



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Анализ библиотек и фреймворков поддержки
межпроцессного взаимодействия

Студент

ИУ6-73Б

(Группа)

А.А. Бушев 20.12.2022

(Подпись, дата)

А.А. Бушев

(И.О. Фамилия)

Руководитель

М.В. Фетисов 20.12.2022

(Подпись, дата)

М.В. Фетисов

(И.О. Фамилия)

ОТЛ
А.А. Бушев
20.12.2022

2022 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ЗАДАНИЕ
на выполнение научно-исследовательской работы

по теме Анализ библиотек и фреймворков поддержки межпроцессного взаимодействия

Студент группы ИУ6-73Б

Бушев Антон Алексеевич

(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

исследовательская

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения НИР: 25% 4 нед., 50% 7 нед., 75% 11 нед., 100% 14 нед.

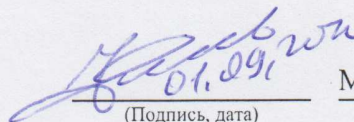
Техническое задание: выполнить анализ библиотек и фреймворков поддержки межпроцессного взаимодействия, осуществить выбор библиотек и фреймворков поддержки межпроцессного взаимодействия для адаптивной системы моделирования

Оформление научно-исследовательской работы:

- 1) Расчетно-пояснительная записка на 25-30 листах формата А4.
- 2) Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)
Необходимый иллюстративный графический материал включить в качестве рисунков в расчетно-пояснительную записку
- 3) Приложение А. Техническое задание на ВКРБ на 5-8 листах формата А4.

Дата выдачи задания « 1 » сентября 2022 г.

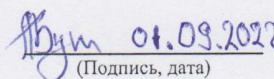
Руководитель


(Подпись, дата)

М.В. Фетисов

(И.О. Фамилия)

Студент


(Подпись, дата)

А.А. Бушев

(И.О. Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

РЕФЕРАТ

РПЗ 25 страниц, 9 рисунков, 1 таблица, 11 источников, 1 приложение
МЕЖПРОЦЕССНОЕ ВЗАИМОДЕЙСТВИЕ, ПРОЦЕСС,
БИБЛИОТЕКА, ФРЕЙМВОРК

Объектами исследования являются фреймворки и библиотеки поддержки межпроцессного взаимодействия.

Цель работы – исследование существующих библиотек и фреймворков поддержки межпроцессного взаимодействия. Разработать протокол взаимодействия с адаптивной системой моделирования SIMODO.

В результате работы был проведён сравнительный анализ библиотек и фреймворков поддержки межпроцессного взаимодействия, выявлены их преимущества и недостатки, на основании которых выбрана наиболее подходящая. Также был разработан протокол взаимодействия с адаптивной системой моделирования SIMODO.

Содержание

Введение	6
1 Описание открытой архитектуры интегрированной среды разработки SIMODO	7
2 Сравнительный анализ существующих библиотек и фреймворков поддержки межпроцессного взаимодействия	11
3 Протокол взаимодействия с адаптивной системой моделирования	20
Заключение	24
Список используемых источников	25
Приложение А (Черновик технического задания ВКРБ)	26

ОБОЗНАЧЕНИЯ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

SIMODO — название программных продуктов, разрабатываемых на кафедре ИУ6 МГТУ им. Баумана.

Git — система контроля версий.

LSP — открытый протокол, разрешающий реализацию и распространение поддержки языка программирования независимо от любого данного редактора или интегрированной среды разработки.

LSP+ — модификация протокола LSP под требования интегрированной среды разработки SIMODO.

ОС — операционная система.

Windows — операционная система для компьютеров от корпорации Microsoft.

Linux — семейство операционных систем на основе открытого ядра.

TCP/IP — сетевая модель передачи данных, представленных в цифровом виде.

UDP — протокол пользовательских датаграмм, один из ключевых элементов набора сетевых протоколов для Интернета.

Введение

В настоящее время для адаптивной системы моделирования SIMODO[1] актуален вопрос межпроцессного взаимодействия. Само моделирование должно происходить отдельно от визуализации результатов моделирования. Однако возникает вопрос, какой инструмент использовать для организации взаимодействия процесса моделирования и процесса визуализации результатов моделирования.

Исследовательская работа посвящена обзору существующих библиотек и фреймворков поддержки межпроцессного взаимодействия. А также проведению сравнительного анализа библиотек и фреймворков поддержки межпроцессного взаимодействия. Помимо этого, в работе разработан протокол взаимодействия с адаптивной системой моделирования SIMODO.

1 Описание открытой архитектуры интегрированной среды разработки SIMODO

Основной задачей интегрированной среды разработки SIMODO является обеспечение удобного средства описания моделей (помимо запуска моделирования и анализа результатов моделирования) на базе открытой архитектуры[2]. Открытая архитектура интегрированной среды разработки SIMODO позволяет расширить базовую функциональность для работы с различными документами.

Основной задачей интегрированной среды разработки SIMODO является предоставление простого средства редактирования текста моделей. Поскольку любой язык программирования может быть использован для написания фрагментов модели, то среда предоставляет гибкие средства настройки, не замыкаясь на наборе предметно-ориентированных языков SIMODO.

На рисунке 1 изображён графический интерфейс интегрированной среды разработки SIMODO.

Интегрированная среда разработки SIMODO состоит из расширяемого редактора — оболочка редактора и расширения (плагины) — и процессов, выполняющие разнообразные функции, которые целесообразно отделить от редактора межпроцессным взаимодействием[1]. Преимуществом такого отделения является отказоустойчивость всей интегрированной среды разработки SIMODO.

Оболочка редактора предоставляет базовую функциональность и механизм расширения. После запуска оболочка редактора загружает плагины. Плагины являются фабриками для соответствующих компонентов. Компоненты, которые поставляются плагинами, расширяют функциональность редактора. В свою очередь оболочка редактора предоставляет для указанных плагинов и порождающих ими компонентов интерфейс доступа к некоторым своим функциям.

