

РАЗРАБОТКА БАЗ ДАННЫХ ИНТЕЛЛЕКТУАЛЬНОЙ ИНТЕГРИРОВАННОЙ СРЕДЫ ПРОЕКТИРОВАНИЯ И ОБУЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ НОТАЦИИ ER

С.И. Бригаднов¹⁵, Д.С. Канев, Е.Ю. Воеводин,
В.С. Хородов, Р.Ф. Гайнуллин

В статье рассматриваются вопросы по исследованию и разработки автоматизированного комплекса анализа проектных решений и обучения проектировщика средствами САПР КОМПАС. Предложена обобщенная архитектура комплекса, описано его функционирование в виде UML-диаграмм.

Введение

Инновационные процессы в технологической сфере требуют постоянного обновления машиностроительного парка. Это диктует необходимость быстрого принятия ответственных проектных решений, в которых должны быть исключены проектные неточности и ошибки. Несмотря на разнородность изделий машиностроения, они приемлемым образом классифицируются как собственно на уровне сборок, так и на уровне узлов и деталей, что является предпосылкой для типизации приемов проектирования, в т.ч. и при разработке конструкторской документации посредством 3D-САПР. В рамках исследования и разработки автоматизированного комплекса анализа проектных решений и обучения проектировщика выбран САПР КОМПАС, который является лучшим CAD-продуктом среди отечественных инструментов такого рода.

Интеллектуализация процессов проектирования должна базироваться не только на создании электронных моделей параметризованных узлов и деталей, но и на базах знаний по типизированным и оптимизированным приемам проектирования. Последний тезис прямо приводит к идее создания обучающих программ по эффективным приемам работы в 3D-САПР для повышения компетенций проектировщика [1, 2].

Авторами разработана и предложена обобщенная схема автоматизированного комплекса анализа проектных решений и обучения проектировщика средствами САПР КОМПАС (рисунок 1).

