

ОЦЕНИВАНИЕ ПОКАЗАТЕЛЕЙ КАЧЕСТВА ФУНКЦИОНИРОВАНИЯ СЕРВИС-ОРИЕНТИРОВАННЫХ СИСТЕМ С ТЕМПОРАЛЬНОЙ ЛОГИКОЙ СИНХРОНИЗАЦИИ СЕРВИСОВ

В. В. Нежинский, Л. К. Птицына, П. В. Смирнова

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А.
Бонч-Бруевича

Представлена открытость оценивания влияния темпоральной логики синхронизации сервисов на показатели качества функционирования сервис-ориентированных систем. Выделено направление совершенствования технологического сопровождения сервис-ориентированных систем. Определены показатели качества функционирования сервис-ориентированных систем с темпоральной логикой синхронизации сервисов. Описаны архитектурные особенности разработанного инструментального средства. Представлены клиентские возможности инструментального средства. Проведено подтверждение корректной работоспособности инструментального средства. Выявлена зависимость качества функционирования сервис-ориентированных систем от характеристик темпоральной логики синхронизации сервисов.

сервис-ориентированная система, интеграция, темпоральная логика, качество, оценивание, инструментальное средство

В настоящее время наблюдается значительное увеличение темпов развития наукоемких отраслей, благодаря чему расширяется диапазон сфер применения сервис-ориентированных систем [1, 2, 3]. Одним из перспективных направлений, в котором активно используются системы сервис-ориентированной архитектуры, является сфера предоставления online услуг широкому спектру клиентов, в том числе различным системам ситуационного управления. С каждым годом функциональные возможности, количество задействованных ресурсов, распределенность и общая сложность подобных систем возрастает.

При разнообразной палитре особенностей систем ситуационного управления основное внимание уделяется масштабируемости, асинхронности и оперативности. Главной причиной формирования асинхронных временных разверток являются распределенность ресурсов в сети и причины, непосредственно связанные с характером предметной деятельности системы. Фактор времени оказывает существенное значение на возможности сервис-ориентированных систем, предназначенных для применения в системах диагностирования, контроля и мониторинга; информационно-управляющих комплексах; центрах коллективного пользования высокопроизводительными системами; многофункциональных центрах; управления производством; банковской деятельности; управления воздушным транспортом; электронной коммерции; управления бизнес-процессами; здравоохранения и многих других приложениях.

В исследованиях, посвящённых проектированию и построению систем с сервис-ориентированной архитектурой, анализу показателей качества их функционирования, описанию и исследованию механизмов интеграции сервис-ориентированных систем, вопрос преодоления априорной неопределённости относительно влияния временных профилей интеграции программных компонентов на их эффективность остаётся без ответа [4, 5, 6, 7]. В связи с этим актуализируется объективная необходимость расширения формализаций и средств, позволяющих преодолеть указанную неопределённость.

Предлагаемое расширение формализаций осуществляется за счёт математического обеспечения процесса анализа влияния механизмов синхронизации параллельных вычислительных подпроцессов в системах мониторинга и управления, представленного в [8].

Проводимое расширение средств, позволяющих преодолеть указанную неопределённость, выполняется путём разработки инструментального программного обеспечения, ориентированного на оценивание показателей качества функционирования сервис-ориентированных систем в зависимости от характеристик темпоральной логики синхронизации интегрируемых сервисов.

В качестве исходной информации для анализа механизмов интеграции сервис-ориентированных средств с темпоральной логикой синхронизации используются плотности распределения длительностей исполнения сервисных функций, а также временные ограничения на ожидание для каждого из сервисов.

Помимо систем с темпоральной логикой синхронизации сервисов в приложение включены механизмы параллельной масштабируемой интеграции сервис-ориентированных средств согласно логическим функциям «И», «ИЛИ».

Результат выполнения представляется в виде плотности распределения вероятностей длительности исполнения объединяемых сервисов. На основании этих данных может находиться любая временная динамическая характеристика профиля качества.

При таком подходе сохраняется и методологическое единство профилирования качества разнообразных механизмов интеграции средств в сервис-ориентированной архитектуре. В рамках разработки программного обеспечения предусматривается вычисление значений математического ожидания и дисперсии для каждого анализируемого механизма синхронизации сервисов:

$$E[k] = \sum_{i=1}^k k_i f_i[k_i],$$

$$D[k] = \sum_{i=1}^k (k_i - E[k])^2 f_i[k_i],$$

где $E[k]$, $D[k]$ – математическое ожидание и дисперсия соответственно.

