

УДК 004.02

ВОССТАНОВЛЕНИЕ ИНФОРМАЦИОННОЙ МОДЕЛИ НАСЛЕДОВАННОЙ СЛОЖНОЙ ТЕХНИЧЕСКОЙ СИСТЕМЫ

Калуцкий Никита Сергеевич

*Инженер отдела прочности;
ООО «Прогрестех-Дубна»;
141980, Московская обл., г. Дубна, ул. Программистов, 4;
e-mail: nikita.kalutsky@progresstech.ru.*

В данной статье предлагается методика работы с наследованными системами. Под термином наследованная система подразумевается такая сложная техническая система (автомобиль, самолет, корабль и т.д.), время жизни которой намного превосходит время жизни программных систем, с помощью которых сложная техническая система создавалась. В таком случае говорят о проблеме наследованных систем. Для решения этой проблемы в статье предлагаются методы и алгоритмы, на основе которых был разработан программный комплекс.

Ключевые слова: наследованная система, информационная модель, программный комплекс.

DATA MODEL RECOVERY OF HERITAGE COMPLEX TECHNICAL SYSTEM

Kalutsky Nikita

*Engineer of stress department;
LLC «Progresstech-Dubna»;
141980, Dubna, Moscow reg., Programmistov str., 4;
e-mail: nikita.kalutsky@progresstech.ru.*

In this article a method of handling heritage systems is proposed. The term «heritage system» stands by a complex technical system (i.e. car, airplane, ship, etc.) which lifetime is significantly greater than that of software packages which were used for creation of this technical system. In this case a problem of heritage systems appears. To solve this problem new software application was developed based on methods and algorithms described in this article.

Keywords: heritage system, data model, software application.

Введение

Идея данной разработки родилась во время работы группы инженеров компании «Прогрестех-Дубна» над проектом, заказчиком в котором являлся крупный авиационный производитель – концерн AIRBUS.

В начале 2012 года представитель концерна AIRBUS в России обратился в компанию «Прогрестех-Дубна» с просьбой выполнить ряд расчетов на прочность новой модификации семейства пассажирских самолетов A320. Сами расчёты не представляют особой сложности и давно запрограммированы в виде библиотек языка FORTRAN. Однако сбор геометрических и других инженерных данных о рассчитываемых деталях (например, материал и толщина обшивки, код термообработки, литеры ревизии, номер вышестоящей сборки и т.д.) представляет существенную сложность по причинам:

- а) ограниченного доступа к PDM системам AIRBUS;
- б) отсутствие нужных данных в PDM системе AIRBUS для старых моделей самолетов.

В итоге инженерам приходилось терять много времени на ручной поиск слабо структурированной информации по библиотеке отсканированных чертежей и спецификаций. Это рутинная и кропотливая работа, которая отнимает много времени и быстро утомляет.

Какое-то время спустя появилось предложение о том, как можно формализовать и затем автоматизировать поиск нужных для расчетов данных.

Решение, которое было предложено можно охарактеризовать как «PDM-наоборот». В отличие от «прямого» PDM, когда мы имеем цифровую модель сложной технической системы (самолета) и, делая к ней запросы, получаем нужную информацию, мы поступаем наоборот: из имеющейся разрозненной информации воссоздаем модель сложной технической системы, а уже воссоздав формализованную модель (пусть и далеко не полностью, но в нужных нам пределах) мы получаем инструмент, которым удобно пользоваться для получения необходимой инженерной информации. При этом достигаются следующие преимущества:

- интеграция с существующими расчётными модулями и средствами автоматизации;
- более высокий уровень автоматизации процесса создания конечного отчёта (интеграция с MS Office);
- отсутствие потерь времени и ошибок ручного поиска.

Недостатки подхода заключаются в изолированности данных, т.к. воссозданная информационная модель не всегда позволяет динамически отслеживать изменения оригинала.

Технически воссоздание формализованной модели достигается через синтаксический разбор имеющихся текстовых документов (спецификаций) и сохранение отобранной информации в базе данных.