¹⁵ 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: sergbrig@yandex.ru

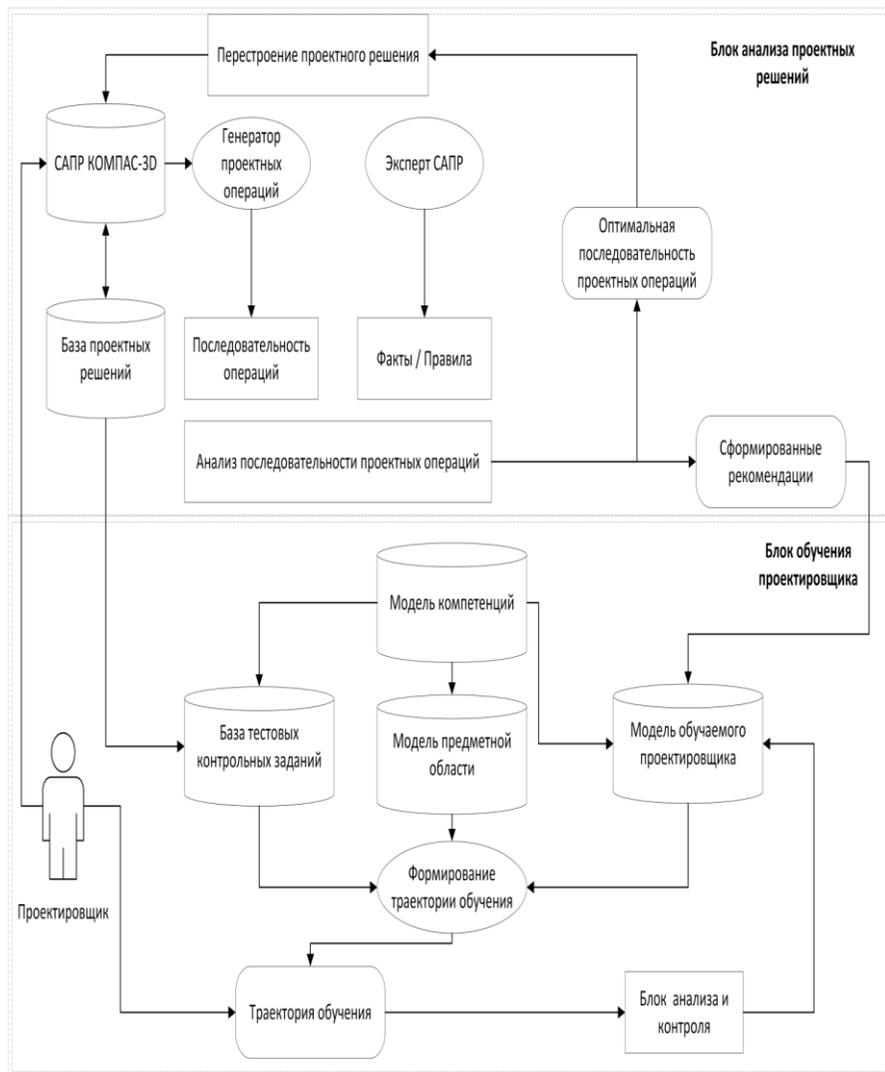


Рис. 1. Схема автоматизированной системы анализа проектных решений и обучения проектировщика.

Основные компоненты системы, блоки, а также функционирование комплекса более подробно изложено в работах [3, 4].

Разработка компонентной архитектуры автоматизированной системы анализа проектных решений и обучения

На рисунке 2 показана uml-диаграмма деятельности в разработанной автоматизированной обучающей системе (АОС). Были выделены 3 области диаграммы деятельности («дорожки»), которые отображают только те действия, за который отвечает определенный объект. Служат они для разбиения диаграммы деятельности в соответствии с распределением ответственности за действия. Имя дорожки, чаще всего, означает роль или объект, которому она соответствует (обучаемый, АОС, преподаватель).

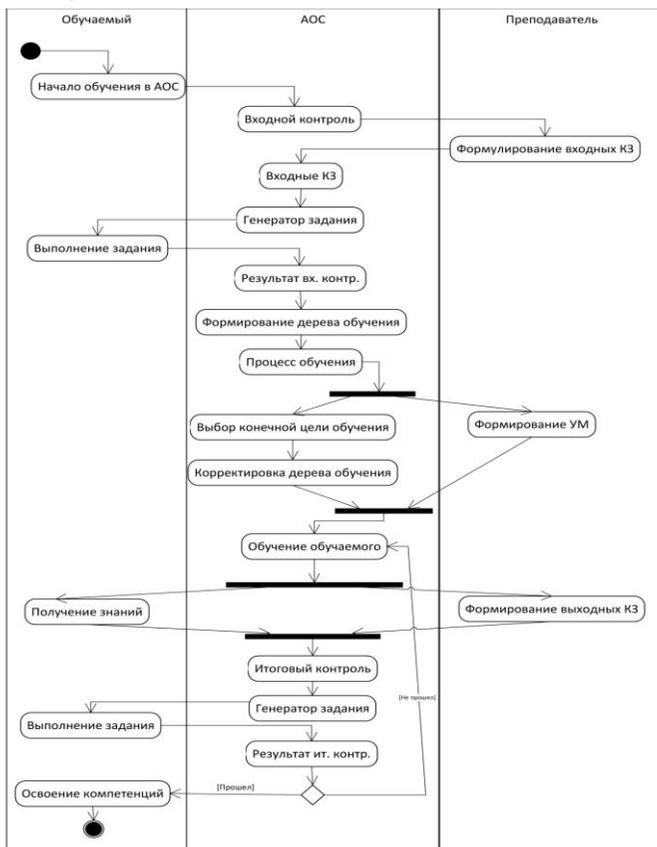


Рис. 2. UML диаграмма деятельности на основе концептуальной модели АОС

Построенная диаграмма позволяет реализовывать особенности процедурного и синхронного управления, обусловленного завершением внутренних деятельностей и действий. Диаграмма представляет собой совокупность отдельных действий и вычислений, приводящих к конечному результату. Результат деятельности может привести к корректировке состояния автоматизированной обучающей системы и возвращению определенного значения или набора значений (оценка за прохождение итогового контроля, промежуточного контроля, входного контроля). Для синхронизации параллельных процессов использовался переход «разделение - слияние».

На рисунке 3 показана uml-диаграмма вариантов использования в разработанной автоматизированной обучающей системе (АОС). Вариант использования (use case) — это внешняя спецификация последовательности действий, которые автоматизированная обучающая система может выполнять в процессе взаимодействия с действующими лицами (обучаемый, преподаватель).