Для реализации приложения выбран язык объектно-ориентированного программирования Java. Он относится к языкам высокого уровня со строгой типизацией данных. Приложения, написанные на языке Java, компилируются в байт-код и исполняются на виртуальной машине – Java Virtual Machine (JVM), что обеспечивает кроссплатформенность разрабатываемого программного обеспечения.

Для реализации визуальной составляющей приложения выбрана платформа JavaFX, поскольку она обладает широкими возможностями для создания графических интерфейсов пользователя. Платформа JavaFX предоставляет инструментарий для разработки интерфейсов приложений для браузеров, мобильных устройств и персональных компьютеров.

Библиотека позволяет значительно облегчить создание визуально привлекательных пользовательских интерфейсов за счет использования файлов fxml и возможности применения каскадных таблиц стилей CSS.

В качестве паттерна проектирования программного обеспечения выбран шаблон MVC (Model–View–Controller). Данная концепция позволяет отделить данные приложения от интерфейса пользователя и логики управления. В структуре «Модель-Представление-Контроллер» представление осуществляет вывод пользовательских данных и обеспечивает взаимодействие пользователя с приложением. Контроллер осуществляет управление и координацию данных модели в ответ на запросы пользователя, модель в свою очередь реагирует на команды контроллера, изменяя свое состояние. Структура программного обеспечения представлена на рис. 3.1.

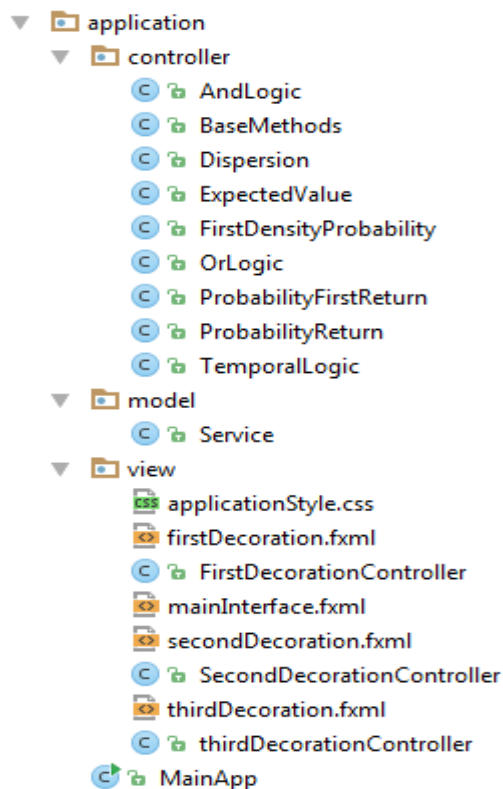


Рис. 1. Структура программного обеспечения

Модель содержит класс Service, представленный на рис. 2. Каждый объект класса содержит данные о временном ограничении на ожидание других сервисов из группы объединяемых, массив значений плотностей распределения вероятностей в дискретные моменты времени, а также методы для получения и установки указанных значений.

К базовым методам контроллера относятся суммирование плотностей распределения вероятностей сервиса от и до некоторых заданных значений, разница единицы и суммы плотностей от и до указанных значений, получение произведения элементов некоторого массива, а также метод расчета индексов для сервиса с учетом значений временных ограничений. Реализация базовых методов представлена на рис. 3 – 6.

```
public class Service {
    private int m; //Значение временного ограничения
    private double f[]; // Значения плотностей распределения вероятностей

    public Service(){}

1   public Service(int m) { this.m = m; }
1   public Service (int m, double f[]){
    |   this.m = m;
    |   this.f = f;
1   }

1   public int getM() { return m; }
1   public double[] getF() { return f; }

    public void setF(double[] f) {this.f = f;}

    public void setM(int m) {this.m = m;}
}
```

Рис. 2. Класс Service

```
public static double sum(Service services, int j, int k) {
    double sum = 0;
    if (k >= 0 && j >= 0) {
        for (; j <= k; j++) {
            sum += services.getF()[j];
        }
    }
    if (k >= 0 && j < 0){
        j = 0;
        for (; j <= k; j++) {
            sum += services.getF()[j];
        }
    }
    return sum;
}
```

Рис. 3. Сумма плотностей распределения вероятностей в указанном диапазоне

```

public static double difference(Service services, int j, int k){
    double diff = 1;
    if(k >= 0 && j >= 0){
        diff -= sum(services, j, k);
    }
    if (k >= 0 && j < 0){
        j = 0;
        for (; j <= k; j++) {
            diff -= services.getF()[j];
        }
    }
    return diff;
}

