

**РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ ANYLOGIC С ПОМОЩЬЮ ВНЕШНИХ
JAVAБИБЛИОТЕК ПРИ МОДЕЛИРОВАНИИ ФУНКЦИОНИРОВАНИЯ ЗДАНИЯ****О. Ю. Марьясин (Ярославль)**

Современное здание, включает в себя множество инженерных систем, таких как система отопления, вентиляции и кондиционирования (Heating, Ventilation, and Air conditioning – HVAC), система энергоснабжения, система безопасности (охранно-пожарной сигнализации, контроля доступа), автоматизированная система управления зданием (Building Management System – BMS), а так же взаимодействующих с ними людей, представляет собой сложную человеко-машинную систему. Определение значений качественных характеристик такой сложной системы невозможно без проведения исследований различных режимов функционирования входящих в ее состав подсистем и всей системы в целом. Так как проведение натуральных экспериментов на действующей системе является весьма затратным, а часто вообще невозможным, то для этого необходимо использовать компьютерное и имитационное моделирование.

Ранее для анализа энергопотребления и функционирования инженерного оборудования здания, при участии автора, была разработана компьютерная модель в системе AnyLogic[1]. Одной из основных причин, по которой была выбрана система AnyLogic, это то, что она поддерживает технологию агентного моделирования, а также включает пешеходную библиотеку для моделирования движения пешеходов (людей) в физическом пространстве. Пешеходная библиотека AnyLogic позволяет моделировать поведение людей внутри здания. Люди могут передвигаться по коридору и заходить в помещения. Люди, попавшие в помещения могут задерживаться там на различное время. Внутри помещения пользователи могут включать различные офисные или бытовые электроприборы, управлять микроклиматом помещений. Все это позволяет точнее моделировать работу, ориентированных на людей, инженерных систем здания и связанные с этим затраты энергоресурсов, а также бытовые тепловыделения от искусственного освещения, людей, включенных электроприборов и технологического оборудования.

Одним из недостатков использования AnyLogic для моделирования работы инженерных систем зданий является то, что палитра компонентов AnyLogic не включает элементы, предназначенные для моделирования объектов, описываемых в виде передаточных функций или уравнений состояния, блоков систем автоматического управления, таких как регуляторы различного типа, наблюдатели, фильтры и другие. В этом отношении система AnyLogic значительно уступает таким пакетам как MATLAB и системам, основанным на языке Modelica. Реализовать необходимые алгоритмы автоматического управления в AnyLogic можно на языке Java. Однако разрабатывать большие фрагменты кода непосредственно в среде AnyLogic довольно затруднительно. Гораздо удобнее это делать в интегрированных системах разработки (IDE), таких как NetBeans или Eclipse. В этом случае разработку кода алгоритмов управления можно производить в одной из упомянутых IDE, а скомпилированную библиотеку в виде файлового архива Java (.jar) подключать к системе AnyLogic.

При участии автора была разработана Java библиотека mcontrol, включающая набор различных алгоритмов автоматического управления. В частности, библиотека содержит класс MPID, реализующий классический ПИД-регулятор параллельного типа. Данный класс имеет методы: setP, setI, setD – для ввода отдельных настроек ПИД-регулятора, setSetpoint – для ввода значения задания, setOutputLimits – для ввода ограничений на выход регулятора. Основным методом класса MPID является метод getOutput, возвращающий выход регулятора и реализующий алгоритм ПИД-регулятора на основе текущего значения регулируемой переменной и задания.

Для подключения внешней Java библиотеки к системе AnyLogic необходимо добавить ссылку на эту библиотеку в секции “Зависимости” панели свойств модели. Кроме того, в модели AnyLogic необходимо создать переменную, например pid, типа mcontrol.MPID, а в качестве

начального значения этой переменной ввести команду создания экземпляра класса – `new mcontrol.MPID(Kp, Ti, 0)`. После этого внутри функций и событий AnyLogic можно обращаться к методам класса MPID, например, как показано на рис. 1а.

