

МОДЕРНИЗАЦИЯ ЯДРА ПТКИ BELSIM¹

А.В. Шайторов, С.А. Альховик

В статье представлена реализация управляющей программы моделирования процессным способом для модернизации ядра ПТКИ BELSIM, которая позволяет повысить его модульность, упростить проектирование и реализацию имитационных моделей

Управляющая программа моделирования, ПТКИ BELSIM

1. ВВЕДЕНИЕ

Широкое применение имитационного моделирования в ходе проектирования и эксплуатации сложных систем делает актуальной задачу создания инструментальных средств построения и исследования имитационных моделей (ИМ) сложных систем (СС).

Существующие системы имитационного моделирования мало эффективны при применении их для создания ИМ сложных многоуровневых иерархических систем. Требуется разработка новых технологий и средств имитации, используя последние достижения в области информационных технологий.

Модернизации ядра ПТКИ BELSIM [1] позволит повысить его модульность, упростить проектирование и реализацию имитационных моделей, свести к минимуму количество ошибок. В результате комплекс будет иметь средства для гибкой разработки компонентов модели, их отладки, простой интерфейс создания схемы проектируемой модели. В итоге устраняются основные недостатки существующего комплекса.

2. РЕАЛИЗАЦИЯ УПРАВЛЯЮЩЕЙ ПРОГРАММЫ МОДЕЛИРОВАНИЯ

UML-диаграмма основных классов разработанной управляющей программы моделирования (УПМ) [2] представлена на *рисунке 1*.

Для пользователя одним из самых важных является класс BaseComponent, который должен быть базовым для всех классов компонентов. Задача его заключается в предоставлении пользователю всех необходимых средств для получения данных из модели, чтение информации со входов компонента, выдачи ее на выходы, а так же для операции ожидания. В методе Run разработчик компонента обязан реализовать алгоритм работы компонента. Метод Wait соответствует оператору ожидания компонентом заданного значения модельного времени. Свойства Input, Output, Control, Parameter используются для организации доступа к входам и выходам компонента. Эти свойства возвращают соответствующий объект-интерфейс, что позволяет обращаться к ним как к массиву.

Связь между компонентами и программой моделирования осуществляется при помощи интерфейса ComponentAdaptor. Это предотвращает доступ компонента к закрытым частям системы.

Входы, входы управления и выходы компонента задаются при помощи структур внутри класса компонента.

¹ Работа выполнена в рамках ХД0663 «Разработка динамической производственно-экономической модели завода органического синтеза»



Рис. 1. Диаграмма основных классов УПМ

Перед описанием каждой из структур необходимо обязательно указать один из атрибутов: `ComponentInputs`, `ComponentControls`, либо `ComponentOutputs`. Для УПМ эти атрибуты служат отличительной чертой этих структур. В классе компонента может быть только по одной структуре в один из данных атрибутов. Это, например, означает, что не может быть две структуры, описывающих входы компонента. Сами структуры служат только для задания входов и выходов компонента и в коде класса использоваться не должны. УПМ по каждому элементу структуры определяет имя входа, либо выхода и его тип. Тип может быть не обязательно атомарным типом, возможно использование более сложных составных типов, таких как классы. Самое главное требование – точное совпадение типов при связывании двух или более компонентов между собой связью. Например, нельзя связать выход с типом `Int32` со входом с типом `Double`. Каких либо ограничений на передаваемые типы данных нет.

Основная часть логики работы УПМ заключена в классе `Component`, который является внутренним классом и в явном виде недоступен пользователю. Класс содержит методы для обеспечения низкоуровневого взаимодействия компонентов и синхронизации в модельном времени.

Создание модели, хранение всех компонентов, управление процессом моделирования реализуется классом `Model`.

Класс `ModelData` является типизированным набором данных (typed dataset) и предназначен для чтения и хранения данных о схеме модели.

Ядро системы представляет собой сборку (DLL файл), поэтому его использование возможно отдельно от всей остальной системы. Что позволяет создавать программные продукты на базе ядра, без использования дополнительных средств как Visual Studio или пакета STATISTICA.

