

Модульный подход к конструированию имитационных моделей технологических процессов

Е.Н. Данилов (ИПУ РАН),

А.В. Локшин, А.Ю. Новичков (ЗАО "Хоневелл")

Рассматривается подход к организации систем конструирования имитационных моделей непрерывных процессов химико-технологического типа. Предлагается разделение задачи создания модели по различным клиентским приложениям. Выбранный подход обеспечивает широкие возможности для разработки имитационных моделей и развитие самих средств конструирования.

Ключевые слова: моделирование, имитационная модель, база данных, модульный подход, параллельная разработка.

Введение

Имитационные модели (ИМ) непрерывных процессов химико-технологического типа все более широко востребованы. Они могут использоваться как для решения задач технологического инжиниринга, разработки и оптимизации работы АСУ¹, так и в качестве базовых элементов тренажерных комплексов для обучения операторов ТП. Создание качественных ИМ очень ресурсоемко и требует привлечения высококвалифицированных специалистов в разных предметных областях — математического моделирования, химической технологии, систем управления.

Все чаще ИМ заказываются для вновь строящихся или существенно реконструируемых объектов, так как при наличии ИМ можно произвести обкатку средств АСУ верхнего уровня и логики систем ПАЗ на имитационной модели до пуска установки, тем самым исключив большую часть проявляющихся при пуске недочетов и существенно уменьшить вероятность возникновения нештатных ситуаций после пуска. ИМ в составе тренажерного комплекса позволяют провести обучение будущих операторов ТП, то есть дать им возможность ознакомиться с операторскими интерфейсами и получить навыки работы в них, более качественно изучить ТП, пройти процедуру пуска/останова установки и отработать действия в нештатных, предаварийных и аварийных ситуациях [1].

Создание ИМ для строящихся установок предъявляет особо жесткие требования к процессу их разработки. В этом случае невозможно использовать экспериментальные данные, полученные на работающем объекте, а приходится использовать только проектную документацию и оценки экспертов. Сроки разработки ИМ в этом случае также сильно ограничены, так как работа над созданием ИМ может быть начата только после получения необходимой информации о ТП и АСУ, а должна быть закончена за несколько месяцев до ввода объекта в эксплуатацию, чтобы операторы успели пройти полно-

ценный курс тренинга. Реально сложившийся на объекте технологический режим почти всегда отличается от проектного, и после ввода установки в эксплуатацию приходится производить подстройку параметров ИМ под реальный режим. Несмотря на это первичная ИМ, полученная до пуска установки и разработанная на основе проектных данных, сразу же должна обеспечивать высокий уровень адекватности. Это необходимо для избежания "ложных" операторских навыков и для исключения возможности некорректного инжиниринга, так как при недостаточной адекватности ИМ в ходе настройки средств управления в алгоритмы их функционирования могут быть внесены изменения, которые могут вызвать нештатные ситуации после пуска [2].

Исходя из вышесказанного, понятно, что даже при наличии квалифицированных специалистов создание ИМ сложных технологических объектов затруднительно, если не невозможно, без инструментальных средств соответствующего уровня, имеющихся в распоряжении разработчиков. Конкурировать на рынке услуг по созданию ИМ могут только компании, имеющие в своем арсенале высокоэффективные автоматизированные средства построения и конфигурирования ИМ, предоставляющие возможность параллельной разработки модели группой специалистов. В то же время постоянно возрастающие требования к качеству и срокам создания ИМ приводят к необходимости непрерывного развития и усовершенствования таких средств.

По аналогии с реальным ТП ИМ можно разделить на следующие основные компоненты (рис. 1):

1. Математическая модель ТП — центральная часть ИМ, отвечающая за моделирование оборудования и протекающих в нем физико-химических процессов;
2. Система управления — средства воспроизведения работы АСУ;
3. Интерфейс оператора — графические средства управления работой ТП.

При этом к моделированию работы систем управления и операторских интерфейсов можно подходить двумя путями: эмуляция средствами ИМ; использование программных средств, применяемых на реальном ТП.

Все три части ИМ представляют собой независимые блоки, которые разрабатываются по отдельности, а затем собираются в единое целое.