Основные понятия

Современные условия, в которых находится практически любое производство, – это постоянное и значительное усложнение инженерно-технических, проектов, программ разработки новой продукции и рост наукоемкости изделий. В таких условиях конкурентоспособными оказываются предприятия, обладающие отлаженными процессами проектирования, производства, поставки и поддержки изделий.

В последние годы произошел качественный скачок в эволюции компьютерных комплексов управления жизненным циклом сложных технических систем. На сегодня весь жизненный путь изделия от его разработки, серийного производства, до поддержки эксплуатации и утилизации формализован в программных пакетах, получивших название PDM-систем (от англ. Product Data Management – система управления данными об изделии).

Однако часто встречается обратная ситуация, когда изделие, разработка которого была выполнена до появления имеющейся PDM-системы, не описано в ней вообще или описано частично. В данном случае можно говорить о проблеме наследованных систем.

Математическая модель изделия

Для восстановления информационного образа технической системы требуется разработать математическую модель изделия и выбрать метод представления данных об изделии, на основании которого будет проходить восстановление информационного образа системы.

Поскольку любое изделие имеет древовидную структуру, и алгоритмы работы с изделиями предполагают обход этого дерева, то для обеспечения эффективности алгоритмов в программном комплексе опишем модель в терминах теории графов.

Представим изделие в CAD- или PDM-системе в виде связного графа, не содержащего циклов (N -арного дерева) [5].

Обозначим иерархическую структуру изделия, содержащую версии составных частей и характеристики этих частей, через граф $G = (X, A)$ (обозначения по работе Кристофидеса [4]):

- корень дерева – X_0 – используется для обозначения версии изделия-сборки;
- остальные узлы – $X_1 \dots X_n$, каждый из которых является деревом используются для обозначения компонентов сборки (деталей и подборок);
- ребра – A_{ij} – используются для обозначения отношения сборки (X_i) – элемент сборки (X_j).

Используем также ряд дополнительных понятий:

- степень узла $X_i - S_{X_i}$ – количество поддеревьев узла, то есть количество подизделий следующего уровня, входящих в данное изделие;
- уровень узла $X_i - L_{X_i}$ – определяется рекурсивно: уровень корня дерева G равен нулю; уровень любого другого узла на единицу выше, чем уровень корня ближайшего поддерева дерева G , содержащего данный узел;
- максимальный уровень узла – L_{max} ;
- лист – узел со степенью, равной нулю.

Для примера рассмотрим упрощенное представление в виде графа секции фюзеляжа пассажирского самолета, показанное на Рис. 1.

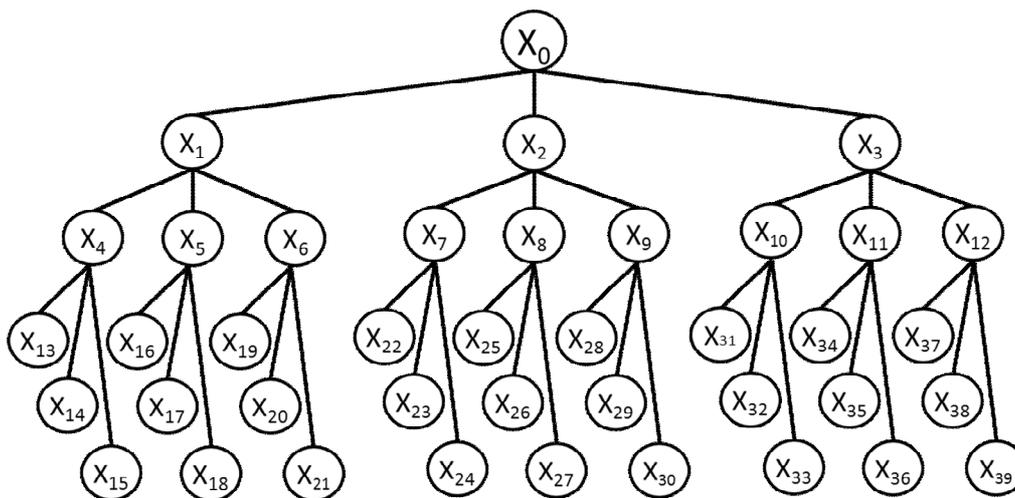


Рис. 1. Иерархическая структура изделия

Обозначим основные составные части секции и их характеристики через X_i .