```

Рис. 4. Разница единицы и плотностей распределения вероятностей в указанном диапазоне

```

public static double multiplication(double multiplier[]){
    double multiplication = 1;

    for(int i = 0; i < multiplier.length; i++){
        multiplication *= multiplier[i];
    }

    return multiplication;
}

```

Рис. 3.5. Произведение элементов массива

```

public static int countInd(Service services, int ind){
    int index = ind - services.getM();
    return index;
}

```

Рис. 6. Получение индекса сервиса с учетом временного регламента

В контроллере представлены методы для получения значений вероятностей первого возврата и вероятности возврата к началу исполнения сервиса в некоторый дискретный момент времени, вероятность успешного завершения сервиса в некоторый момент без учета возможного возврата к началу исполнения сервиса.

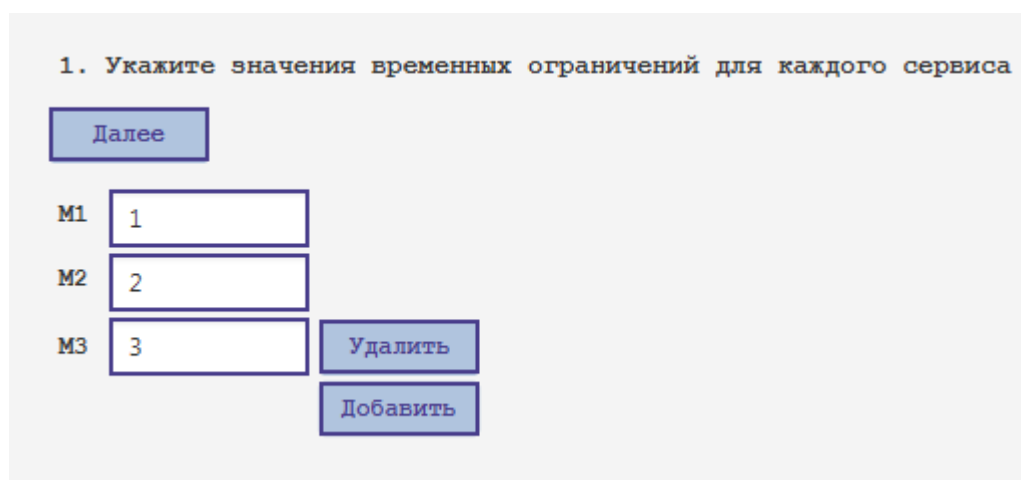
Кроме того, контроллер содержит методы для вычисления вероятностей успешного завершения группы сервисов в дискретный момент времени, учитывающие тип механизма синхронизации, значения математического ожидания и дисперсии.

Визуальную составляющую интерфейса приложения обеспечивают файлы fxml с применением каскадных таблиц стилей CSS. Для каждого шага работы программного обеспечения создается отдельный fxml-файл, содержащий необходимые для визуального ввода и вывода данных

структуры. Для каждого из указанных файлов создается подконтроллер, обеспечивающий связь графического интерфейса с контроллером.

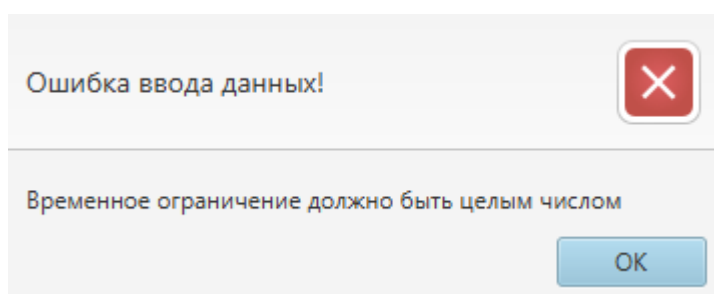
При запуске программного обеспечения открывается окно с полем для ввода значения временного ограничения первого сервиса (рис. 7). Увеличить количество объединяемых сервисов можно при помощи кнопки «Добавить», удаление сервисов происходит по нажатию на кнопку «Удалить». Переход к следующему этапу осуществляется при клике на кнопку «Далее», переход на следующий этап возможен, только в том случае, если все поля заполнены целыми значениями большими нуля, в противном случае пользователю выдается предупреждение (рис. 8).

На следующем шаге пользователю предлагается ввести значения плотностей распределения вероятности в дискретные моменты времени. Сумма плотностей для каждого сервиса должна равняться единице, при необходимости пользователь может увеличить или уменьшить количество моментов времени нажатием кнопки «Добавить» и «Удалить» соответственно.



