенными правами доступа;
- просмотр иерархий групп активов различных типов и выбор экземпляров активов для просмотра;
- просмотр данных выбранных экземпляров активов, включая прикрепленные к ним файлы;
- поиск активов в архиве по различным критериям;
- встраивание элементов данных выбранных активов в разрабатываемые проекты.

Для выполнения этих функций пользователю, помимо базовых средств работы с вопросно-ответными протоколами, предоставлены плагином возможности «Каталога активов» инструментария WIQA.NET.

Плагин «Каталог активов» является компонентом комплекса программ «Рабочее место пользователя вопросно-ответной среды WIQA.NET». Компонент «Каталог активов» предназначен для представления структурированных данных, хранимых в репозитории, в удобном для пользователя виде.

Основные функциональные возможности, предоставляемые компонентом «Каталог активов», следующие:

- визуализация каталогов активов в виде иерархического дерева;
- просмотр полной информации об активе;

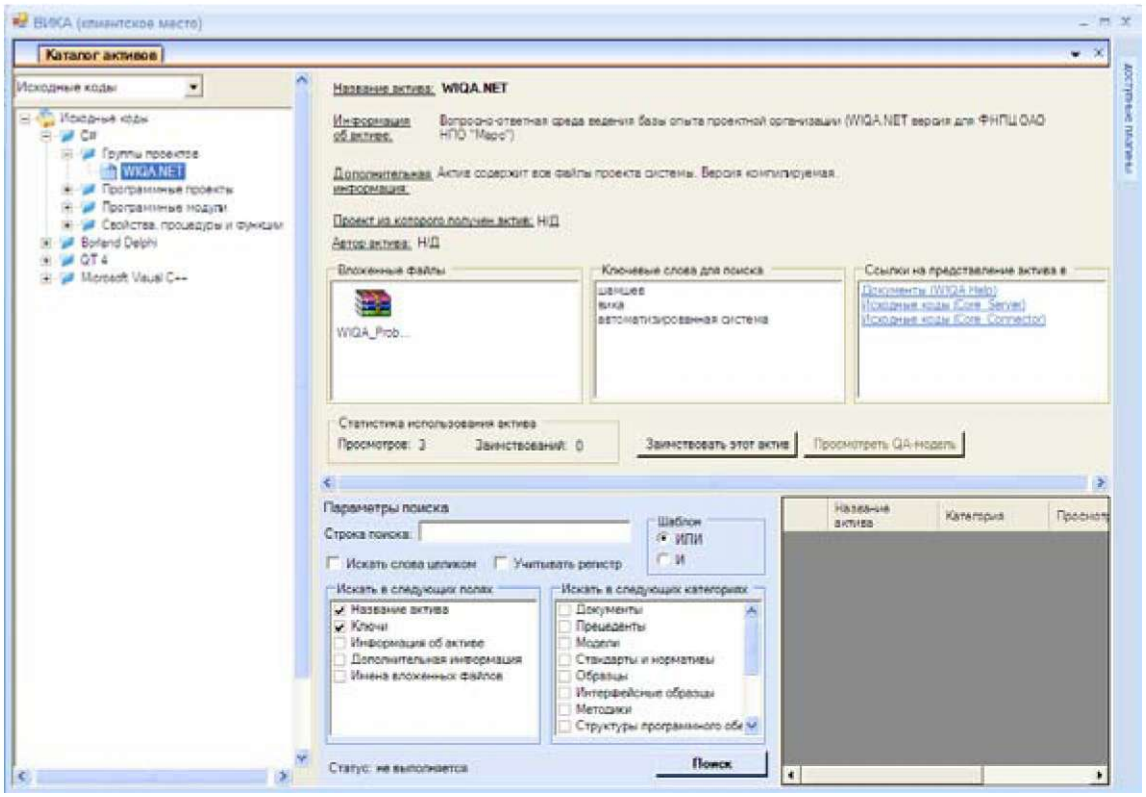

Рис. 3. Операционная обстановка доступа к активам репозитория

• поиск активов по ключам/названию/описанию/категориям;

• заимствование актива в свой проект.

Визуализация каталогов в виде иерархического дерева предполагает загрузку данных активов и выстраивание их в соответствии с отношениями вложенности.

Графическая визуализация активов предполагает загрузку данных выбранного актива и отображение информации в виде блоков текста, списков, иконических изображений (вложенные файлы), деревьев (вопросно-ответная модель).

Поиск активов предполагает добавление в список тех активов, которые удовлетворяют условиям поиска, и отображение этого списка в специальном окне для дальнейшего просмотра каждого из найденных активов.

Заимствование предполагает добавление в проект пользователя информации, которая содержится в заимствуемом активе в качестве особой структуры (вопросно-ответное дерево).

Операционная обстановка, в которой пользователь взаимодействует с «Каталогом активов», представлена на рисунке 3.

В левой части интерфейсной формы отображается структура каталогов выбранной категории. При смене категории в дерево загружается соответствующая ей структура каталогов. В виде папок в дереве отображаются каталоги, в виде файлов - активы, содержащиеся в этих каталогах. При выборе какого-либо актива в основную форму загружается информация об этом активе:

• название актива;

• информация об активе;

• проект, из которого получен актив;

• автор актива;

• вложенные файлы - отображаются в виде значка, соответствующего зарегистрированному в системе расширению файла;

• ключевые слова для поиска - показывается список ключевых слов, по которым ведется поиск активов;

• ссылки на представление актива в других типах активов - отображается список ссылок на другие активы, для перехода к которым необходимо выполнить двойное нажатие клавишей мыши на интересующий пользователя актив;

• статистика использования актива - показывается количество просмотров и заимствований выбранного актива;

• заимствование этого актива - вызывается форма заимствования актива в проект пользователя;

• просмотр QA-модели - вызывается форма, отображающая вопросно-ответное дерево, соответствующее модели, вложенной в выбранный актив.

ЗАКЛЮЧЕНИЕ

Разработан и представлен комплекс средств для извлечения активов (для их повторных применений) из исходных кодов программных продуктов. Утилиты, обеспечивающие извлечение активов, запрограммированы на специализированном псевдо-коде так, что их легко модифицировать и приспосабливать для дополнительных алгоритмических языков. Каталогизация активов унифицирована и открыта для доступа в удобной форме.

СПИСОК ЛИТЕРАТУРЫ

1. Маклаев В.А. Подход к представлению и оперативному использованию профессиональных активов проектной организации // Автоматизация процессов управления. - 2011. - №1(23). - С. 5-12.

2. Соснин П.И. Вопросно-ответное моделирование в разработке автоматизированных систем. -Ульяновск: УлГТУ, 2007. - 333 с.

3. Basili A. V., M. Lindvall M. and Costa P. Implementing the experience factory concepts as a set of experience bases, In Proc. of the 13 th International Conference on Software Engineering & Knowledge Engineering, (2001), 102-109.

4. Henninger S. Tool Support for Experience-based Software Development Methodologies, Advances in Computers, 59, (2003), 29-82.

5. Sosnin P. Question-Answer Approach to Human-Computer Interaction in Collaborative Designing. Chapter in the book "Cognitively Informed Intelligent Interfaces: Systems Design and Development" Published IGI Global, (2012), pp. 157-176.