Входные и выходные данные системы показаны на *рисунке 2*.

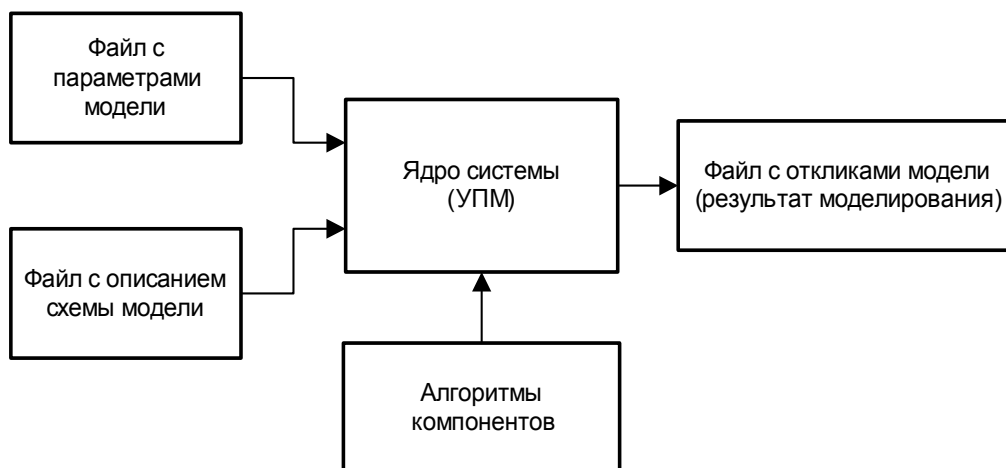


Рис. 2. Входные и выходные данные УПМ

Входные данные ядра включают в себя:

- описанием схемы модели;
- алгоритмы компонентов;
- параметры моделирования.

Выходные данные – это данные, собранные в процессе моделирования. Обычно данные выводятся из системы специально предназначенным для этого компонентом, либо той функцией, которая запускает процесс моделирования. Алгоритм Работы таких

компонентов заключается в записи получаемых данных в файл или буфер памяти, для того чтобы по окончании моделирования эти их можно было бы преобразовать в нужный формат и выдать в виде результата моделирования на устройство вывода.

ER-диаграмма используемой модели данных приведена на *рисунке 3*.