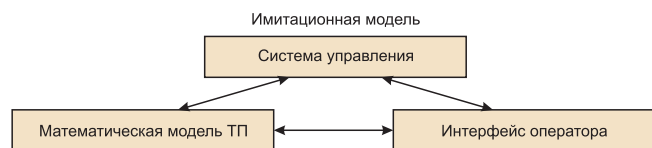


Рис. 1. Структурная схема ИМ

¹ Под АСУ в контексте данной статьи будем подразумевать системы базового регулирования (КИПиА) и противоаварийной защиты (ПАЗ), АСУ верхнего уровня — систему диспетчерского контроля и сбора данных (SCADA).

На сегодняшний день типовые средства имитационного моделирования сложных технологических объектов не получили широкого распространения, и компании-производители ИМ разрабатывают собственные системы для их создания. В то же время имеются свойства, присущие всем таким системам. Это определяется тем, что процесс создания ИМ непрерывных процессов химико-технологического типа включает следующие типовые стадии:

1. Выбор (исходя из специфики ТП) базовых алгоритмов физико-химических расчетов;

2. Определение набора используемых в математической модели химических веществ, составов входящих на установку потоков, задание возможных химических реакций, а также условий, влияющих на направление и скорость их протекания;

3. Конфигурирование топологической схемы оборудования и настройка его параметров;

4. Реализация функций систем управления – конфигурирование элементов КИПиА (в частности задаются параметры дискретных сигнализаторов, аналоговых датчиков и регуляторов, границы параметров сигнализаций и блокировок, параметры ПИД-регуляторов) и воссоздание работы контроллеров (в том числе систем ПАЗ) или подключение контроллеров, применяемых на реальном ТП;

5. Создание операторского интерфейса или подключение интерфейса, используемого на реальном ТП;

6. Сборка ИМ, окончательная отладка и настройка.

Наиболее ресурсоемкими являются разработка математической модели процесса с настройкой параметров моделируемого оборудования (п. 1-3) и окончательная настройка и отладка ИМ (п. 6). Однако для объектов со сложным управлением временные затраты на реализацию этих систем (п. 4) могут оказаться соизмеримыми с затратами на моделирование ТП.

На качество получаемой ИМ наибольшее влияние оказывает правильность задания конфигурационных параметров моделируемого оборудования, а также глубина и адекватность математических базовых моделей оборудования, доступных разработчику ИМ.

Поэтому основной вектор развития средств конструирования имитационных моделей (СКИМ) направлен на:

- упрощение процесса конфигурирования моделей технологического оборудования и улучшение средств верификации задаваемых параметров, ограничивающих возможность задания некорректных конфигураций;
- совершенствование инструментов настройки и отладки ИМ;
- повышение качества и функциональности базовых моделей оборудования;

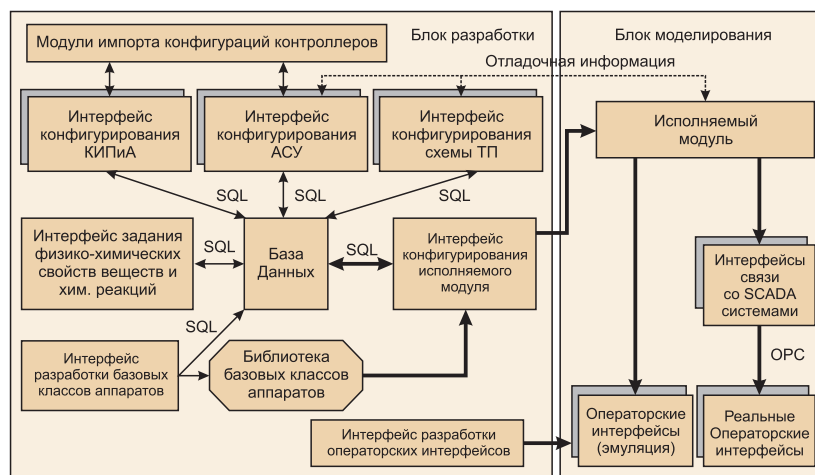


Рис. 2. Структурная схема средства СКИМ, построенного по модульному принципу

- расширение возможностей интеграции работы ИМ с промышленными контроллерами разных производителей.