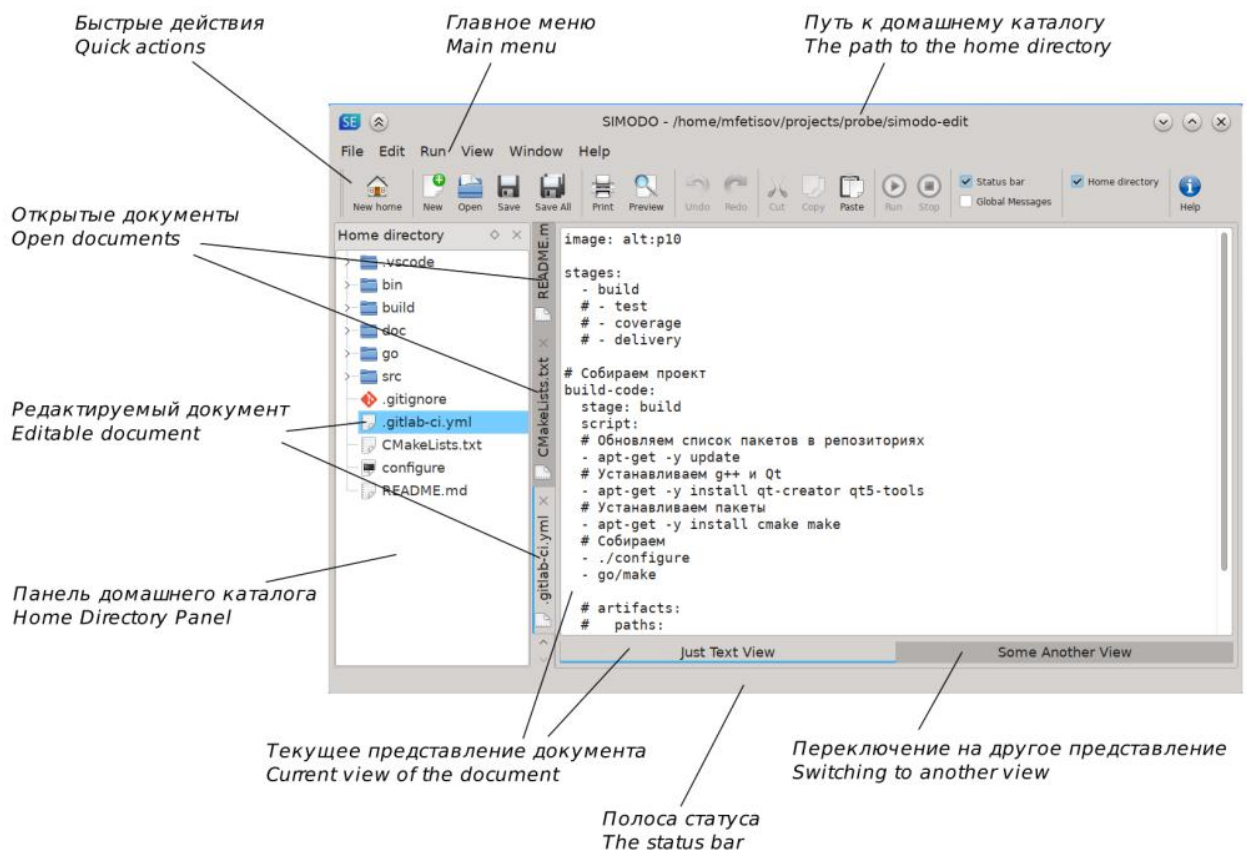


Рисунок 1 — Графический интерфейс интегрированной среды разработки SIMODO.

Сама оболочка не выполняет отображение или управление документом. Вместо этого она предоставляет пространство, где размещает панели и представление документов, отображение которых определяется реализацией плагинов.

На рисунке 2 представлена схема интерфейсов оболочки.

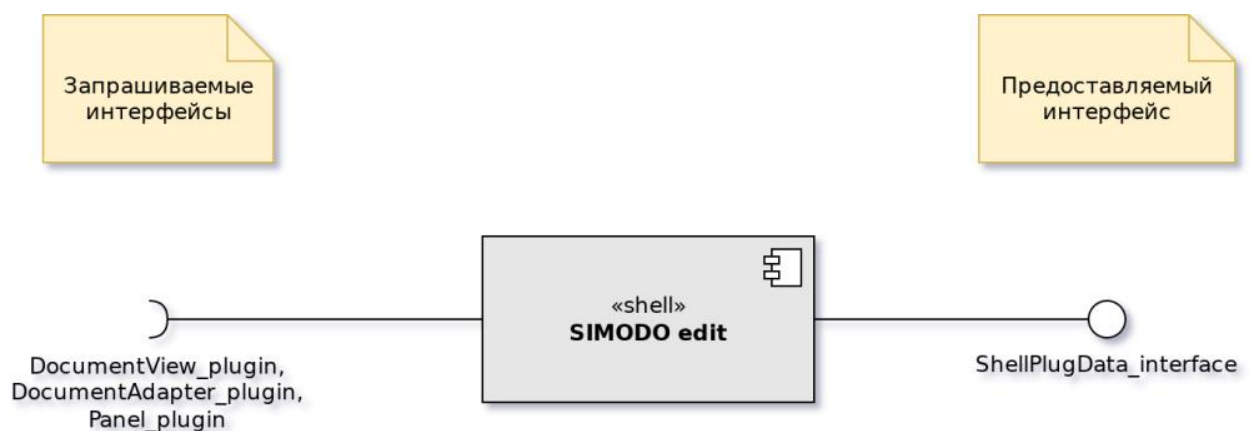


Рисунок 2 — Интерфейсы оболочки

Следующим слоем открытой программной архитектуры являются процессы, выполняющие разнообразные функции, которые целесообразно отделить от редактора межпроцессным взаимодействием. Преимуществом такого отделения является отказоустойчивость всей программы.

Процессы запускаются конкретными плагинами как консольные приложения и обмениваются между ними через каналы — механизм взаимодействия через стандартный ввод-вывод. Возможен как синхронный, так асинхронный обмен.

Примерами выделения в отдельные процессы могут быть: взаимодействие с системами управления версиями (например, git), языковые протоколы (например, LSP[3]; для языков серии SIMODO используется модифицированная версия LSP+[1]), сервер моделирования SIMODO и т.д.

На рисунке 3 представлено взаимодействия слоёв интегрированной среды разработки SIMODO.