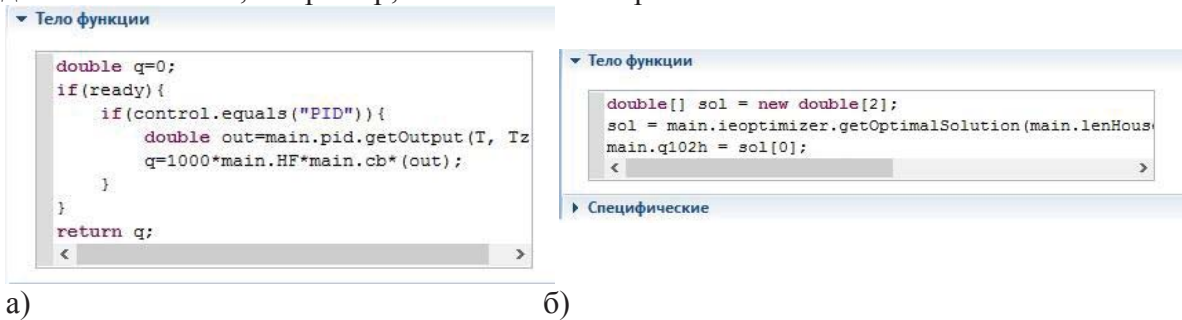


Рис. 1. Примеры использования внешних Java библиотек: а) mcontrol б) ieoptimizer

В модели, представленной в работе [1] использовались только два типа агентов: Main (основной класс модели, играет роль среды обитания для других агентов) и Пешеход (класс, моделирующий пешеходов). В дальнейшем, модель была существенно доработана и стала действительно мультиагентной. Подробное описание мультиагентной системы управления зданием и соответствующей ей модели AnyLogic приведено в работе [2]. Состав агентов модели включает локальных агентов системы отопления, вентиляции и кондиционирования (локальные HVAC агенты), локальных агентов системы освещения, локальных агентов энергоснабжения, зональных HVAC агентов и центрального HVAC агента. Кроме этого, для лучшего моделирования поведения людей в модели был создан тип агента человека HumanAgent, расширяющий класс Пешеход пешеходной библиотеки.

Локальный агент может быть связан с конкретным оборудованием, например, термостатом и локальной системой регулирования, реализующей, например, ПИД-закон регулирования температурой. Локальный агент всегда привязан к конкретному помещению. Кроме того, локальный агент привязан к конкретным видам энергоресурсов и по каждому из них имеет определённый лимит. Владельцем локального агента может быть или другой локальный агент или зональный объект. Для выполнения своих операций локальный агент может получать информацию (сообщения) от агента человека, от зонального агента и от другого локального агента. Работа локальных HVAC агентов, локальных агентов системы освещения и локальных агентов энергоснабжения моделировалась с помощью диаграмм действий и состояний, и программирования на языке Java, в том числе, с использованием описанной ранее Java библиотеки mcontrol.

Зональные агенты также привязаны к одному или нескольким помещениям. Особенностью зонального агента является наличие процедур выявления предпочтений человека и оптимизации микроклимата и энергопотребления. Зональный агент может получать информацию от локальных агентов, от агентов людей, от центрального агента, от других зональных агентов. Особенностью центрального агента является наличие процедуры координации зональных агентов, а также функций по взаимодействию с агентами поставщиками. Центральный агент может получать информацию от зональных агентов, от агентов людей и от агентов поставщиков. У каждого агента есть своя функция полезности, которую агент стремится максимизировать.

Для обеспечения максимальной комфортности для людей при минимуме энергозатрат зональные и центральные агенты должны уметь решать задачи статической или динамической оптимизации. Система AnyLogic не имеет мощных встроенных возможностей по выполнению функций оптимизации и оптимального управления в процессе моделирования. Поэтому для решения задачи оптимального управления микроклиматом и энергосбережением в статике была написана внешняя Java библиотека ieoptimizer, использующая известную свободно-доступную библиотеку оптимизации JOM [3]. Библиотека JOM позволяет использовать популярные решатели задач оптимизации, в том числе, решатель нелинейных задач оптимизации большой размерности IPOPT [4]. Система AnyLogic может напрямую обращаться к библиотеке JOM. Однако для

сокращения операций при подготовке и решении задачи оптимизации энергопотребления удобнее пользоваться дополнительной специализированной Java библиотекой. Пример использования библиотеки `ieoptimizer` показан на рис. 16.