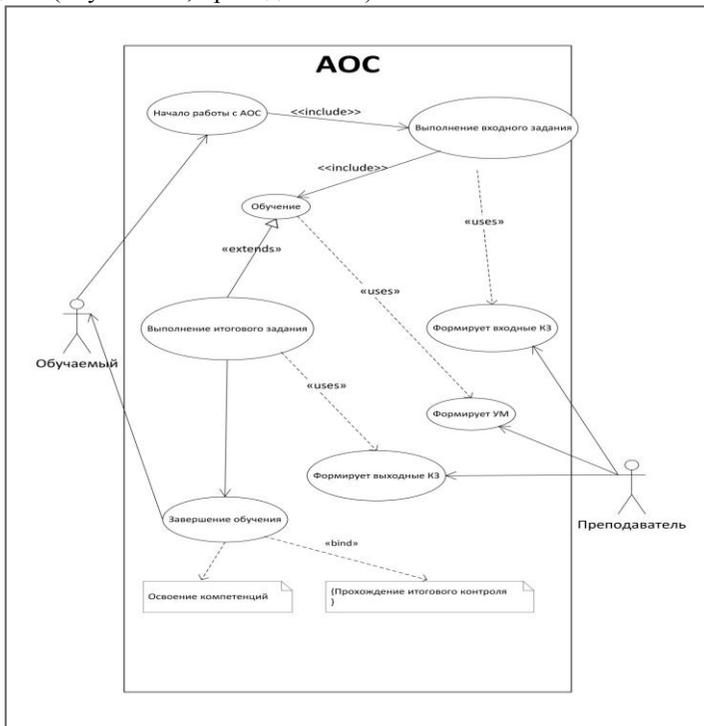


Рис. 3. UML диаграмма вариантов использования на основе концептуальной модели АОС

Вариант использования отображает спецификацию общих особенностей поведения или функционирования моделируемой автоматизированной обучающей системы без детального рассмотрения внутренней структуры этой системы. Каждый вариант использования отображается на диаграмме эллипсом, внутри которого находится его краткое имя в форме существительного или глагола с пояснительными словами. Актер представляет собой любую внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач. Каждый актер может рассматриваться как некая отдельная роль относительно конкретного варианта использования.

На рисунке 4 показана uml-диаграмма последовательности в разработанной автоматизированной обучающей системе (АОС). Объекты на диаграмме обозначаются в виде прямоугольников с подчеркнутыми именами, сообщения (вызовы методов) отображаются в виде линий со стрелками, возвращаемые результаты обозначаются на диаграмме как пунктирные линии со стрелками. Прямоугольники на вертикальных линиях под каждым из объектов показывают "линию жизни" (фокус) объектов.

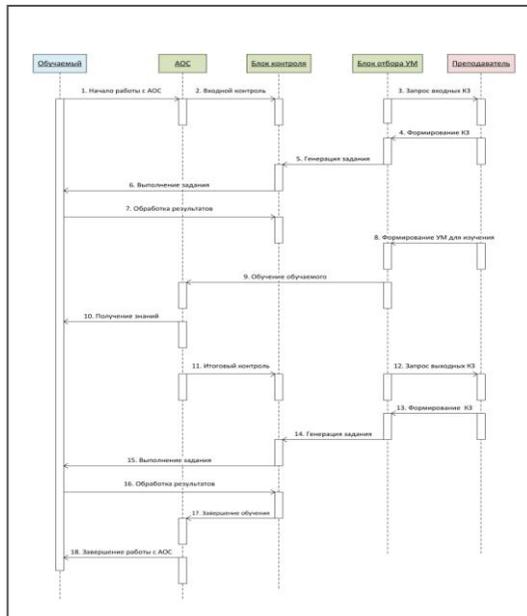


Рис. 4. UML Диаграмма последовательности на основе концептуальной модели АОС

Разработка базы данных среды проектирования и обучения

Для удобства хранения информации о проектных решениях машиностроительных объектов была разработана база данных, включающая классификацию изделий и перечень их параметров [5, 6].

Хранение данных осуществляется в базе данных LiteDB, представляющую собой легковесную и быстродействующую документно-ориентированную базу данных, предназначенную для .NET. LiteDB не требует внешних серверов базы данных и хранит все данные в переносимом файле базы данных. В качестве данных она поддерживает обычные классы C#. LiteDB поддерживает файловое хранилище для хранения файлов. Кроме того, поддерживаются индексы для более быстрого доступа к данным.