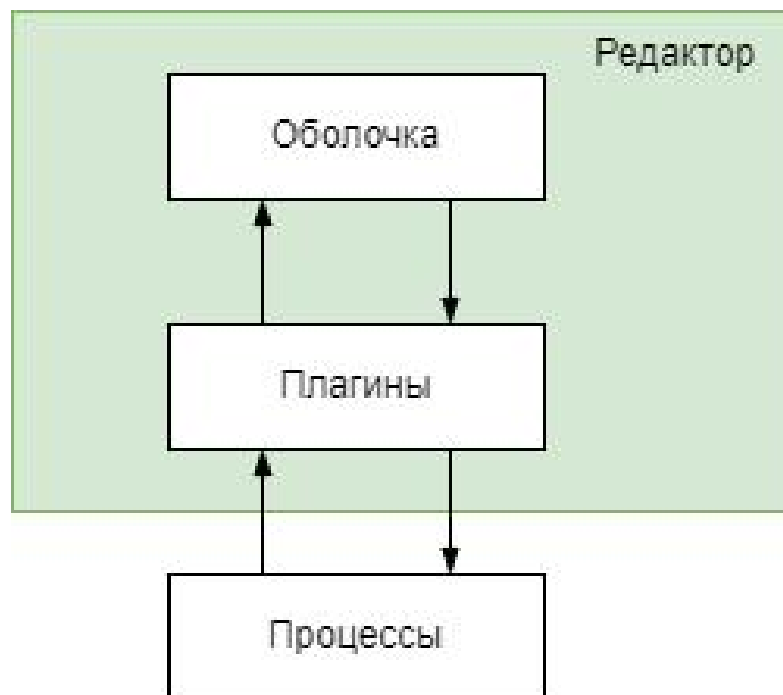


Рисунок 3 — Взаимодействие элементов интегрированной среды разработки SIMODO

Открытая программная архитектура интегрированной среды разработки SIMODO представляет собой вариант многоуровневой архитектуры, где нижележащие слои не зависят от вышележащих, а обмен сообщениями возможен только между соседними слоями.

Адаптивная система моделирования SIMODO относится к третьему слою архитектуры интегрированной среды разработки SIMODO. Среда должна выполняться в отдельном процессе, с которым плагин расширяемого редактора обменивается сообщениями.

Начало и завершение работы адаптивной системы моделирования SIMODO, а также управление ходом моделирования должны программно контролироваться плагином расширяемого редактора.

Таким образом возникает необходимость выбора инструмента реализации межпроцессного взаимодействия между плагинами и процессами.

2 Сравнительный анализ существующих библиотек и фреймворков поддержки межпроцессного взаимодействия

Для выявления достоинств и недостатков существующих библиотек и фреймворков, необходимо провести их сравнительный анализ, на основе которого будет выполнен выбор библиотеки или фреймворка.

Исходя из того, что в качестве целевой операционной системы для интегрированной среды разработки SIMODO является ОС Windows и дистрибутивы Linux[1], при выборе аналогов, будут рассматриваться кросс-платформенные библиотеки и фреймворки, которые разработаны под указанные ОС.

На основании того, что в качестве основного языка разработки интегрированной среды разработки SIMODO используется язык программирования C++[1], при выборе аналогов, будут рассматриваться библиотеки и фреймворки, которые обладают интерфейсом, совместимым с языком программирования C++.

На сегодняшний день существует множество библиотек и фреймворков поддержки межпроцессного взаимодействия. Одними из самых популярных являются [4]:

- Boost;
- eCAL;
- ACE;
- POCO;
- Qt.

Boost

Библиотека Boost.Process позволяет управлять системными процессами. Она предоставляет следующие возможности[5]:

- Создание дочернего процесса.
- Настройка каналов для дочернего процесса.

- Синхронное и асинхронное взаимодействие с дочерним процессом через каналы.
- Синхронное и асинхронное ожидание завершения процесса.
- Прерывание процесса.

Библиотека Boost.Interprocess снижает сложность использования межпроцессного взаимодействия и синхронизации, а также предлагает ряд механизмов межпроцессного взаимодействия и синхронизации[5]:

- разделяемая память;
- файлы, проецируемые в память;
- семафоры;
- мьютексы;
- условные переменные;
- размещаемые в разделяемой памяти и файлах, проецируемых в память, мьютексы;
- именованные объекты синхронизации;
- блокировка файлов;
- относительные указатели;
- очередь сообщений.

Библиотека Boost.Interprocess также предлагает высокоуровневые межпроцессного механизмы динамического выделения разделяемой памяти или файлов, проецируемых в память. Библиотека не требует компиляции, потому что состоит только из заголовочных файлов.

eCAL

Фреймворк eCAL (улучшенный абстрактный уровень связи) является связующим программным обеспечением, которое делает возможным масштабируемое, высокопроизводительное межпроцессное взаимодействие на одном вычислительном узле или между несколькими узлами в

вычислительной сети[6]. eCAL разработан для характерных сценариев облачных вычислений, когда разные процессы обмениваются данными используя шаблон издатель-подписчик.

Фреймворк eCAL автоматически выбирает лучший механизм передачи данных для каждого соединения:

- разделяемая для связи на одном вычислительном узле;
- UDP протокол для сетевого взаимодействия.

На рисунке 4 показан принцип работы фреймворка eCAL.