Итак, секцию фюзеляжа X_0 состоит из следующих составных частей:

- панель верхняя – X_1 ;
- панель боковая – X_2 ;
- панель нижняя – X_3 .

Панель верхняя X_1 состоит из следующих элементов:

- сборка обшивки – X_4 ;
- продольный силовой набор – X_5 ;
- поперечный силовой набор – X_6 .

Сборка обшивки X_4 состоит из следующих элементов:

- обшивка – X_{13} ;
- лента стыковая – X_{14} ;
- усиление выреза двери – X_{15} .

Продольный силовой набор X_5 состоит из следующих элементов:

- стрингеры – X_{16} ;
- стыковые профили – X_{17} ;
- уплотнения – X_{18} .

Поперечный силовой набор X_6 состоит из следующих элементов:

- шпангоуты – X_{19} ;
- стыковые накладки – X_{20} ;
- фитинги – X_{21} .

Методы и алгоритмы

Разработанная модель изделия позволяет хранить все необходимые данные об изделии в формализованном виде. Предлагаемый программный комплекс приводит имеющиеся разрозненные данные об изделии к требуемому формализованному виду. Общий алгоритм комплекса приведен на рис. 2.

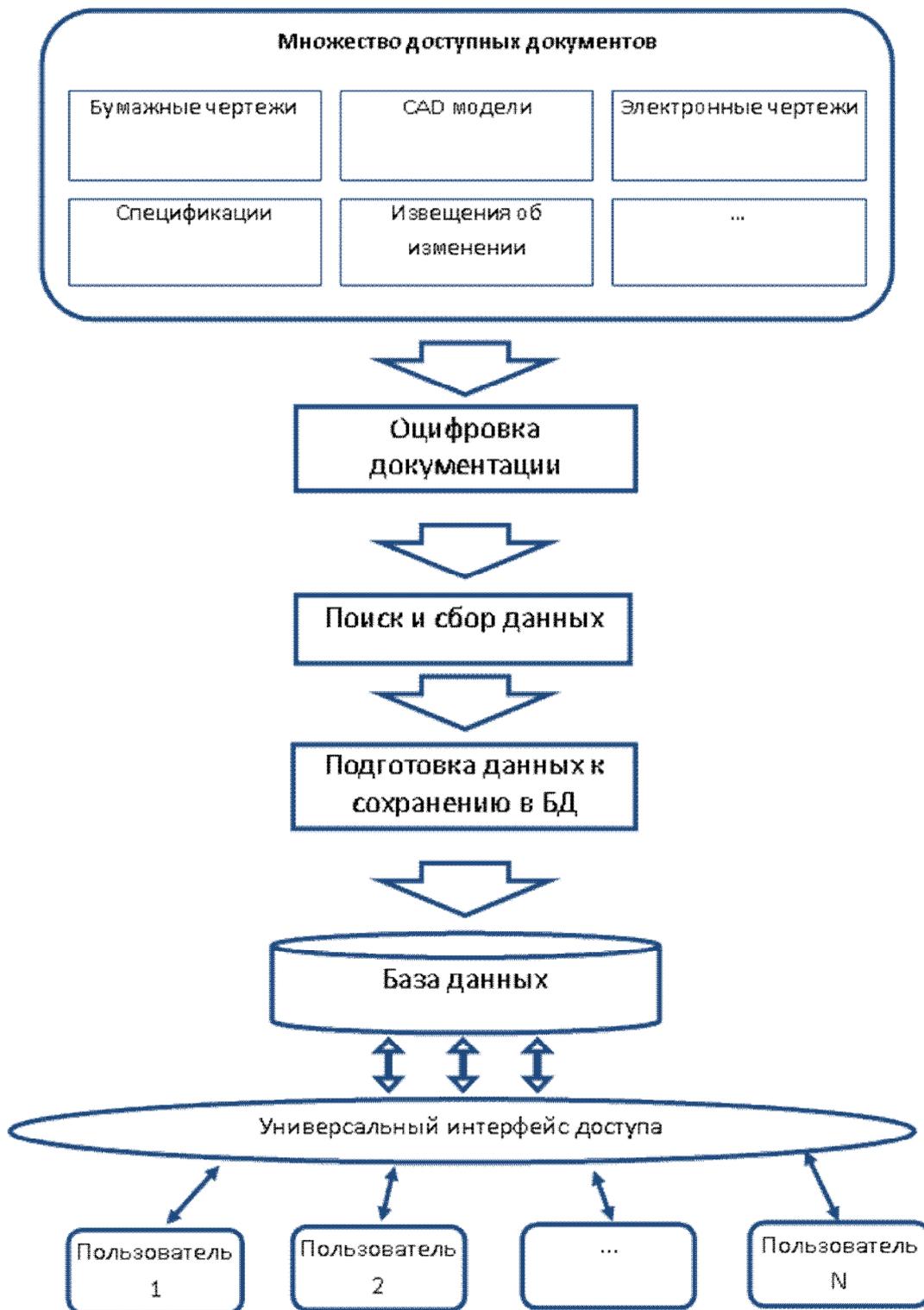


Рис. 2. Схема комплекса восстановления информационной модели

Как видно из рисунка, описываемый комплекс представляет собой процесс наполнения базы данных структурированной информацией об изделии с одной стороны и обеспечение пользовательского доступа к этой информации через стандартизованный интерфейс доступа – с другой стороны.

Для наполнения базы данных описываемого комплекса, требуется провести анализ и синтаксический разбор имеющихся в распоряжении документов, описывающих наследованную систему, для извлечения из них нужных данных. Для этих целей был разработан модуль добычи данных.

Алгоритм работы модуля добычи данных приведен на Рис. 3.