Большинство существующих СКИМ построено на основе сложных монолитных многофункциональных программных продуктов, позволяющих производить полный цикл разработки, отладки, а иногда и являющихся основным средством конечного использования ИМ (средой исполнения). Несмотря на высокую функциональность и удобство использования, развитие таких комплексов, поддерживающих большое число принципиально разных возможностей (от редактирования свойств веществ и создания топологической схемы оборудования, до подключения "реальных" контроллеров и настройки связи со средствами верхнего уровня АСУ), оказывается очень затратным. Большинство усовершенствований в таких системах вносится в виде пакета обновлений, а более существенные изменения реализуются в новой версии программного продукта. При этом часто возникают проблемы с совместимостью проектов ИМ, созданных в разных версиях. Для их решения прибегают к специальным миграционным средствам, позволяющим конвертировать ранее созданные проекты в формат новых версий. Но даже использование миграционных средств не дает стопроцентной гарантии успешного переноса, а во многих случаях после перехода на новую версию приходится дополнительно вручную перенастраивать часть конфигурационных параметров уже собранных моделей.

Модульная архитектура СКИМ

Для преодоления указанных трудностей и обеспечения возможности эффективного развития СКИМ был принят подход, основанный на модульной архитектуре [3]. Единое средство конструирования ИМ было разделено на набор отдельных специализированных компонентов, реализованных в виде независимых программных продуктов, решающих конкретные задачи построения ИМ. Структурная схема модульного СКИМ представлена на рис. 2.

Программный комплекс разделен на два основных блока – блок, предназначенный для конструирования ИМ, и блок имитационного моделирования, обеспечивающий работу созданной модели.

Блок разработки состоит из следующих модулей:

1. БД – реляционная БД, которая является центральным хранилищем всех данных, необходимых для создания ИМ. Здесь содержатся модули функциональных блоков и аппаратов ТП, единая БД физико-химических свойств веществ, результирующие конфигурации всех уже разработанных ИМ, включая топологию оборудования ТП, настроечные параметры технологических блоков и конфигурацию средств КИПиА;

2. Интерфейс конфигурирования схемы ТП – среда для создания топологической схемы ТП, конфигурирования отдельных технологических аппаратов и отладки ИМ. Чаще всего (хотя и необязательно) в этом интерфейсе используется графическое представление ТП. В нотации функционально-блочных диаграмм (ФБД) каждый блок на диаграмме представляет собой аппарат с задаваемыми свойствами, блоки соединяются материальными потоками ("трубами");

3. Интерфейс конфигурирования КИПиА – средство редактирования настроек регуляторов, датчиков, клапанов и других элементов базового регулирования;

4. Интерфейс конфигурирования АСУ – графическая среда конфигурирования указанных систем (исключая КИПиА). Здесь также чаще всего используется нотация ФБД, где каждый блок является логическим элементом с задаваемыми свойствами; блоки соединяются логическими связями;

5. Модуль импорта конфигураций контроллеров – средство для автоматизированного импорта в ИМ используемых на реальной установке конфигураций программируемых контроллеров;

6. Интерфейс задания физико-химических свойств веществ и химических реакций – средство редактирования допустимых в моделируемом ТП химических реакций и их характеристик;

7. Библиотека базовых классов аппаратов – набор доступных разработчику аппаратов ТП, содержащий алгоритмы расчета моделируемых процессов и гидравлики;

8. Интерфейс разработки базовых классов аппаратов – средство для разработки базовых классов технологических аппаратов и внесения необходимой информации о них в БД;

9. Интерфейс разработки операторских интерфейсов – среда создания интерфейсов оператора, используемых для управления моделируемым ТП;

10. Интерфейс конфигурирования исполняемого модуля – средство создания на основе БД конфигурационных файлов исполняемого модуля ИМ.

Блок моделирования включает:

- исполняемый модуль – "проигрыватель" сконфигурированной ИМ. Исполняемый модуль является сервером данных, полученных в ходе моделирования. При необходимости данные могут передаваться на

SCADA-системы или операторские интерфейсы. В ходе отладки эти данные могут запрашиваться различными интерфейсами конфигурирования для визуального представления хода выполнения процесса и внесения в него изменений;

- операторские интерфейсы (эмуляция) – интерфейсы оператора ТП, эмулирующие внешний вид реально используемых на установке интерфейсов;

- интерфейсы связи со SCADA-системами – модули для подключения ИМ к системам, реально используемым на установке;

- реальные операторские интерфейсы – используемые на установке интерфейсы оператора, подключаемые к ИМ (используются в тренажерных моделях для наиболее полного воспроизведения работы с реальным ТП).