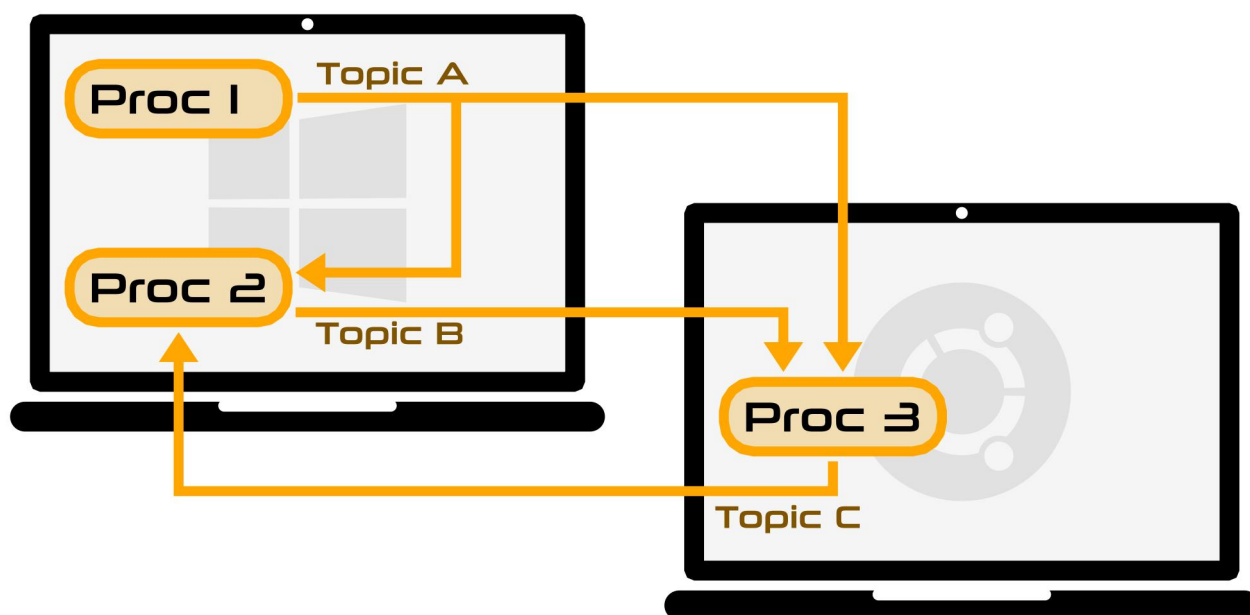


Рисунок 4 — Принцип работы фреймворка eCAL

Фреймворк eCAL обладает следующими особенностями[6]:

- скорость обмена данными 1–10 Гб/с;
- поддержка шаблона издатель-подписчик и сервер-клиент;
- обмен данных происходит без посредника;
- обладает мощным инструментарием отслеживания и анализа децентрализованных потоков данных;
- не требует настройки перед использованием;
- позволяет использовать пользовательский протокол сообщений;
- работает на ОС Windows и Linux.

ACE

Фреймворк ACE (адаптивное окружение связи) является объектно-ориентированным фреймворком, который реализует основные шаблоны для программного обеспечения параллельной связи. Фреймворк ACE предоставляет широкий набор C++ фасадов-обёрток, предназначенных для многократного использования, и компонент фреймворка, выполняющих общие для ряда ОС задачи программной связи[7]:

- Демультимплексирование событий и диспетчеризация обработки событий.
- Обработка сигналов.
- Инициализация сервисов.
- Межпроцессное взаимодействие.
- Управление разделяемой памятью.
- Маршрутизация сообщений.
- Динамическая настройка распределённых сервисов.
- Параллельное выполнение и синхронизация.

Фреймворк ACE предназначен для разработчиков высокопроизводительных сервисов связи и приложений реального времени[7]. Фреймворк упрощает разработку объектно-ориентированных сетевых приложений и сервисов, использующие межпроцессное взаимодействие, демультимплексирование событий, явное динамическое связывание и параллелизм. В дополнение, Фреймворк ACE автоматизирует системную настройку с помощью сервисов динамического связывания во время выполнения приложений и выполняет данные сервисы в одном или нескольких процессах или потоках.

Диаграмма на рисунке 5 иллюстрирует ключевые компоненты в фреймворке ACE и их иерархические связи.

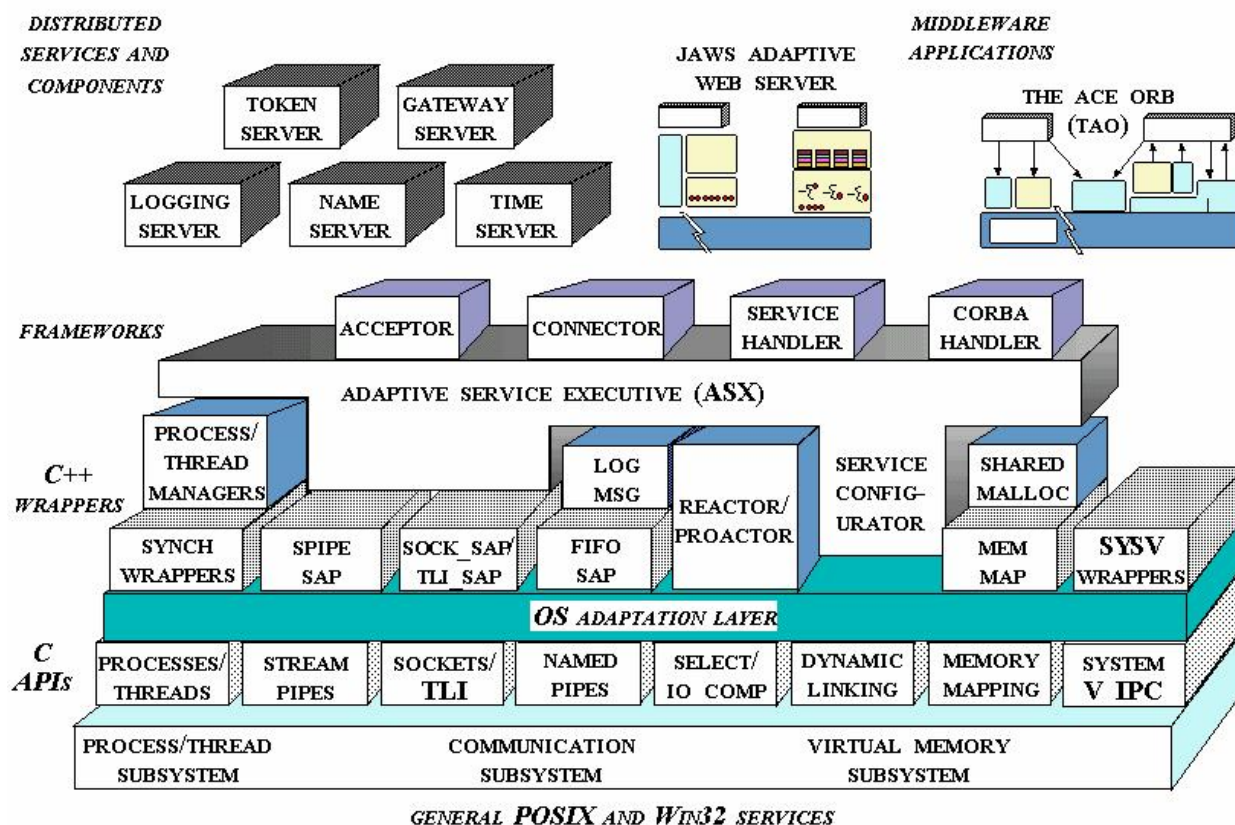


Рисунок 5 — Структура и функциональность фреймворка ACE

POCO

Библиотеки POCO являются кросс-платформенными библиотеками для создания ориентированных на сеть или интернет приложений, которые работают на персональных компьютерах, серверах, мобильных устройствах, устройствах интернета вещей и встроенных системах[8].