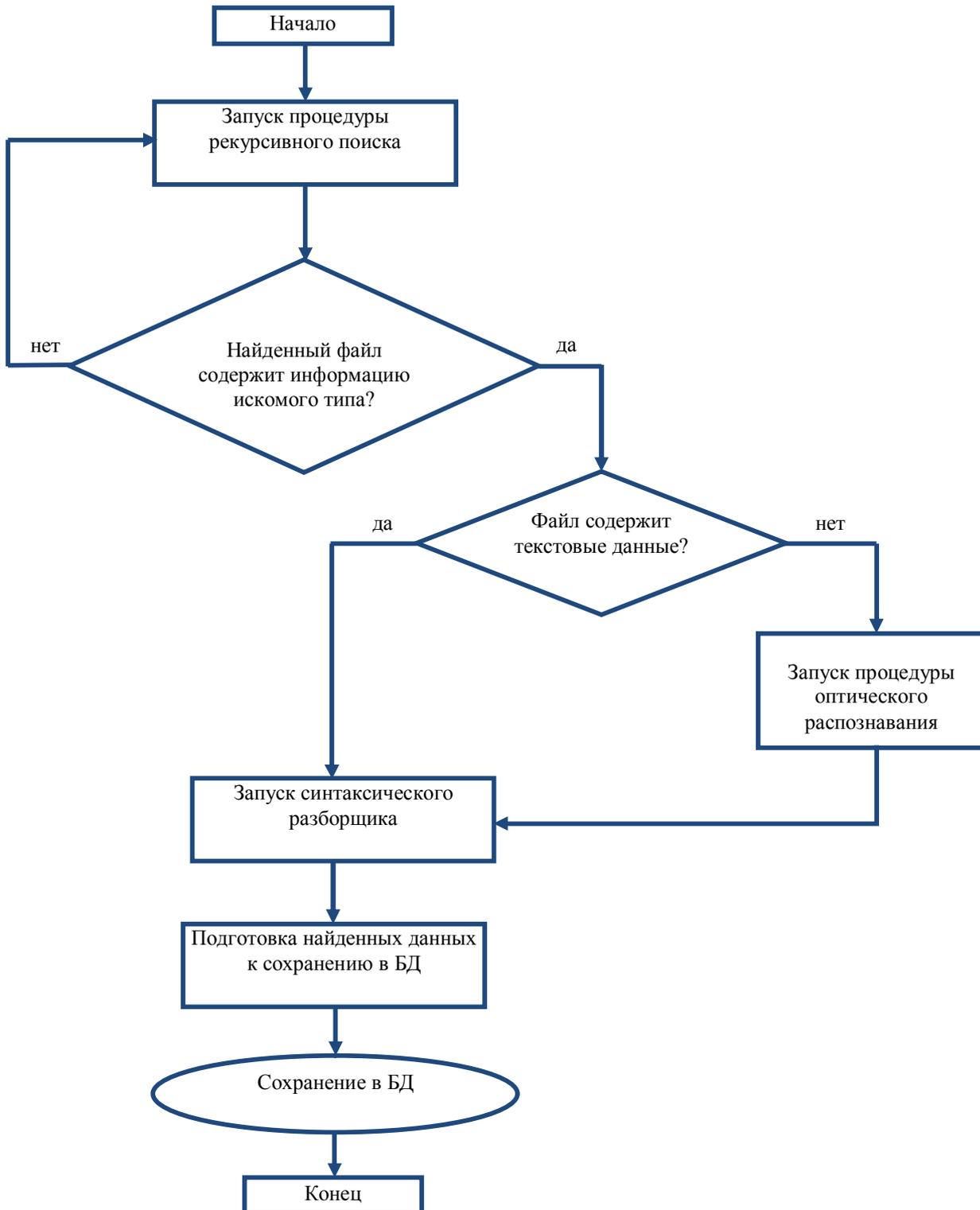


Рис. 3. Алгоритм работы модуля добычи данных

Практическая реализация

Поскольку формат статьи не позволяет привести подробное описание разработанных программных модулей, приведем лишь принципиальные положения.

В качестве языка программирования был выбран C# [6].

В качестве системы управления базой данных был выбран Microsoft SQL Server Express [2].

В качестве основы интерфейсного компонента, реализующего взаимодействие конечного пользователя и разработанной системы, в состав системы включена библиотека Microsoft ActiveX Data Objects 6.0 Library (сокр. ADO) [1]. Основное назначение модуля ADO – обеспечить программистам-разработчикам прикладных систем удобный доступ к данным, хранящимся в какой-либо БД. Получив доступ, прикладная система может отсылать запросы на нужную информацию. Данные, хранящиеся в восстановленной информационной модели, становятся, таким образом, доступными многим популярным приложениям. А приложения, в свою очередь, пополняются мощным инструментарием работы с данными.

Рис. 4 схематически показывает организацию взаимодействия клиентского приложения с восстановленной моделью изделия через интерфейсный компонент.