The screenshot shows a window titled "1. Укажите значения временных ограничений для каждого сервиса". At the top left is a blue button labeled "Далее". Below it are three input fields labeled "M1", "M2", and "M3" on the left. The "M1" field contains the number "1", "M2" contains "2", and "M3" contains "3". To the right of the "M3" field are two blue buttons: "Удалить" (top) and "Добавить" (bottom).

Рис. 7. Окно для ввода временных ограничений



The screenshot shows an error dialog box with a red "X" icon in the top right corner. The text inside the dialog reads "Ошибка ввода данных!" followed by "Временное ограничение должно быть целым числом". At the bottom right is a blue button labeled "ОК".

Рис. 8. Окно с предупреждением

На данном этапе также проверяется корректность вводимых значений и осуществляется вывод предупреждения.

На рис. 9 представлено окно для ввода значений плотностей вероятности в дискретные моменты времени.

На заключительном этапе осуществляется вывод полученных результатов. Результаты вычислений представлены в двух видах – график и таблица полученных значений (рис. 10, 11).

2. Укажите плотности распределения вероятностей для каждого сервиса

Назад Далее

| | | | | | |
|---------|-----|---------|-----|---------|-----|
| F1[k=1] | 0.1 | F2[k=1] | 0.2 | F3[k=1] | 0.3 |
| F1[k=2] | 0.4 | F2[k=2] | 0.5 | F3[k=2] | 0.6 |
| F1[k=3] | 0.4 | F2[k=3] | 0.2 | F3[k=3] | 0.1 |
| F1[k=4] | 0.1 | F2[k=4] | 0.1 | F3[k=4] | 0 |

Удалить
Добавить

Рис. 9. Окно для ввода значений плотностей вероятности

3. Результаты

Исходные данные Результат График

| И | ИЛИ | ТЕМР |
|----------|----------|---------|
| 0.006 | 0.496 | 0.006 |
| 0.309 | 0.489 | 0.309 |
| 0.495 | 0.015 | 0.46836 |
| 0.19 | 0.0 | 0.13854 |
| --- | --- | --- |
| 2.869 | 1.519 | 2.58324 |
| 0.505839 | 0.279639 | 0.47958 |

Рис. 10. Окно вывода численных результатов

Кроме того, пользователю доступны для просмотра введенные ранее данные, на основании которых производились вычисления (рис. 12). Переход к каждому из представлений осуществляется по нажатию на соответствующую кнопку «Исходные данные», «Результат» и «График».

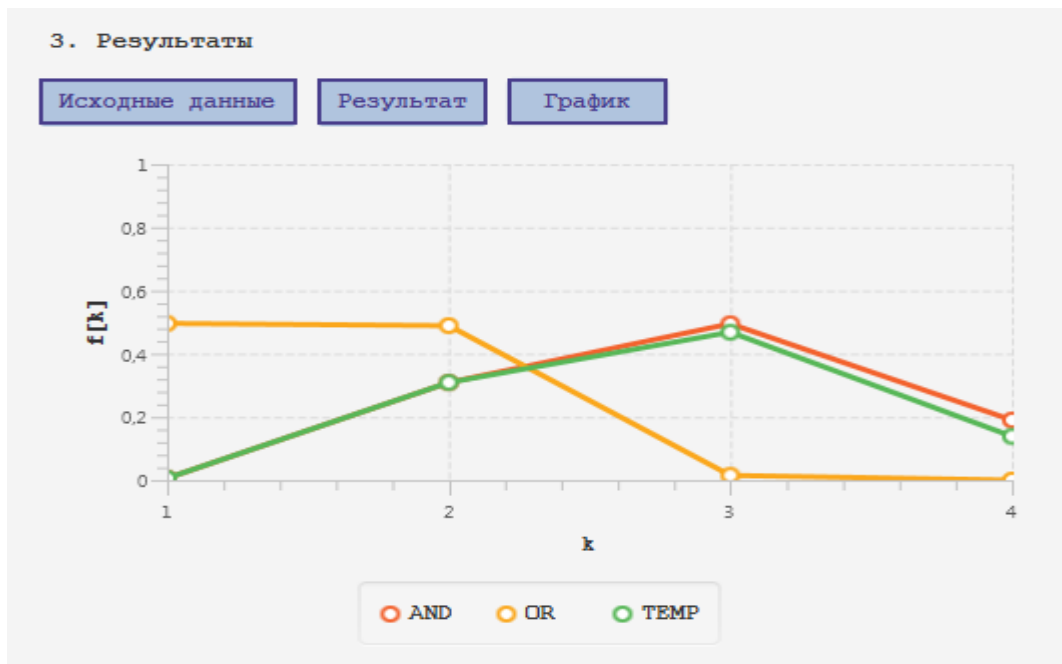


Рис. 11. Окно вывода графика значений

3. Результаты

Исходные данные Результат График

| M_i | 1 | 2 | 3 |
|---------|-----|-----|-----|
| $k = 1$ | 0.1 | 0.2 | 0.3 |
| $k = 2$ | 0.4 | 0.5 | 0.6 |
| $k = 3$ | 0.4 | 0.2 | 0.1 |
| $k = 4$ | 0.1 | 0.1 | 0.0 |

