

Международный журнал информационных технологий и энергоэффективности |



Том 5 Номер 4(18)



2020



СОДЕРЖАНИЕ / CONTENT

ЭНЕРГЕТИКА И ЭНЕРГОЭФФЕКТИВНОСТЬ

1. **Дударовская О.Г.** Энергетическая оценка интенсификации поверхностей нагрева **3**

Dudarovskaya O.G. Energy estimation of heating surface intensification T

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

2. **Дук Г.В., Антонов А. А., Чернышев С. А** Анализ существующих подходов к проектированию имитационных моделей **9**

Duk G.V., Antonov A.A., Chernyshev S.A. Analysis of existing approaches to the design of simulation models

3. **Нагорных М. Э., Быков А.Н., Чернышев С. А.** Кодогенерация на основе формируемой онтологии предметной области **15**

Nagornykh M.E., Bykov A.N., Chernyshev S.A. Code generation based on the generated ontology of the subject area

4. **Балашов О.В., Букачев Д.С** Подход к разработке технологии автоматизированного планирования и оперативного управления организационно-техническими системами **21**

Balashov O.V., Bukachev D.S. Approach to the developing technologies for automated planning and operational management of organizational and technical systems

5. **Федулов Я.А., Тychинская А.М., Якушев В.А., Федулова А.С.** Сравнительный анализ виртуальных ассистентов для разработки программ **33**

Fedulov Ya.A., Tychinskaya A.M., Yakushev V.A., Fedulova A.S. Comparative analysis of virtual assistants for program development



ОТКРЫТАЯ НАУКА
издательство

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 66.045.122

ЭНЕРГЕТИЧЕСКАЯ ОЦЕНКА ИНТЕНСИФИКАЦИИ ПОВЕРХНОСТЕЙ НАГРЕВА

Дударовская О.Г.

ФГБОУ ВО «Казанский государственный энергетический университет», Казань, Россия (420066, г. Казань, ул. Красносельская, 51); e-mail: dg.olga5@mail.ru

Данная работа посвящена математическому описанию тепловых процессов в аппаратах, каналах с насадками. Математическая модель описана на основе применения гидродинамической аналогии и модели турбулентного пограничного слоя C.S. Lin, R.W. Moulton, G.L. Putnam. Оценка эффективности интенсификации теплообмена в каналах с насадками проводилась с использованием энергетического коэффициента, предложенного М.В. Кирпичевым. Аналитическое описание теплового процесса в канале с насадками реализовано в виде программы для ЭВМ, которая позволяет производить тепловой расчет аппаратов, каналов с различными насадками, при различных режимах работы, что в дальнейшем позволяет проводить сравнительный анализ энергетической эффективности различных интенсификаторов в виде насадочных элементов. С целью повышения точности расчетов применяется программное обеспечение MathCad 15.0. Проведенные расчеты энергетических характеристик насадочных элементов позволяют оценить рассматриваемые интенсификаторы теплообмена и определить наиболее оптимальные.

Ключевые слова: теплообмен, интенсификаторы, энергоэффективность

ENERGY ESTIMATION OF HEATING SURFACE INTENSIFICATION

Dudarovskaya O.G.

Federal state-funded educational institution of the higher education "Kazan state power engineering university" Kazan, Russia (420066, Kazan, street Krasnoselsky, 51); e-mail: dg.olga5@mail.ru

This work is devoted to the mathematical description of thermal processes in devices, channels with nozzles. The mathematical model is described based on the application of the hydrodynamic analogy and the turbulent boundary layer model C.S. Lin, R.W. Moulton, G.L. Putnam. Evaluation of the efficiency of heat transfer intensification in channels with packing was carried out using the energy coefficient proposed by M.V. Kirpichev. An analytical description of the thermal process in a channel with nozzles is implemented in the form of a computer program that allows for thermal calculation of devices, channels with different nozzles, under different operating modes, which further allows for a comparative analysis of the energy efficiency of various intensifiers in the form of packed elements. In order to increase the accuracy of calculations, the software MathCad 15.0 is used. The performed calculations of the energy characteristics of the packed elements make it possible to evaluate the considered heat transfer intensifiers and determine the most optimal ones.

Keywords: heat exchange, intensifiers, energy efficiency

В настоящее время доказана высокая эффективность применения насадочных элементов в качестве интенсификаторов теплообмена [1,2]. Однако, в связи с разнообразием

конструкций интенсификаторов в виде насадочных элементов, исследований в этой области еще недостаточно. Ограниченная информация по конструкции различных насадочных элементов приводит к отсутствию обобщенных критериальных уравнений с последующей рекомендацией к практическому использованию, что затрудняет определение наилучшего интенсификатора среди насадочных элементов и проведение последующего расчета оборудования. Поэтому оценка эффективности интенсификации теплообмена в каналах с насадочными элементами считается актуальной и перспективно сложной проблемой, требующей уточнений для лучшего описания процесса.