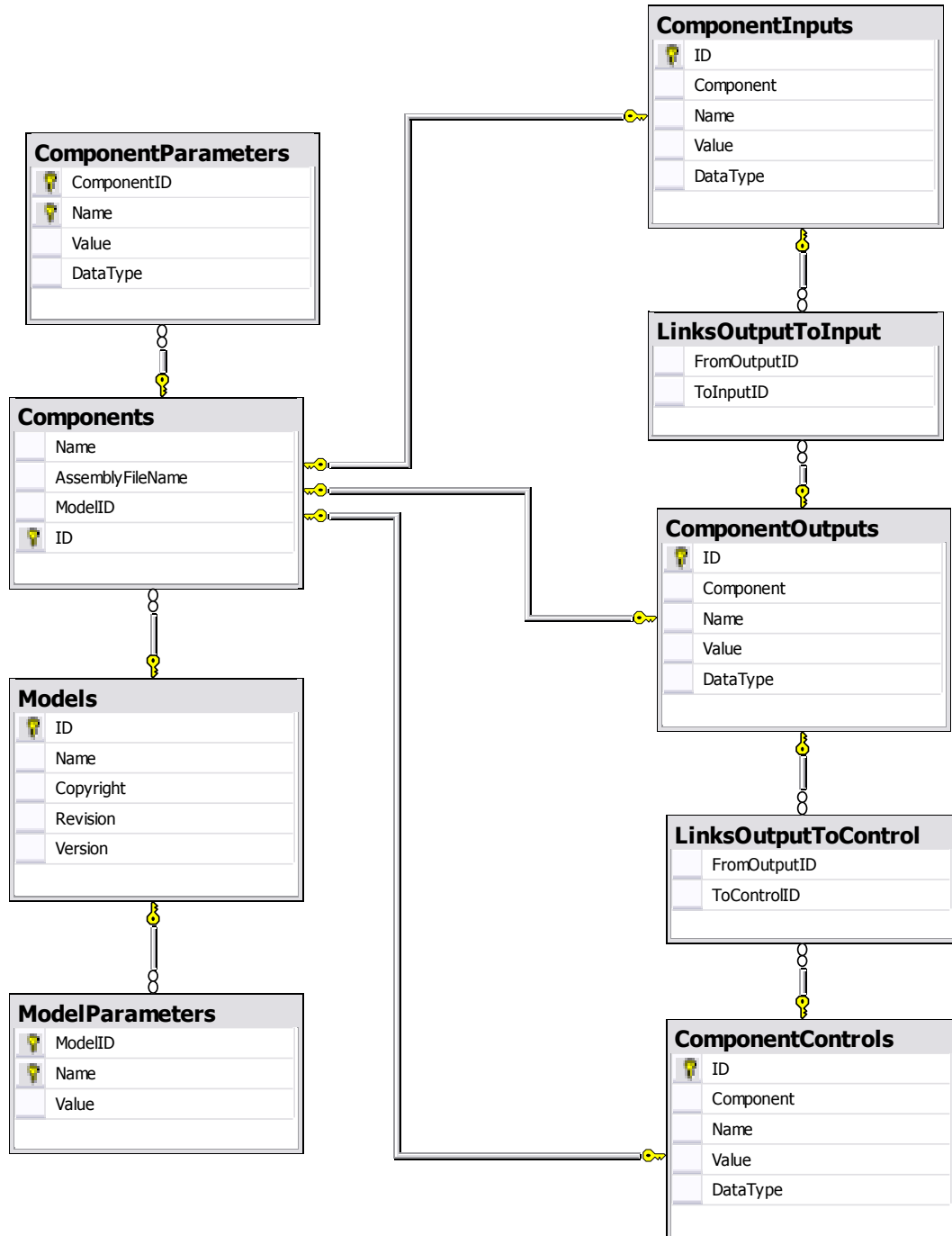


Рис. 3. ER диаграмма модели данных

Сущности:

- Models – имитационные модели;
- Components – компоненты модели;
- ComponentParameters – параметры компонентов;
- ModelParameters – параметры модели;

- LinksOutputToInput – связь между выходом одного компонента и входом другого;
- LinksOutputToControl – связь между выходом одного компонента и управляющим входом другого.

3 ЗАКЛЮЧЕНИЕ

В результате выполнения работы разработана новая технология имитации процессным способом [3], а также программное обеспечение ядра системы имитационного моделирования, на основе последних достижений в области информационных технологий.

Разработанная управляющая программа моделирования и технология ее применения позволяет существенно расширить возможности ПТКИ BELSIM для построения и исследовании имитационных моделей сложных многоуровневых иерархических систем.

Литература

1. *Якимов А.И., Альховик С.А.* Имитационное моделирование в ERP-системах управления. – Мн.: Бел. наука, 2005. – 197 с.: ил.
2. *Максимей И.В.* Имитационное моделирование на ЭВМ. – М.: Радио и связь, 1988. – 232с.: ил.
3. *Шайторов А.В.* Технология имитационного моделирования сложных систем процессным способом. // III-я Межрегиональная науч.-техн. конф. студентов и аспирантов Информационные технологии, энергетика и экономика: Сб. трудов. В 3 т. Т 2. – Смоленск: 2006. – С. 214-220.

Шайторов Андрей Валерьевич

Выпускник электротехнического факультета
ГУ ВПО Белорусско-Российский университет, г. Могилев
Тел.: +375(29) 689-78-14
E-mail: rain@fenomen-games.com

Альховик Сергей Александрович

Доцент «Автоматизированные системы управления», канд. техн. наук
ГУ ВПО Белорусско-Российский университет, г. Могилев
Тел.: +375(222) 42-42-02
E-mail: asa@tut.by