LiteDB является безсерверной базой данных, поставляемой в одной DLL (менее 350 кб), полностью написанной в управляемом коде .NET C# (совместимом с .NET 3.5, 4.x, NETStandard 1.3 и 2.0).

LiteDB хранит данные в виде документов, которые являются JSON-подобными полями и парами значений. Документы представляют собой структуру данных без схемы. Каждый документ содержит пользовательские данные и структуру. LiteDB хранит документы в формате данных BSON (Binary JSON). BSON представляет собой двоичное представление JSON с дополнительной информацией о типе. В документах значение поля может быть любым из типов данных BSON, включая другие документы, массивы и массивы документов. BSON - это быстрый и простой способ сериализации документов в двоичном формате.

LiteDB организует документы внутри хранилищ, известных как коллекции. Каждая коллекция идентифицируется уникальным именем и содержит один или несколько документов, которые имеют одну и ту же схему. Для работы с документами используются методы коллекции, приведенные ниже.

- Insert - используется для добавления нового документа в коллекцию.
- Update - используется для обновления существующего документа.
- Delete - используется для удаления документа.
- FindById или Find - используется для запроса документа.
- Include - используется для заполнения свойств из других коллекций.
- EnsureIndex - используется для создания нового индекса, если он не существует.

LiteDB предоставляет FileStorage коллекцию для работы с файлами. Выделяют две специальные коллекции для разделения содержимого файла:

- `_files` коллекция хранит только ссылки на файлы и метаданные,

- `_chunks` коллекция хранит двоичные данные в кусках 1 МБ.

Файлы идентифицируются `_id` строковым значением со следующими правилами:

- начинается с буквы, числа, `_`, `-`, `$`, `@`, `!`, `+`, `%`;
- если содержит `/`, должна быть последовательность с символами.

`FileStorage` коллекция содержит следующие методы для работы с хранилищем базы данных `LiteDB`.

- `Upload`: отправка файла или потока в базу данных. Может использоваться с файлом или `Stream`. Если файл уже существует, содержимое файла перезаписывается.
- `Download`: получить файл из базы данных и скопировать в `Stream` параметр.
- `Delete`: удалить ссылку на файл и все фрагменты данных.
- `Find`: найти один или несколько файлов в `_files` коллекции. Возвращает `LiteFileInfo` класс, который может быть загружен после.
- `SetMetadata`: обновление метаданных хранящихся файлов. Этот метод не изменяет значение сохраненного файла. Он обновляет значение `_files.metadata`.
- `OpenRead`: найти файл `_id` и вернуть `LiteFileStream` для чтения содержимого файла в виде потока.

Запись документов в базу данных.

Для начала необходимо создать класс `POCO`, который используется для создания строго типизированного документа.

```
public class DBObject {
    public int Id { get; set; } – уникальный идентификатор документа;
    public string Type { get; set; } – тип изделия;
    public string FileName { get; set; } – название файла с изделием;
    public string Material { get; set; } – материал изготовления изделия;
    public string[][] Parameters { get; set; } – массив параметров
    трехмерной модели;
}
```

Для каждого записываемого в базу данных документа создается экземпляр класса `DBObject`.

```
var detail = new DBObject {
    Type = dtype,
    FileName = name_file,
    Material = material,
    Parameters = new string[numpart][]
};
```

В соответствии с типом изделия выбирается таблица (коллекция) для записи.

```

var details = db.GetCollection<DBObject>("Rings");
var parameters = db.GetCollection<DetailsParameter>("RingsParameter");

```

В выбранную таблицу данные заносятся методом Insert. Трехмерная модель изделия, выполненного в САПР КОМПАС-3D, записывается в файловое хранилище FileStorage с помощью метода Upload(<путь к файлу>, <название файла>).

Разработанная база данных состоит из множества таблиц, соответствующих выделенным классам изделий («Кольцо», «Гайка», «Шайба» и др.) и списком параметров для каждого класса изделий.

Была разработана ER-диаграмма базы данных (рисунок 5), основные концепции модели «сущность-связь» включают типы сущностей, типы связей и атрибуты. Объектные отношения хранят данные обо всех объектах одного и того же типа, по одному кортежу на объект, и содержат ключ для распознавания объекта. Остальные поля функционально зависят от этого ключа.