Описанное распределенное средство создания ИМ не является монолитным программным продуктом, а представляет собой набор самостоятельных, независимых программ (модулей), в комплексе позволяющих решить поставленную задачу конструирования ИМ.

Особое место в распределенной структуре СКИМ занимает БД, являющаяся основной частью блока разработки. В ней содержится информация и по разрабатываемым проектам, и по доступным средствам разработки. Каждый модуль, обращаясь к БД, получает информацию о доступных ему возможностях и реализует их в рамках своей компетенции. Например, интерфейс конфигурирования схемы ТП получает набор доступных аппаратов и их свойств; интерфейс конфигурирования КИПиА получает набор характеристик, необходимых для настройки регуляторов и датчиков, и т.д. Все данные по создаваемой ИМ также находятся в БД. Одновременно несколько "модельеров" могут использовать различные интерфейсы для конфигурирования своей части модели, и в результате в БД получается полностью сконфигурированная и готовая к выполнению ИМ. Интерфейс конфигурирования исполняемого модуля на основе информации из БД создает файлы конфигурации для исполняемого модуля, который на их основе выполняет имитационное моделирование ТП. Интерфейсы конфигурирования могут обращаться к исполняемой модели и получать от нее отладочную информацию о ходе моделирования или внести в нее изменения "на лету", необходимые для более точной настройки параметров ИМ.

Преимущества предложенной схемы

Разработчики единожды при разработке СКИМ определяют интерфейсы (протоколы) обмена данными между БД (SQL-храняемые процедуры и представления), исполняемым модулем и клиентскими конфигурационными модулями. Определив их, разработчики получают возможность произвольно изменять внешние по отношению к БД интерфейсы конфигурирования, придерживаясь только указанного соглашения. При этом, поскольку все данные хранятся централизованно в БД, автоматически решается

проблема поддержки работоспособности ранее созданных моделей, которая всегда остро стоит в такого рода задачах. Более того, даже если в какой-то момент возникает необходимость в кардинальных изменениях данного стандарта, разработчикам достаточно перейти на новые версии конфигураторов отдельных подсистем без использования каких-либо специальных средств миграции.

Каждый отдельный модуль СКИМ может развиваться самостоятельно и не требует доработки остальных компонентов. Это дает возможность вести развитие параллельно в нескольких направлениях разными группами разработчиков. Последние не ограничены в средствах разработки: нет жестких требований к языку программирования, на котором будет написан данный компонент; неважно, будет ли он целиком самостоятельным средством или надстройкой к уже существующим программным продуктам. Единственное требование к модулям – это поддержка утвержденных форматов и протоколов связи. Благодаря этому каждый компонент разрабатывается на средствах, наиболее удобных для решения возлагаемой на него задачи.

Непрерывное развитие модулей не доставляет каких-либо неудобств конечным пользователям – специалистам, непосредственно занятым созданием ИМ. Появление новых компонентов благодаря сохранению формата обмена данными с БД и исполняемым модулем не приводит к замене старых средств на новые, а лишь предоставляет новые возможности, не отменяя имеющихся. При этом для использования этих возможностей для уже созданных или создающихся проектов не требуется проводить процедуры миграции проекта на новую версию.

Сказанное относится не только к интерфейсам конфигурирования, но и к исполнителю модулю. Новые, более современные его версии могут требовать иного формата конфигурационных файлов, но и в этом случае не требуется производить существенную переделку всех средств конфигурации, а достаточно написать новый модуль интерфейса формирования конфигурационных файлов, отвечающего новым требованиям. Такой подход позволяет формировать конфигурационные файлы не только для текущей версии исполнительного модуля, но и поддерживать его предыдущие релизы.

Развитие СКИМ может происходить одновременно в нескольких направлениях.

1. Развитие библиотеки классов аппаратов – создание модулей новых компонентов ТП (аппаратов) и развитие функциональных возможностей уже существующих. При этом сразу после добавления нового аппарата в БД возможность его использовать ментально появляется у всех разработчиков, без перехода на новую версию ПО;

2. Развитие средств конфигурирования всех элементов ИМ – ТП, АСУ, интерфейсов, а также расширение возможностей их отладки;

3. Развитие исполняемой части ИМ – совершенствование связи с операторскими интерфейсами и поддержки новых SCADA-систем.