Рис. 12. Окно вывода исходных данных

Разработанное приложение позволяет вычислять значения плотности распределения вероятностей длительности исполнения группы сервисов, а также математическое ожидание и дисперсию для функций объединения по «И», «ИЛИ» и с учетом возможных возвратов к началу исполнения подпроцесса при темпоральной синхронизации сервисов. Результаты вычислений представляются как в численном, так и в графическом виде.

Функционал приложения нуждается в подтверждении работоспособности. Первый этап проверки осуществляется на детерминированных значениях. Рассмотрим единственный сервис, для которого временное ограничение равно 1 и сервис выполняется в первый момент времени с вероятностью 1.

При таких первоначальных значениях для каждого механизма объединения в первый момент ожидается завершение комплексирования с вероятностью 1, математическим ожиданием и дисперсией равными 1 и 0 соответственно.

На рис. 13 и 14 представлены исходные данные и результаты вычисления значений приложением соответственно.

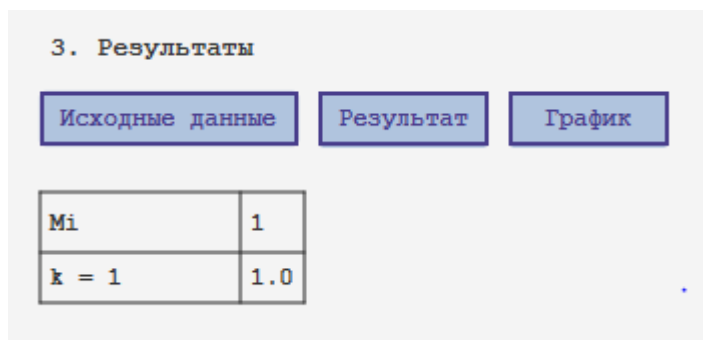


Рис. 13. Исходные данные для первого этапа эксперимента



Рис. 14. Результаты первого этапа эксперимента

По результатам первого этапа эксперимента (рис. 14) видно, что ожидаемые значения полностью соответствуют данным, полученным с помощью программного обеспечения.

Проведем испытания при более сложных ситуациях.

Пусть группу объединения составляют два сервиса с одинаковыми временными ограничениями на ожидание равными двум дискретным моментам времени.

Один из сервисов с высокой вероятностью завершается в первые два момента времени, другой, напротив, с большей вероятностью завершится в последующие моменты.

При такой постановке задачи очевидно, что в первый момент времени значение вероятности успешного объединения сервисов при комплексировании согласно логической функции «И» и с учетом

темпоральной логики будут схожи и несколько ниже, чем при комплексировании согласно логической функции «ИЛИ».

В первые два момента времени при объединении по «ИЛИ» вероятность успеха будет близка к единице, с механизмом объединения группы сервисов по «И» результат близкий к единице будет получен в последующие моменты.

Синхронизация сервисов, учитывающая возможность возврата подпроцесса к началу в первые два момента времени, должна совпадать с механизмом объединения по «И», затем результаты данных функций комплексирования будут отличны друг от друга.

На рис. 15 указаны исходные данные для вычислений, на рис. 16 представлен график полученных по результатам вычислений оценок.

Для третьего этапа эксперимента проведены вычисления оценок показателей по заданным значениям исходной информации, представленным в табл. 3.1 и 3.2.

3. Результаты

Исходные данные Результат График

| | | |
|---------|------|------|
| M_i | 2 | 2 |
| $k = 1$ | 0.05 | 0.7 |
| $k = 2$ | 0.1 | 0.25 |
| $k = 3$ | 0.25 | 0.05 |
| $k = 4$ | 0.6 | 0.0 |