Первоначально, в компьютерной модели [1] моделирование микроклимата в помещениях здания производилось с помощью палитры компонентов “Системная динамика” системы AnyLogic. Это позволяет реализовывать простые модели, описываемые системами обыкновенных дифференциальных уравнений. Однако, при большом числе помещений и учитываемых ограждающих конструкций, компьютерная модель микроклимата здания получается слишком громоздкой. Для моделирования микроклимата и энергопотребления зданий давно и успешно используются специализированные системы энергомоделирования (Building Energy Modeling – BEM), такие EnergyPlus, ESP-r, IDA ICE, TRNSYS и др. На сегодняшний день, наиболее популярной из BEM систем является свободно-доступная программа EnergyPlus [5]. Применение этой системы стало общепринятой практикой при проведении энергомоделирования зданий. С помощью дополнительных инструментов и библиотек можно организовать совместное моделирование системы EnergyPlus с MATLAB/Simulink. В этом случае EnergyPlus используется для моделирования микроклимата и энергопотребления здания, а MATLAB/Simulink позволяет применять для управления микроклиматом современные эффективные алгоритмы оптимального управления, такие как Model Predictive Control (MPC) [6]. В работе [7] описывается программный комплекс для моделирования инженерных систем здания, в состав которого входит BEM система EnergyPlus, пакет MATLAB/Simulink и среда AnyLogic. MATLAB/Simulink совместно с EnergyPlus используется для моделирования микроклимата здания, сложного HVAC оборудования и алгоритмов системы управления. Среда AnyLogic позволяет моделировать поведение людей внутри здания и работу простого HVAC оборудования, систем энергоснабжения, освещения и безопасности. Использование для моделирования инженерных систем зданий пакетов, в которых эти системы реализуются наиболее удобно, повышает эффективность моделирования и сокращает время проектирования.

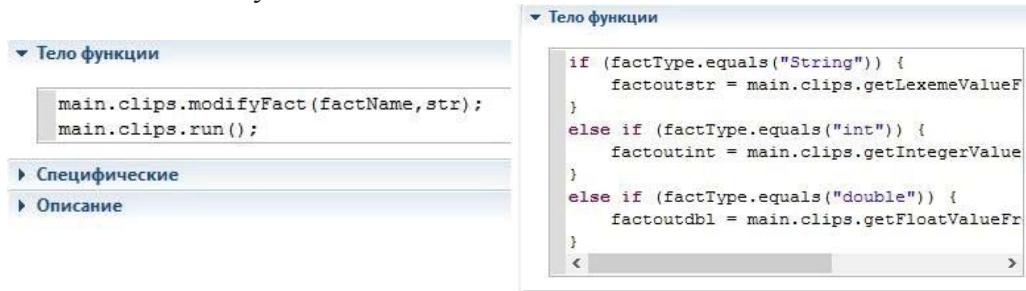
Для взаимодействия между MATLAB/Simulink и AnyLogic использовался язык Java, интерфейс OPC и разработанный ранее автором OPC Server. Данный OPC-сервер имеет целый набор тестовых сигналов, внутренних моделей объектов и свободные теги для использования внешними приложениями и моделями. Для доступа к OPC-серверу со стороны MATLAB/Simulink использовались функции OPC Toolbox. Они позволяют подключиться к определенному OPC-серверу, добавлять группы и теги, читать и записывать значения тегов. Для доступа к OPC-серверу со стороны AnyLogic использовалась внешняя Java библиотека JOPCCClientLib, выполняющая роль OPC-клиента. OPC-сервер позволяет подключить к программному комплексу любое программное приложение, являющееся OPC-клиентом. Например, для визуализации данных к программному комплексу может быть подключена SCADA система.

Автор предлагает повысить интеллектуальность зональных агентов описанной ранее модели в AnyLogic за счет использования технологий искусственного интеллекта. Это позволит агентам лучше реагировать на события, связанные с изменением предпочтений и поведением людей, и принятием решений в нештатных и чрезвычайных ситуациях. Для реализации интеллектуальных функций зональных агентов использовалась экспертная система продукционного типа на базе популярной свободно-доступной оболочки CLIPS [9]. Взаимодействие системы AnyLogic с CLIPS было организовано посредством разработанной автором внешней Java библиотеки CLIPSLib с использованием механизма CLIPS Java Native Interface (CLIPSJNI). Библиотека CLIPSLib позволяет выполнять стандартные функции CLIPS, такие как добавление, изменение и удаление фактов, загрузку фактов и правил в рабочую память, запуск машины вывода и другие.

При моделировании данные об изменении обстановки в здании, а также данные о предпочтениях людей передаются зональным агентом из AnyLogic в библиотеку CLIPSLib. Библиотека преобразует полученную информацию в символьные строки, являющиеся текстовым представлением фактов CLIPS и добавляет их в список фактов. Полученные экспертной системой

результаты передаются обратно в AnyLogic и используются зональным агентом для формирования команд локальным агентом или изменения параметров решения задачи оптимального управления микроклиматом. Пример вызова библиотеки при изменении значения факта CLIPS показан на рис. 2а, а при получении результатов от экспертной системы на рис. 2б.