Поставленная задача предполагает наличие 8 сущностей: пользователь, модель предметной области, элемент, связь между элементами, модель обучаемого, тестовое задание, вариант ответа, связанный учебный материал предметной области с ответом на тестовое задание.

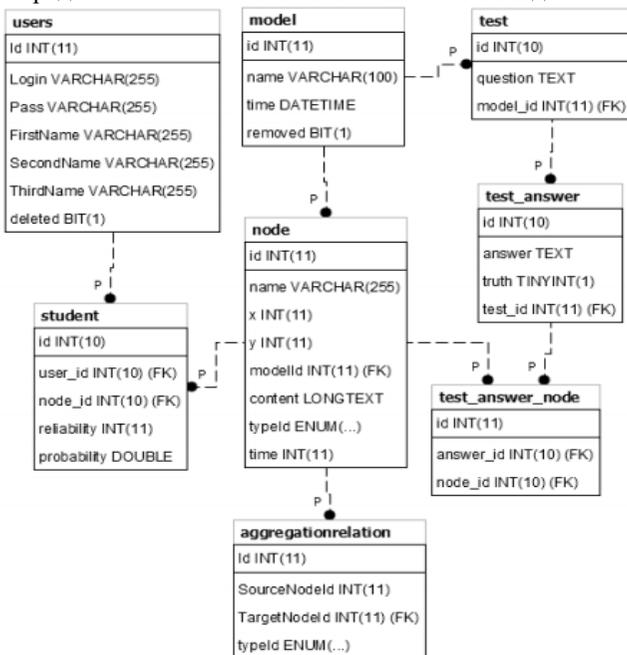


Рис. 5. ER-модель базы данных

Заключение

На примере системы трехмерного проектирования КОМПАС – 3D предлагается упрощенный вариант мониторинга, проверки и поиска проектных решений.

Архитектура системы состоит из следующих частей: база изделий; программа анализа изделий - состоит из различных модулей анализа определенных классов изделий; база данных об изделиях с классификацией изделий и перечнем их параметров; программа поиска изделий в соответствии с условиями, которые задает пользователь (класс изделия, поиск по значению параметра), и выводом результата в виде списка изделий.

Рекомендуется использование разработанного программного комплекса для повышения качества проектных решений, выполненных в САПР КОМПАС-3D, а также повышения эффективности и качества обучения проектировщика изделий машиностроения.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417.

Список литературы

1. Бригаднов С.И. Разработка комплексной автоматизированной интеллектуальной системы анализа проектных решений и обучения проектировщика // V Международная научно-практическая конференция «Электронное обучение в непрерывном образовании 2018», г. Ульяновск. – Ульяновск, 2018. – С. 136-142.
2. Brigadnov S.I. Development of associative-oriented models of competencies and trainees in automated training systems // INTERACTIVE SYSTEMS: Problems of Human - Computer Interaction Collection of scientific papers. 2017. – P. 185-189.
3. Афанасьев, А.Н. Разработка автоматизированной системы анализа проектных решений в САПР КОМПАС-3D [Текст] / А. Н. Афанасьев, С. И. Бригаднов, Д. С. Канев // Автоматизация процессов управления. – 2018. – № 1(51). – С. 108-117.
4. Афанасьев, А.Н. Методы и средства комплексной системы анализа проектных решений и обучения проектировщика [Текст] / А. Н. Афанасьев, С. И. Бригаднов // Автоматизация процессов управления. – 2018. – № 2(52). – С. 50-57.
5. Бригаднов, С.И. Разработка базы проектных решений машиностроительных объектов [Текст] / С. И. Бригаднов, М. Е. Уханова, И. С. Ионова, А. Г. Игонин // Информационно-измерительные и управляющие системы. – 2017. – № 12. – С. 79-85.
6. Бригаднов С.И. Разработка компонентов программного обеспечения интеллектуальной системы проектирования и обучения / В.А Гульшин, Н.Н. Вейт, С.И. Бочков, М.Е. Уханова, Д.С. Канев // V Международная научно-практическая 23 конференция «Электронное обучение в непрерывном образовании 2018», г. Ульяновск. – Ульяновск, 2018. – С. 168-176.