Библиотека POCO Process позволяет:

- Получить информацию о текущем процессе.
- Запустить новый процесс.
- Прервать другой процесс.
- Перенаправить ввод-вывод запускаемого процесса.
- Синхронизировать процессы, используя примитивы синхронизации: именованный мьютекс и именованное событие.
- Использовать разделяемую память.

Библиотека POCO является кросс-платформенной, модульной и масштабируемой библиотекой, используемой в программах корпоративного уровня[8]. Библиотека обеспечивает целостный и простой для понимания программный интерфейс.

Исходные тексты библиотеки POCO обладают особым качеством кода, читаемостью, полнотой, согласованностью и тестируемостью.

Состав библиотеки POCO представлен на рисунке 6.

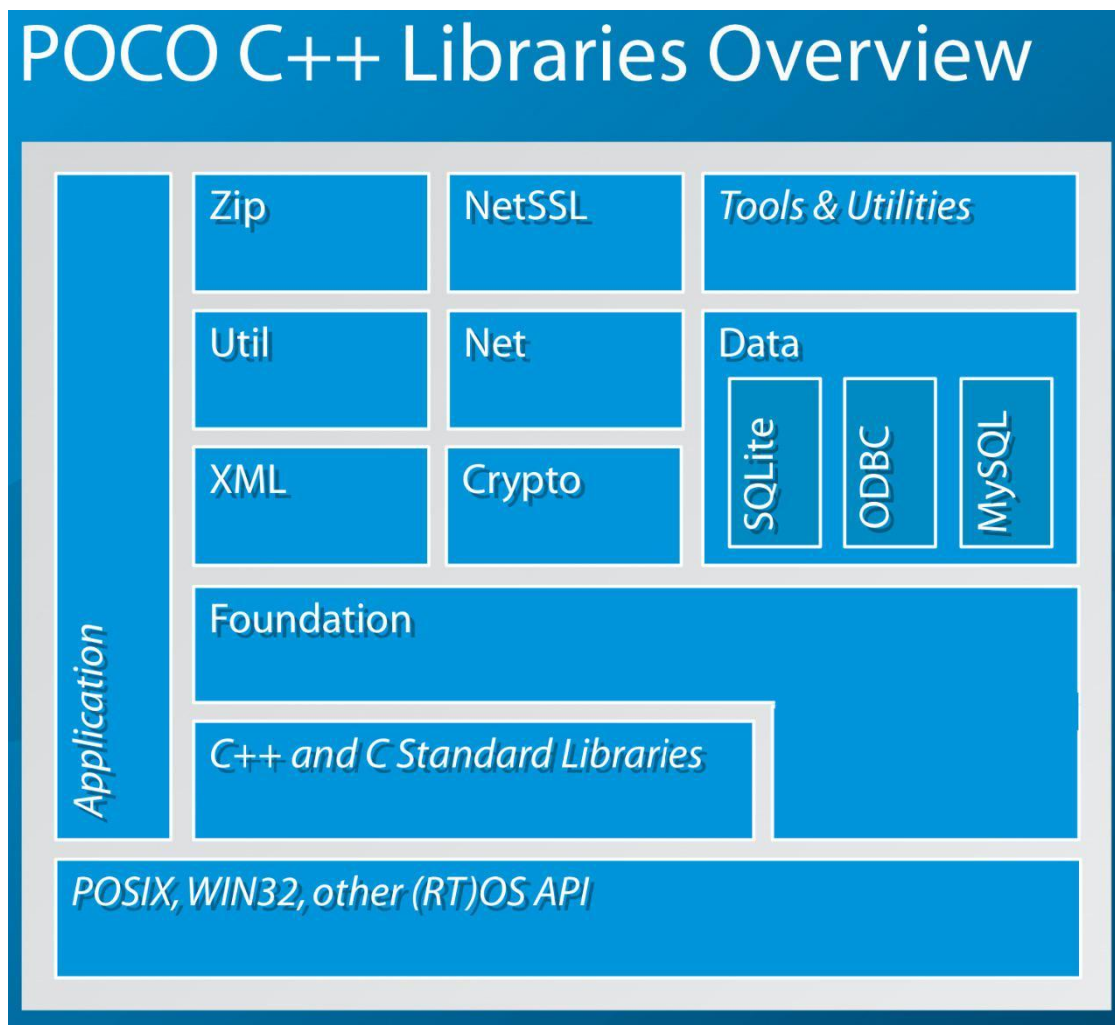


Рисунок 6 — Состав библиотеки POCO.

Qt

Фреймворк Qt является кросс-платформенным программным обеспечением для создания как графических пользовательских интерфейсов, так и кросс-платформенных приложений, запускаемых на различном программном и аппаратном обеспечении, например, операционные системы

Linux, Windows, macOS, Android или встраиваемые системы с минимальными изменениями нижележащей кодовой базы, оставаясь нативным приложением с соответствующими возможностями и скоростью[9].

Фреймворк Qt предоставляет несколько кросс-платформенных способов реализации межпроцессного взаимодействия[10]:

- Использование сетевого TCP/IP межпроцессного взаимодействия в вычислительной сети.
- Использование локального сервера и сокетов для межпроцессного взаимодействия на одном вычислительном узле.
- Разделяемая память.
- Класс QProcess позволяет запускать внешние программы в качестве дочерних процессов и взаимодействовать с ними.
- Управление сессиями, позволяющее распространять события между процессами.

Терминология для каналов в фреймворке Qt может ввести в заблуждение[10]. Выходные каналы процесса являются каналами чтения, а входные каналы процесса являются каналами записи. Данная терминология обусловлена тем, что из процесса читается результат его работы, а запись в процесс становится его входом.

Фреймворк Qt позволяет перенаправить только один выходной канал.

Интегрированная среда разработки SIMODO разрабатывается с использованием фреймворка Qt[1], поэтому механизм плагинов реализуется с использованием данного фреймворка. Каждый плагин, расширяющий функциональность оболочки будет зависеть от небольшой части фреймворка Qt, связанного с системой плагинов.