Структура программного комплекса для моделирования функционирования здания, полученного в результате всех описанных разработок, показана на рис. 3. Для выполнения совместного моделирования в одном модельном времени, необходима синхронизация по времени во всех используемых системах моделирования. Если функции автоматического и оптимального управления могут быть полностью реализованы в MATLAB/Simulink, то соответствующие внешние Java библиотеки могут не использоваться.



а) б)

Рис. 2. Примеры использования внешней библиотеки CLIPSLib: а) при изменении значения факта б) при получении результатов

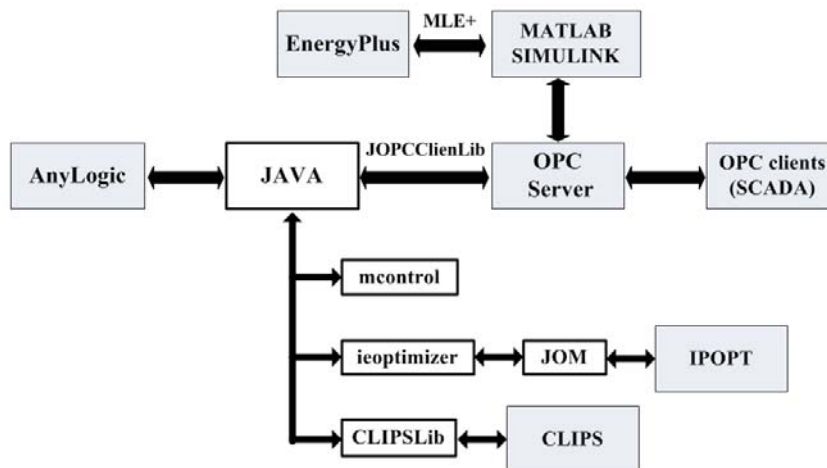


Рис. 3. Структура программного комплекса для моделирования функционирования здания

Разработанные внешние Javaбиблиотеки могут использоваться, при необходимости, как по отдельности, так и совместно в составе единого программного комплекса. Кроме того, данные библиотеки можно использовать с любыми другими приложениями, написанными на языке Java.

Выводы

Использование внешних Javaбиблиотек позволяет существенно расширить возможности имитационного моделирования в системе AnyLogic, добавить новые функции, организовать взаимодействие с другими системами моделирования или внешними приложениями. Таким образом, можно создавать программные комплексы, позволяющие комбинировать различные технологии и пакеты моделирования для решения сложных научных и практических задач.

Рассмотренный в статье программный комплекс позволяет не только моделировать работу различных инженерных систем здания, но и имитировать поведение людей в здании, а также исследовать зависимость энергопотребления здания от большого числа факторов (числа людей в коридорах и помещениях здания, количества включённых светильников и электроприборов, внешних метеоусловий, применяемых алгоритмов автоматического управления и т.д.). Он также

позволяет выявить проблемы системы безопасности здания, связанные с поздним оповещением о пожаре, закрытием или загромождением путей эвакуации, работой противодымной вентиляции и многое другое.

Литература

1. Марьясин О.Ю., Огарков А.А. Имитационное моделирование и оптимизация энергопотребления офисного здания. Электронный сборник трудов восьмой всероссийской научно-практической конференции “Имитационное моделирование. Теория и практика (ИММОД-2017)”, 2017. – С. 480-484.
2. Марьясин О.Ю. Проектирование мультиагентной системы управления зданием с использованием онтологий. Онтология проектирования, 2018, Т. 8, №3 (29). – С.387-399.
3. JOM (Java Optimization Modeler). URL: <http://www.net2plan.com/jom/index.php> (дата обращения: 07.08.2019).
4. Welcome to the Ipopt home page. URL: <https://projects.coin-or.org/Ipopt> (дата обращения: 07.08.2019).
5. EnergyPlus. URL: <https://energyplus.net/> (дата обращения: 07.08.2019).
6. Wang L. Model Predictive Control System Design and Implementation using MATLAB. Springer-Verlag London Limited, 2009, 396 p.
7. Марьясин О.Ю. Программный комплекс для моделирования инженерных систем зданий. В сборнике “Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM – 2018)”. – М.: ИПУ РАН, 2018. – С. 24-27.
8. CLIPS: A Tool for Building Expert Systems. URL: <http://www.clipsrules.net> (дата обращения: 07.08.2019).