Рис.15. Исходные данные для второго этапа эксперимента

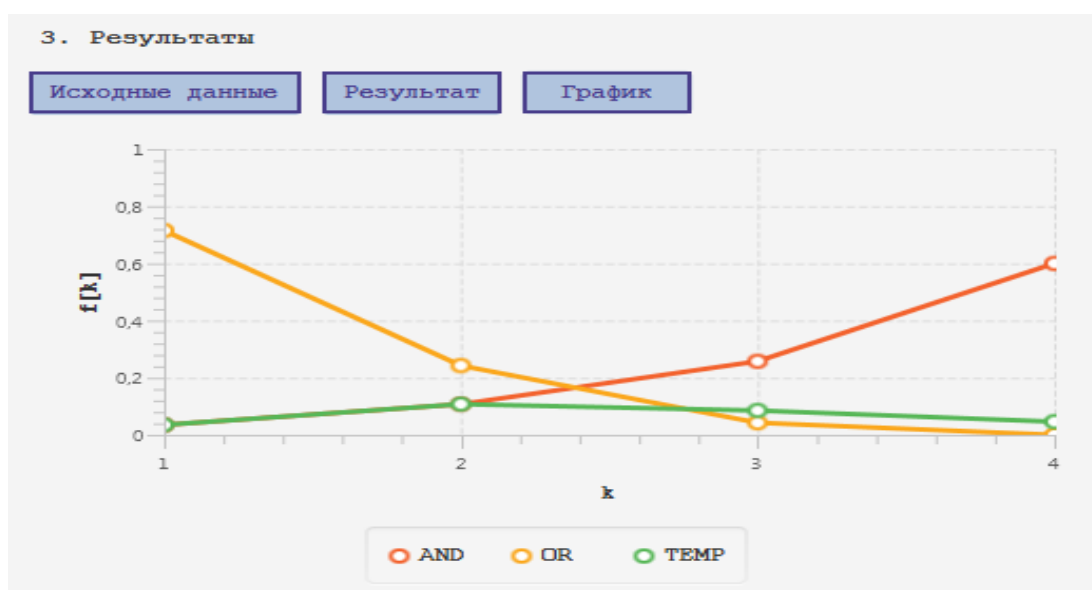


Рис.16. График результатов второго этапа эксперимента

Таблица 1

Исходные данные для третьего этапа эксперимента

| | Сервис 1 | Сервис 2 | Сервис 3 |
|-------|------------|------------|------------|
| M_i | 1 | 2 | 3 |
| k_j | $f_1[k_j]$ | $f_2[k_j]$ | $f_3[k_j]$ |
| 1 | 0.1 | 0.1 | 0.1 |
| 2 | 0.2 | 0.2 | 0.2 |
| 3 | 0.3 | 0.3 | 0.3 |
| 4 | 0.3 | 0.3 | 0.3 |
| 5 | 0.1 | 0.1 | 0.1 |

Таблица 2

Ожидаемые результаты

| | «И» | «ИЛИ» | Темпоральная логика |
|----------|-------|-------|---------------------|
| $f[k_1]$ | 0,001 | 0,271 | 0,001 |
| $f[k_2]$ | 0,026 | 0,386 | 0,018 |
| $f[k_3]$ | 0,189 | 0,279 | 0,09 |
| $f[k_4]$ | 0,513 | 0,063 | 0,149 |
| $f[k_5]$ | 0,271 | 0,001 | 0,055 |
| $E[k]$ | 4,027 | 2,137 | 1,178 |
| $D[k]$ | 0,572 | 0,792 | 2,301 |

На рис. 17 и 18 представлены результаты вычислений с помощью программного обеспечения.

Как следует из анализа представленных результатов вычислений, ожидаемые результаты соответствуют полученным данным.

В качестве дополнительной проверки значительно увеличим значения временных ограничений, остальные входные данные третьего этапа эксперимента остаются без изменений.

При значениях M_1, M_2, \dots, M_7 стремящихся к бесконечности, графики для функции объединения по «И» и функции, учитывающей темпоральную логику, должны совпасть. На рис.19 представлен полученный график.

3. Результаты

Исходные данные Результат График

| И | ИЛИ | TEMP |
|----------|----------|----------|
| 0.001 | 0.271 | 0.001 |
| 0.026 | 0.386 | 0.018 |
| 0.189 | 0.279 | 0.090288 |
| 0.513 | 0.063 | 0.149264 |
| 0.271 | 0.001 | 0.05536 |
| --- | --- | --- |
| 4.027 | 2.137 | 1.18172 |
| 0.572271 | 0.792231 | 2.303258 |