Состав фреймворка Qt представлен на рисунке 7.

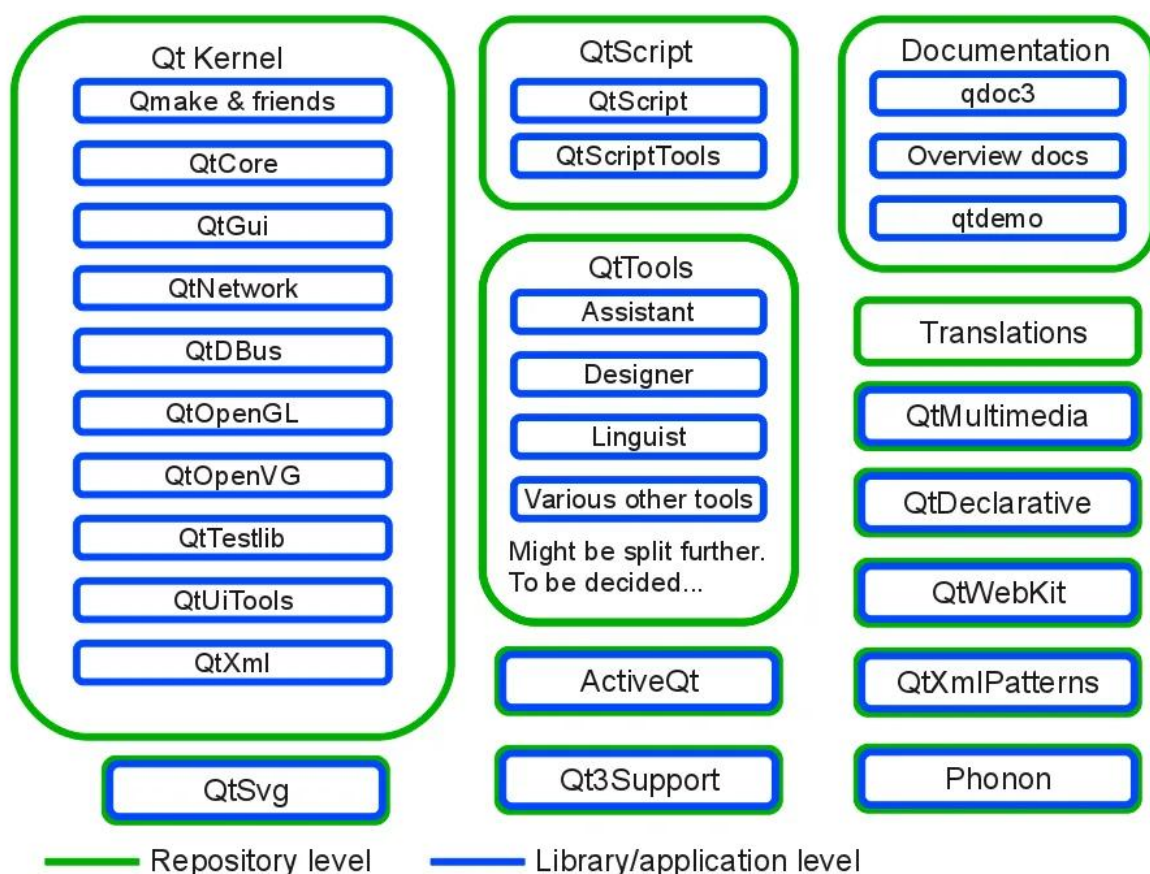


Рисунок 7 — Состав фреймворка Qt.

Анализа данных библиотек и фреймворков показывает, что инструменты реализации межпроцессного взаимодействия предоставляют широкий выбор методов межпроцессного взаимодействия. Однако для поставленной задачи нет необходимости в использовании всего инструментария тяжелых библиотек. Выделим критерии выбора:

- Возможность программного создания процесса: интегрированная среда разработки должна самостоятельно запускать сторонние процессы, чтобы этого не приходилось делать пользователю.
- Взаимодействие с процессом через каналы[11]: каналы являются наиболее простым способом взаимодействия с процессом, что позволит ограничить работу адаптивной системы моделирования

SIMODO с вводом-выводом до работы со стандартными потоками ввода-вывода.

- Программное завершение процесса: интегрированная среда разработки SIMODO должна иметь возможность преждевременно завершить работу внешних процессов, не дожидаясь завершения работы этих процессов в случае завершения работы среды.
- Библиотека или фреймворк должен состоять только из заголовочных файлов C++, чтобы исключить издержки, связанные с поставкой правильной версией библиотек времени выполнения программы.
- Размер полностью скомпилированной библиотеки должен быть минимален, чтобы итоговый размер приложения занимал как можно места в постоянном запоминающем устройстве пользователя.

По выделенным критериям проведён сравнительный анализ библиотек, который продемонстрирован в таблице 1.

Таблица 1 — сравнительный анализ библиотек поддержки межпроцессного взаимодействия

Критерий \ Библиотека	Программное создание процесса	Взаимодействие через каналы	Программное прерывание процесса	Состоит только из заголовочных файлов	Размер библиотеки, Мб
Boost	+	+	+	+	3
Qt	+	+	+	-	15
POCO	+	+	+	-	8
ACE	+	-	+	-	89
eCal	-	-	-	-	112

Сравнение приложений по выделенным критериям выявило, что среди предложенных библиотек и фреймворков поддержки межпроцессного взаимодействия библиотека Boost обладает самыми лучшими показателями относительно остальных кандидатов.

3 Протокол взаимодействия с адаптивной системой моделирования

Архитектура интегрированной среды разработки SIMODO предполагает три уровня:

1. Расширяемый редактор — оболочка, задача которой состоит в обеспечении механизма расширения и базовой функциональности (например, отображение окна программы и базовые действия работы с файлами).
2. Плагины — расширения оболочки, которые обеспечивают основную функциональность (например, возможность редактировать текстовые файлы или работать с файловой системой).
3. Процессы — внешние процессы, выполняющие разнообразные функции (например лексический, синтаксический и семантический анализ или выполнение моделирования), которые разумно изолировать от оболочки межпроцессным взаимодействием.

На рисунке 8 приведён пример представления каждого слоя архитектуры интегрированной среды разработки SIMODO.