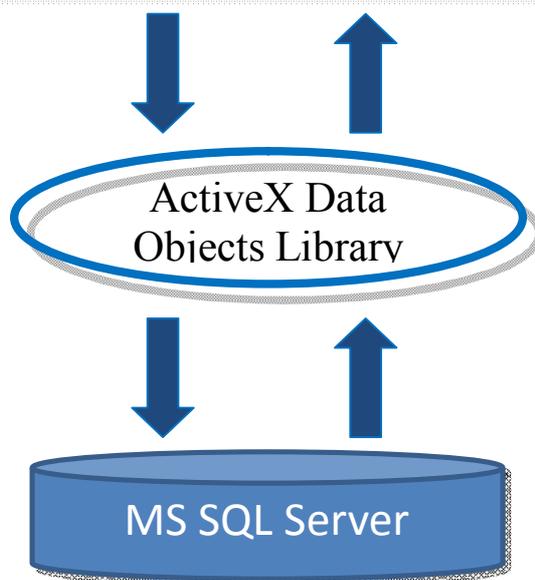
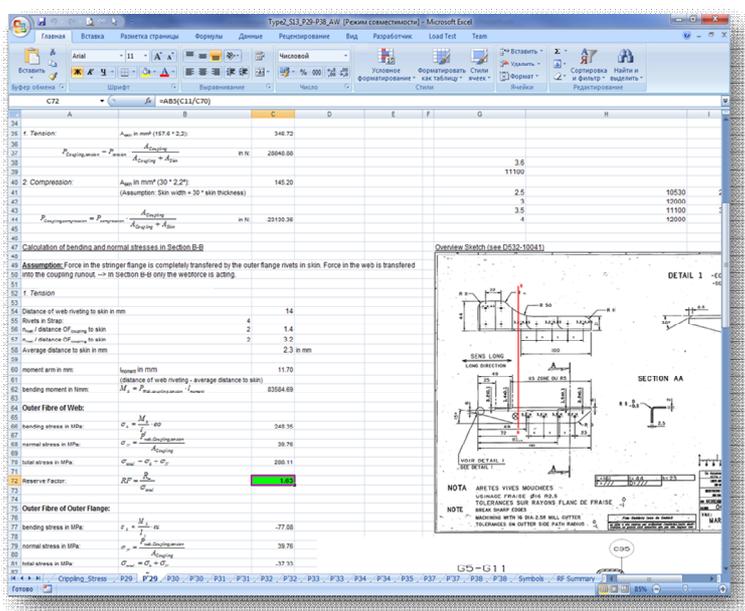


Рис. 4. Пример взаимодействия MS Excel с восстановленной моделью изделия

Заключение

Опытная эксплуатация программного комплекса с предложенной архитектурой проводилась во время работы группы инженеров «Прогрестех-Дубна» над прочностными расчетами семейства пассажирских самолётов AIRBUS A320.

Результат эксплуатации позволяет говорить о следующих достоинствах предлагаемого метода:

- существенный выигрыш по затрачиваемому времени на поиск нужных данных (геометрических размеров деталей, материалов и т.д.), исчисляемый десятками часов в месяц;
- удобство работы по сравнению с «ручным» методом поиска в слабоструктурированном массиве файлов и папок;
- надежность получаемых данных и простота последующей проверки;
- исключение дублирования усилий по поиску нужного документа;
- высокая степень универсальности предлагаемого решения, его применимость для решения многих задач.

Недостатки предложенной архитектуры:

- комплекс работает с отсоединенными (изолированными) данными;
- данные обновляются с запозданием;
- риск хранения и получения устаревшей информации.

Описанные недостатки могут быть критичны для динамически развивающихся систем, то есть изделий, находящихся в активной фазе проектирования и разработки. Однако, как говорилось в начале статьи, новые разработки, как правило, ведутся под управлением какой либо PDM-системы, поэтому прибегать к предложенному методу не имеет смысла. В то же время для наследованных систем, где скорость изменений невелика или близка к нулю, этот недостаток не является существенным.

Список литературы

1. Microsoft Corporation. Microsoft ActiveX Data Objects // [MSDN] – [Электронный ресурс]. URL: [http://msdn.microsoft.com/en-us/library/windows/desktop/ms675532\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms675532(v=vs.85).aspx) (дата обращения: 10/10/2012).
2. Вийера Р. Программирование баз данных Microsoft SQL Server 2005 для профессионалов / пер. Птицын К.А. – М.: Вильямс, 2008 – С. 1072.
3. Долидзе С. Вопросы интеграции средств информационного обеспечения на этапе эксплуатации и послепродажного обслуживания самолетов / С.Долидзе, В.Климов, В.Клишин, А.Недер // САПР и графика. – 2007. – №6. – С. 78-82.
4. Кристофидес Н. Теория графов. Алгоритмический подход / Н.Кристофидес. – М.: Мир, 1978. – С. 432.
5. Оре О. Теория графов / О. Оре. – 2-е изд. – М.: Наука, 1980. – С. 336.
6. Шилдт Г. C# 4.0. Полное руководство / пер. Берштейн И.В. – М.: Вильямс, 2007.