Типичным примером развития может служить добавление в систему нового моделируемого объекта. Так как приходится разрабатывать ИМ для большого числа различных производственных объектов, принципиально невозможно заранее предусмотреть сразу весь спектр необходимого оборудования. Хотя большинство встречающихся на моделируемом объекте аппаратов, как правило, попадает в разряд часто встречающихся (типовых) и уже содержится в библиотеке базовых классов и доступно к использованию, зачастую встречается и уникальное, ранее не использовавшееся оборудование. В этом случае необходимо создать математическую модель нового технологического объекта и добавить его в систему конфигурирования.

Рассмотрим подробнее, как происходит этот процесс. Разработчик, используя интерфейс разработки базовых классов аппаратов, добавляет новый класс аппарата в библиотеку классов и заносит в БД информацию о нем: возможные параметры конфигурации, доступные для "модельеров", то есть характеристики новых аппаратов, определяющие тип их взаимодействия с остальным оборудованием (наличие входов/выходов и их тип; материальные, тепловые, информационные потоки и т.д.). При этом конфигурационные параметры могут быть как обязательные, без задания которых использование данного аппарата невозможно, так и вспомогательные, позволяющие произвести более точную настройку. После того как данные о новом оборудовании внесены в БД, это оборудование автоматически становится доступно для использования в интерфейсах конфигурирования схемы ТП.

Такой результат достигается благодаря тому, что сами интерфейсы конфигурирования не содержат данных об используемом оборудовании, а читают эти данные из БД. Так, при запуске клиента для конфигурирования топологической схемы ТП он получает из БД список доступного технологического оборудования, а затем для каждого типа аппарата получает набор конфигурационных параметров.

Отметим, что новый аппарат или обновленная (улучшенная) версия существующего аппарата становятся доступными не только в будущих проектах, но и во всех предыдущих, разработанных до того, как данное оборудование появилось в конфигураторе. После его появления старые проекты могут быть доработаны. При этом, чтобы начать использовать новое оборудование, не требуется какой-либо доработки других компонентов.

Выводы

Резюмируем основные особенности выбранного подхода.

1. Разделение монолитного средства разработки ИМ на отдельные независимые программные модули дает возможность параллельной разработки ИМ несколькими специалистами одновременно.

2. Развитие отдельных модулей ограничено только форматом протоколов обмена данными с БД и исполняемым модулем, что дает большую свободу для разработчика в плане выбора средств разработки.

3. Развитие модулей может идти параллельно с разработкой ИМ и не доставляет никаких неудобств конечному пользователю ("модельеру").

4. Отсутствуют проблемы, связанные с переходом на новую версию ПО и поддержкой ранее созданных моделей.

Строго говоря, проблемы перехода на новую версию стандарта хранения ИМ в БД никуда не исчезают, однако в данном случае они ложатся на плечи разработчиков БД (а не на конечных пользователей), которые гораздо лучше понимают внутреннюю структуру организации данных и могут более квалифицированно совершить процесс перехода.

Единственным ограничением описанной схемы является "привязка" разработчиков к единому хранилищу данных, что, вообще говоря, затрудняет автономную удаленную разработку ИМ. Однако это ограничение преодолевается путем использования локальных

копий БД и периодической синхронизацией данных с основной БД.

Предложенный подход неоднократно опробован в проектах ЗАО "Хоневелл". В период с 2005-2009 гг. на его основе разработано более 25 стандартных и специализированных ИМ, используемых для обучения операторов и инжиниринга систем управления на предприятиях нефтепереработки и нефтехимии. Модульный подход показал свою высокую эффективность и производительность при моделировании ТП, а также расширенные возможности развития инструментальной части СКИМ, поддержки и сопровождения ранее созданных моделей.

Список литературы

1. Дозорцев В.М. Компьютерные тренажеры для обучения операторов технологических процессов. М. СИНТЕГ. 2009.
2. Новичков А.Ю. Имитаторы технологических процессов для тестирования и настройки систем программно-логического управления // Тр. научно-технич. конф. "Тренажерные технологии и симуляторы". СПб., 2002.
3. Данилов Е.Н., Локшин А.В. Автоматизированная система построения и отладки логических систем управления технологическими процессами // Тр. XX Межд. гонф. "ММТТ". Т7. Ярославль. 2007.