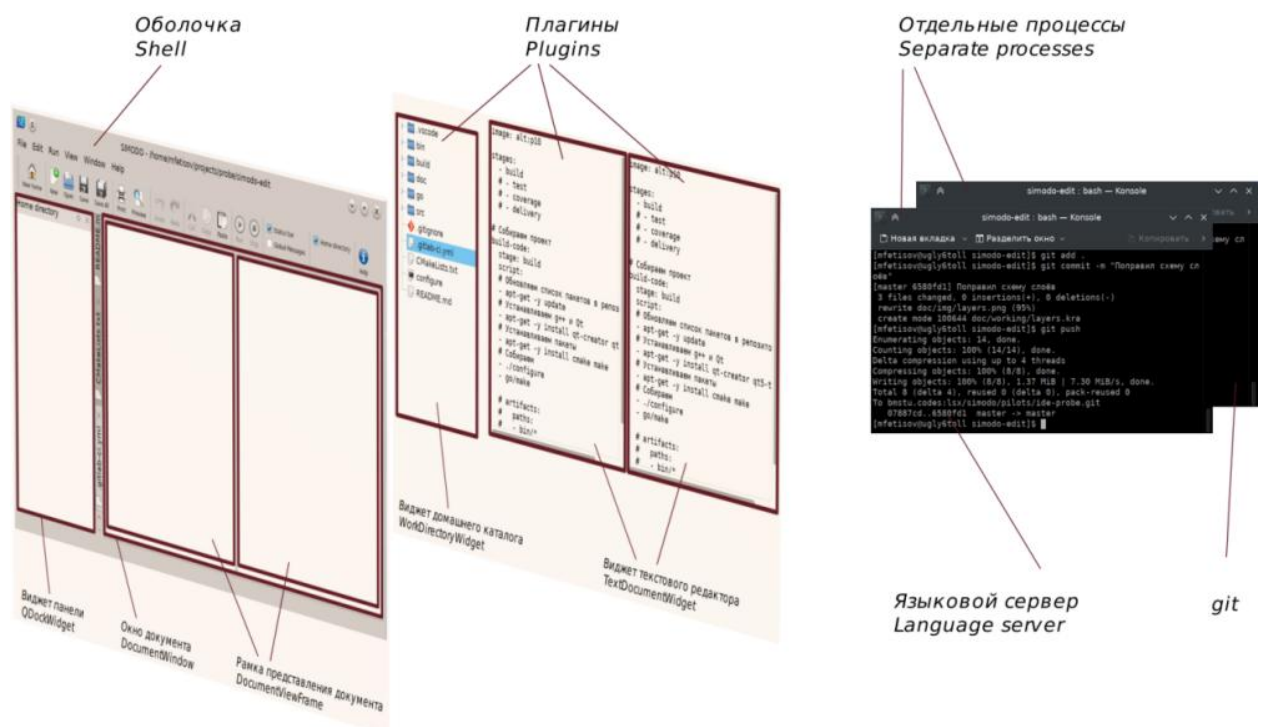


Рисунок 8 — Архитектура интегрированной среды разработки SIMODO

В архитектуре интегрированной среды разработки SIMODO нижние уровни независимы от верхних, а взаимодействие возможно только между смежными уровнями.

Адаптивная система моделирования SIMODO принадлежит к третьему уровню в архитектуре интегрированной среды разработки SIMODO. Среда выполняется в отдельном процессе, с которым взаимодействует плагин расширяемого редактора для передачи исходных текстов моделей, скриптов запуска моделирования или других программных модулей. В свою очередь, среда отправляет информацию о состоянии моделей и ходе моделирования. Среде может понадобиться дополнительная информация (например, модель использует внешние модули), поэтому она может запросить дополнительные ресурсы у плагина (например, исходные тексты внешних модулей).

Запуск и остановка адаптивной системы моделирования SIMODO, запуск, временная и полная остановка моделирования должны программно контролироваться плагином расширяемого редактора.

Таким образом, можно выделить следующие сообщения, передаваемые между плагином расширяемого редактора и процессом адаптивной системы моделирования SIMODO:

- `init`: инициализация адаптивной системы моделирования SIMODO.
- `info`: запрос информации о текущем состоянии адаптивной системы моделирования SIMODO
- `start`: запуск моделирования с передачей исходного текста сценария, описывающего ход моделирования.
- `request`: запрос дополнительного ресурса системой моделирования.
- `process`: передача исходного текста модуля.
- `send`: передача данных.
- `pause`: временная остановка моделирования.
- `resume`: продолжение моделирования.
- `stop`: полная остановка моделирования.
- `terminate`: остановка адаптивной системы моделирования SIMODO

На рисунке 9 изображена диаграмма последовательности сообщений между плагином и адаптивной системой моделирования SIMODO (сервером) при нормальном ходе моделирования, т.е. без возникновения ошибок при обмене сообщениями.