Рис.17. Результаты третьего этапа эксперимента

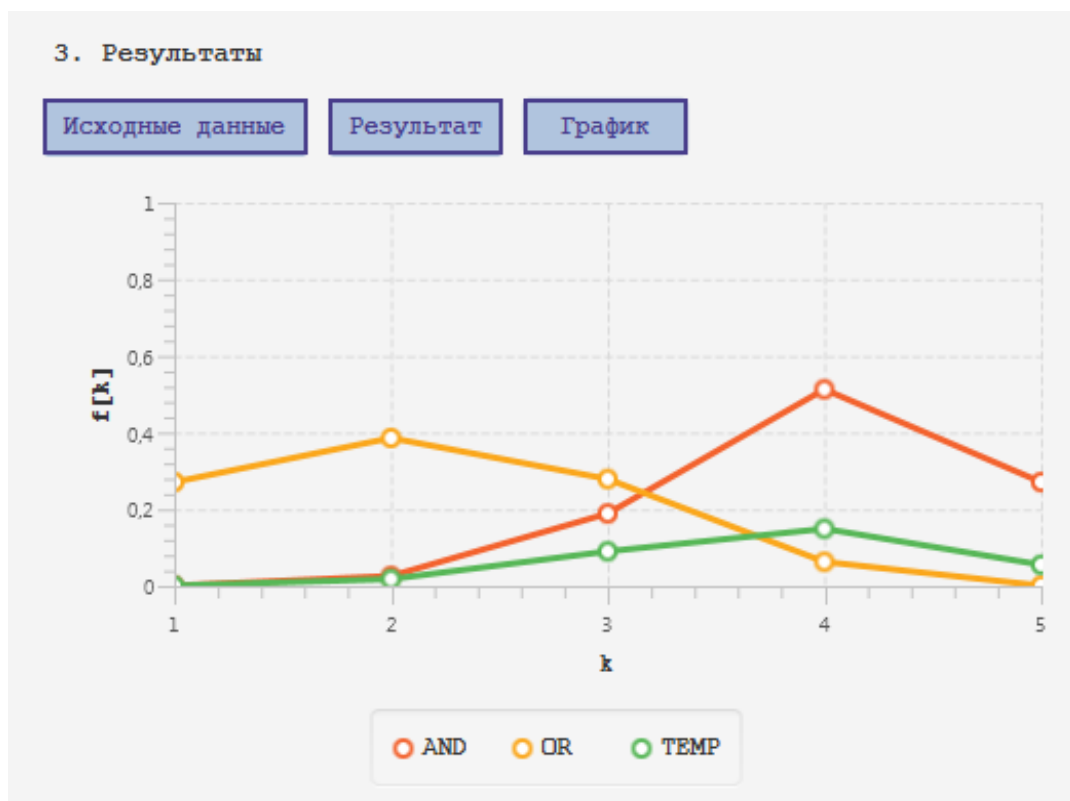


Рис.18. График результатов третьего этапа эксперимента

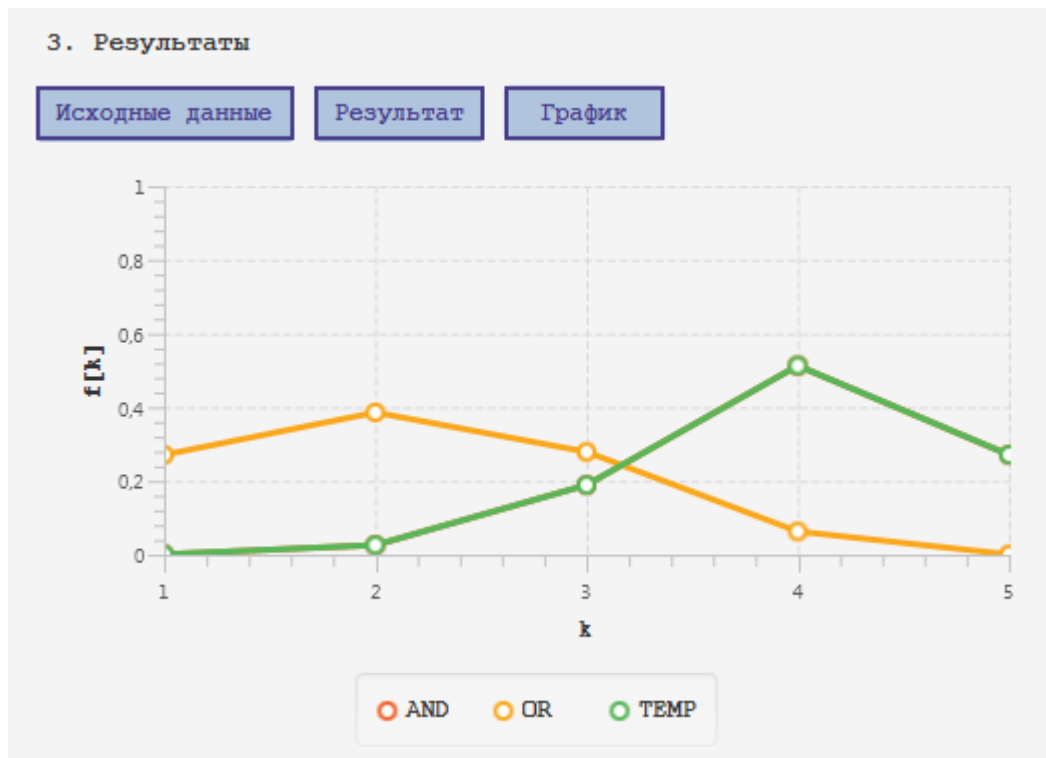


Рис.19. График результатов при увеличенных значениях временных ограничений

Как видно из последнего графика, значения функции, учитывающей возможность возврата к началу исполнения сервисов и функции синхронизации по логической функции «И», полностью соответствуют друг другу, что также подтверждает правильность результатов, полученных с помощью разработанного программного обеспечения.

Корректность вычислений с использованием разработанного приложения полностью подтверждена. Уже при проверке работы программного обеспечения явно прослеживается влияние темпоральной логики на значение плотности распределения вероятностей длительности исполнения группы сервисов, её математическое ожидание и дисперсию.

Разработанное наукоёмкое программное обеспечение позволяет проанализировать влияние характеристик темпоральной логики на объединение сервисов в сервис-ориентированных системах в зависимости от их количества, значений временных регламентов на ожидание для каждого из подпроцессов, а также статистических характеристик временных профилей самих объединяемых сервисов и выбрать наилучшие сочетания с позиций соблюдения требований клиентов.

Список используемых источников

1. Птицына Л. К., Кондратьев Д. А., Эльсабаяр Шевченко Н. Интеллектуальные профили сервис-ориентированных архитектур // 70-ая региональная научно-техническая конференция студентов, аспирантов и молодых учёных «Студенческая весна – 2016»: сб. науч. ст., под ред. К. В. Дукельского; сост. А. Г. Владыко, Е. А. Аникевич, Л. М. Минаков. – СПб. : СПбГУТ, 2016. С. 340 – 344.