Данилов Евгений Николаевич – аспирант Института проблем управления им. В. А. Трапезникова РАН,

Локшин Александр Вячеславович – старший инженер,

Новичков Алексей Юрьевич – консультант ЗАО "Хоневелл".

Контактный телефон (926) 366-08-83. E-mail: danilov_e@mail.ru

Произведена модернизация системы измерения количества нефти на "Адамовой заставе"

На пункте сдачи-приема нефти (ПСП) "Адамова застава" (Республика Польша) произведена модернизация системы измерения количества нефти (СИКН) №702. Работы выполнены в рамках программы модернизации коммерческого узла учета нефти ПСП "Адамова застава". Решение о модернизации было принято руководством предприятия с целью приведения находившейся в эксплуатации с 2001 г. системы коммерческого учета нефти к действующим нормативным документам. Работы по модернизации системы выполнены совместно фирмой "ТРЭИ ГМБХ" и НПФ "КРУГ".

В процессе модернизации СИКН №702 были выполнены следующие работы: запущена в работу польская редакция SCADA КРУГ-2000 вер. 3.0 на АРМ операторов, метролога и диспетчера; приведен в соответствие с МИ 2632 ("Рекомендация. Плотность нефти и нефтепродуктов и коэффициенты объемного расширения и сжимаемости. Методы и программа расчета") алгоритм расчета плотности в нормальных условиях (15 °С), а также коэффициентов объемного расширения и сжатия нефти;

приведен в соответствие с ГОСТ Р 8.595 ("Масса нефти и нефтепродуктов. Общие требования к методикам выполнения измерений") алгоритм расчета массы нефти; конвертирован под широкоформатное разрешение новых мониторов (1680x1050 точек) графический проект СИКН; запущены в работу: версия ПО СРВК "TREI-5B-00" на новых процессорных модулях контроллеров, функция архивирования значений параметров, протоколов сообщений и печатных документов, Сервер единого времени TimeVisor, синхронизирующий время в системе при помощи GPS; проведены приемочные испытания (поверка вторичной аппаратуры); проведена поверка рабочих и контрольной линий по трубопоршневой установке, а также сличение рабочих линий по контрольной.

По окончании модернизации системы проведена проверка измерительно-вычислительного комплекса, и на основании положительных результатов СИКН №702 признана годной к проведению учетно-расчетных операций учета нефти между Польшей и РФ.

[Http://www.krug2000.ru](http://www.krug2000.ru)

Контроллеры от ICP DAS на базе Windows Embedded Standard 2009

Компания ICP DAS выпустила новую серию программируемых контроллеров XP-8000 на базе ОС Microsoft Windows Embedded Standard 2009, сочетающих богатые возможности графического интерфейса современного ПК и надежность промышленного контроллера. ОС Windows Embedded Standard 2009 по сути является версией широко распространенной ОС Windows XP, оптимизированной для встраиваемых PC-платформ. XP-совместимая ОС позволяет использовать многообразие уже имеющихся Windows-приложений, включая ПО на основе SCADA-пакетов. Разработка нового ПО может производиться в среде Microsoft Visual Studio.NET, Delphi и C++Builder. Процесс разработки ПО для XP-8000 не отличается от написания Windows-приложений для обычного ПК.

XP-8000 имеет VGA-интерфейс для подключения монитора и безвентиляторную конструкцию, в его корпусе преду-

смотрены прорези для конвективного обдува. Рациональная внутренняя компоновка обеспечивает необходимый тепловой режим при температурах окружающей среды до 75 °С. В XP-8000 применен не механический FLASH-накопитель, что в сочетании с безвентиляторным конструктивом делает его по аппаратной надежности, сопоставимым с промышленными контроллерами. Также имеются дополнительные средства повышения функциональной надежности: сторожевой таймер, энергонезависимая статическая память, дублированные Ethernet-интерфейс и вход питания. XP-8000 имеют модульную конструкцию, что позволяет устанавливать в него до семи модулей аналогового и дискретного ввода/вывода из широкой номенклатуры серий I-87000W (с последовательной шиной) и I-8000W (с параллельной шиной).

[Http://www.nnz-ipc.ru](http://www.nnz-ipc.ru)