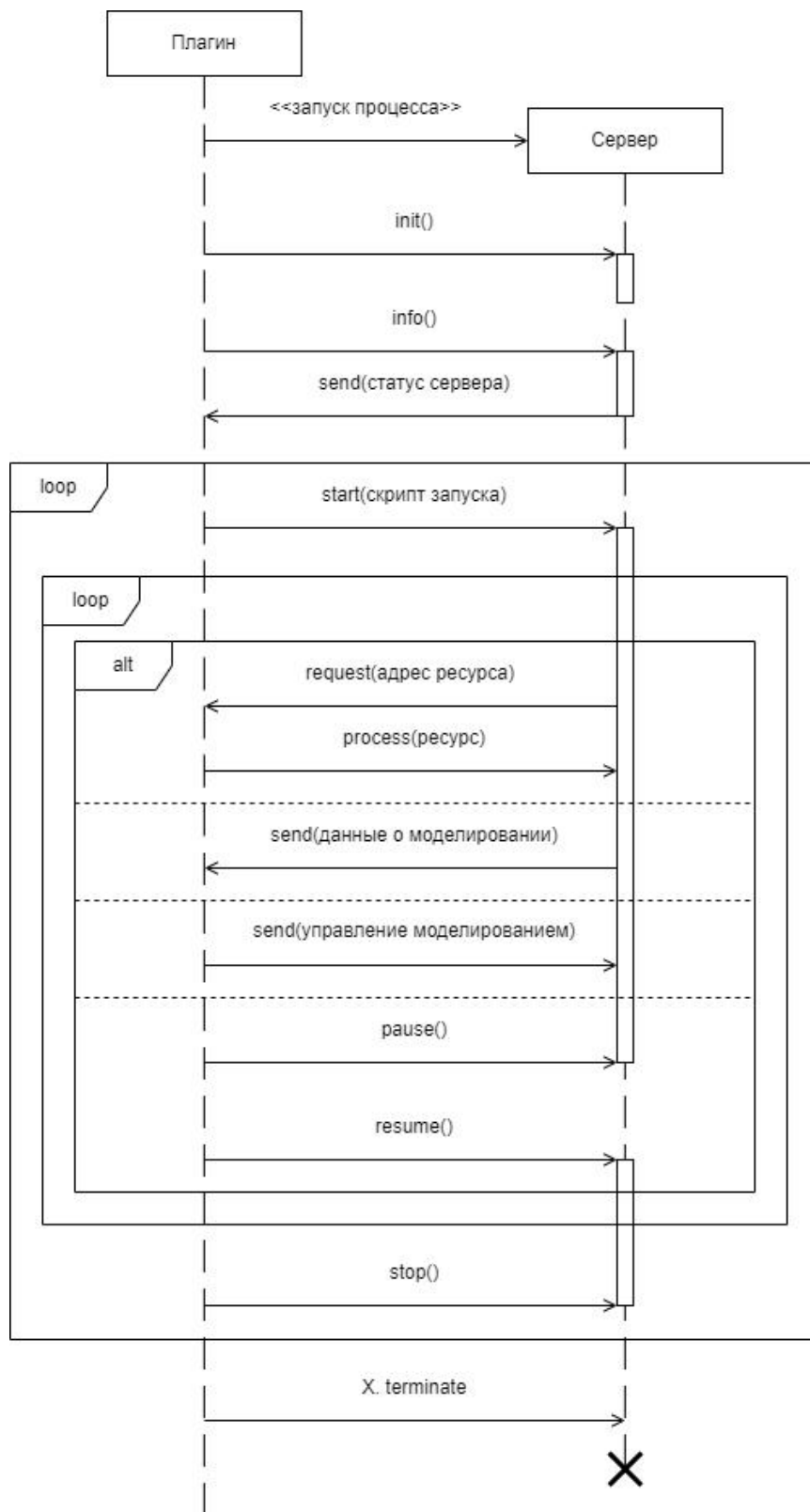


Рисунок 9 — Пример нормального хода моделирования

Основная концепция взаимодействия между процессом плагина и процессом сервера моделирования заключается в том, что после запуска процесса, на котором работает сервер моделирования, плагин инициирует запуск, управление и остановку моделирования, в свою очередь предоставляя серверу необходимые ресурсы.

В первую очередь плагин запускает процесс сервера моделирования, после чего плагин должен отправить сообщение инициализации серверу. О готовности сервера плагин может узнать отправив запрос на получение информации о состоянии сервера. В ответ на запрос сервер отправит запрашиваемую информацию стандартным сообщением данных.

Моделирование запускается на инициализированном сервере моделирования сообщением `start` и передаётся сценарий моделирования. После этого нормальным считается следующее поведение:

- Запрос сервером моделирования дополнительных ресурсов и отправка соответствующих ресурсов плагином.
- Отправка плагину данных о ходе моделирования.
- Отправка серверу моделирования команд управления, которые обрабатываются в соответствии со сценарием моделирования.
- Временная приостановка хода моделирования и возобновления хода моделирования.

Завершение хода моделирования возможно в соответствии с логикой сценария моделирования (не приведено на рисунке 9) или при получении сообщения `stop`.

Сервер моделирования позволяет проводить моделирование неограниченное число раз в общем случае с разными сценариями моделирования.

Заключение

Сравнительный анализ библиотек и фреймворков поддержки межпроцессного взаимодействия, показал, что наиболее подходящей библиотекой для организации межпроцессного взаимодействия между плагином интегрированной среды разработки SIMODO и сервером адаптивной системы моделирования SIMODO является библиотека Boost.

На основе описания открытой архитектуры интегрированной среды разработки SIMODO разработан протокол взаимодействия с адаптивной системой моделирования SIMODO.

В результате научно-исследовательской работы был составлен черновой вариант технического задания на разработку «Модуля поддержки параллельных вычислений для адаптивной системы моделирования».

Список используемых источников

1. SIMODO в репозитории МГТУ им. Н.Э. Баумана. [Электронный ресурс]. URL: <https://bmstu.codes/lxx/simodo> (дата обращения: 07.07.2022).
2. Иванова Г.С., Жильцов А.И., Фетисов М.В., Чулин Н.А., Юдин А.Е. Адаптивная система моделирования. — Автоматизация. Современные технологии, номер 11 за 2020 год, стр. 500.
3. Официальная страница LSP [Электронный ресурс]. — URL: <https://microsoft.github.io/language-server-protocol/> (дата обращения: 30.11.2022)
4. Популярные библиотеки C++ [Электронный ресурс]. — URL: <https://en.cppreference.com/w/cpp/links/libs> (дата обращения: 30.11.2022)
5. Библиотека Boost [Электронный ресурс]. — URL: <https://www.boost.org/> (дата обращения: 30.11.2022)
6. Документация на фреймворк eCal [Электронный ресурс]. — URL: https://continental.github.io/ecal/_api/ecal_root.html (дата обращения: 30.11.2022)
7. Фреймворк ACE [Электронный ресурс]. — URL: <http://www.dre.vanderbilt.edu/~schmidt/ACE.html> (дата обращения: 30.11.2022)
8. Библиотека POCO [Электронный ресурс]. — URL: <https://pocoproject.org/> (дата обращения: 30.11.2022)
9. Документация на фреймворк Qt [Электронный ресурс]. — URL: <https://doc.qt.io/qt-5/> (дата обращения: 30.11.2022)
10. Межпроцессное взаимодействие в Qt [Электронный ресурс]. — URL: <https://doc.qt.io/qt-6/ipc.html> (дата обращения: 30.11.2022)
11. Неименованный канал [Электронный ресурс]. — URL: <https://www.baeldung.com/linux/anonymous-named-pipes> (дата обращения: 30.11.2022)

Приложение А

Черновик технического задания ВКРБ