2. Птицына Л. К., Кондратьев Д. А., Эльсабаяр Шевченко Н. Моделирование интеллектуальных сервис-ориентированных систем // Информационные системы и технологии в моделировании и управлении: материалы всероссийской научно-практической конференции. Гуманитарно-педагогическая академия (филиал) ФГАОУ ВО «КФУ им. В.И. Вернадского» в г. Ялте; Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», 2016. С. 57-60.

3. Кондратьев Д. А., Птицына Л. К., Эльсабаяр Шевченко Н. Концептуальные модели интеллектуализации сервис-ориентированных архитектур // Актуальные проблемы инфотелекоммуникаций в науке и образовании. V Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 3 т.; Т.2 / под ред. С. В. Бачевского, сост. А. Г. Владыко, Е. А. Аникевич, Л. М. Минаков. – СПб. : Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, 2016. С. 108 – 113.

4. Птицына Л. К., Смирнов Н. Г. Разработка и анализ моделей интеграции сервис-ориентированных средств в гетерогенных сетях // Научно-технические ведомости СПбГПУ 6.1 (138). 2011. С. 71-80.

5. Птицына Л. К., Смирнов Н. Г. Системно-аналитическая основа интеграции сервис-ориентированных средств // Промышленные АСУ и контроллеры. 2011. № 5. С. 31-36.

6. Птицына Л. К., Веселов В. О. Анализ интеграции сервис-ориентированных средств в активных инфокоммуникационных средах // Научоёмкие технологии в космических исследованиях Земли. 2015. Т. 7. № 2. С. 42-47.

7. Веселов В. О., Птицына Л. К. Моделирование интеграции сервис-ориентированных средств в активных инфокоммуникационных средах // Актуальные проблемы инфотелекоммуникаций в науке и образовании. IV Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 2 т.; Т.1 / под ред. К. В. Дукельского, сост. А. Г. Владыко, Е. А. Аникевич, Л. М. Минаков. – СПб. : Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, 2015. С. 465 – 470.

8. Птицына Л. К. Параллельные вычислительные процессы в системах мониторинга и управления : учеб. пособие / Л. К. Птицына. Н. В. Соколова ; Федер. агентство по образованию, С.-Петерб. гос. политехн. ун-т.. СПб. : Изд-во Политехн. ун-та, 2008. 134 с. ISBN 978-5-7422-2066-4.