Данная работа посвящена разработке математической модели, позволяющей проводить исследования различных по геометрической форме насадочных элементов, а также составлять методы и алгоритмы численной реализации поставленных задач в области интенсификации теплообмена.

Коэффициент переноса теплоты

На основе применения моделей турбулентного пограничного слоя и гидродинамической аналогии получим выражения для вычисления коэффициента переноса теплоты для канала с насадками.

Запишем уравнение теплового потока

$$q = -c_p \rho (\alpha + \alpha_T) \frac{dT}{dy}, \quad (1)$$

где α – коэффициент теплоотдачи, Вт/(м²·К); α_T – коэффициент турбулентного обмена, который определяется законом затухания турбулентных пульсаций по толщине вязкого подслоя $\alpha_T = \alpha_T(y)$ и зависит от гидродинамики потока; c_p – теплоемкость среды при постоянном давлении, Дж/(кг·К); ρ – плотность среды, кг/м³.

Поток тепла на стенке можно записать в виде (т.е. $y = 0$)

$$q_{ст} = \alpha (T_\infty - T_{ст}), \quad (2)$$

где T_∞ – температура в ядре потока, К; $T_{ст}$ – температура стенке, К.

Тогда сопротивление переносу тепла имеет вид

$$\frac{1}{\alpha} = \int_0^\delta \frac{q^* dy}{c_p \rho (\alpha + \alpha_T(y))}, \quad (3)$$

где $q^* = q / q_{ст}$ – относительный поток тепла; δ – толщина пограничного слоя, м.

В результате выражение для определения коэффициента теплоотдачи в пограничном слое толщиной δ примет вид

$$\alpha = \frac{c_p \rho}{\int_0^\delta \frac{q^* dy}{\alpha + \alpha_T(y)}}. \quad (4)$$

В литературе представлено большое количество функций $\nu_T(y)$, обычно принимают $\alpha_T \approx \nu_T$.

В данной работе применим модель C.S. Lin, R.W. Moulton, G.L. Putnam [3] турбулентного пограничного слоя, в которой характеристики турбулентного обмена имеют вид

$$\frac{\nu_T}{\nu} = \left(\frac{y^+}{14,5} \right)^3, \text{ где } 0 \leq y^+ \leq 5; \quad (5)$$

$$\frac{\nu_T}{\nu} = 0,2y^+ - 0,959, \text{ где } 5 \leq y^+ \leq 30, \quad (6)$$

где y^+ – безразмерная координата в пограничном слое; ν – коэффициент кинематической вязкости, $\text{м}^2/\text{с}$; ν_T – коэффициент турбулентной вязкости, $\text{м}^2/\text{с}$.

На основе выражения (4) и модели (5), (6) после интегрирования получен коэффициент переноса тепла

$$\alpha = \frac{c_p \rho u_*}{R_1}, \quad (7)$$

где $R_1 = 12,9$ – безразмерная толщина вязкого подслоя; $u_* = u_0 \sqrt{\xi_0/8}$ – динамическая скорость в канале (трубе), $\text{м}/\text{с}$; u_0 – скорость среды в канале (трубе), $\text{м}/\text{с}$.

Используя гидродинамическую аналогию переноса получим

$$a = \frac{c_p \rho u_*}{R_1 \text{Pr}^{n/m}} = \frac{c_p \rho u_0 \sqrt{\xi_0/8}}{12,9 \text{Pr}^{0,57}}, \quad (8)$$

где n, m – значения при числе Прандтля.

С применением выражения (8) запишем безразмерный комплекс Нуссельта для теплоотдачи в канале

$$\text{Nu} = \frac{\text{Re} \text{Pr}^{0,43} \sqrt{\xi_0/8}}{R_1} = \frac{\text{Re} \text{Pr}^{0,43} \xi^{0,5}}{36,5}. \quad (9)$$

Для канала с насадкой характеристики пограничного слоя на стенке отличаются от характеристик плоского пограничного слоя без возмущений на пластине.

Поэтому для определения безразмерной толщины вязкого подслоя в пограничном слое воспользуемся известными свойствами консервативности законов трения и балансовым соотношением переноса импульса

$$\overline{R_1} = R_1 \frac{u_{*0}}{u_*} = R_1 \frac{u_{\text{ср}} \sqrt{\xi_0/8}}{1,85 \frac{\nu}{d_3} \text{Re}_3^{0,75} (\xi/2)^{0,25}}, \quad (10)$$

откуда получаем

$$\overline{R_1} = 1,65 \text{Re}_3^{0,125} / \xi^{0,25}, \quad (11)$$

где u^*o – динамическая скорость в трубе при скорости среды u_0 , м/с; u_{cp} – средняя скорость среды в канале с насадкой, м/с.

Средняя динамическая скорость в канале с насадкой, полученная в работе [4], имеет вид

$$u_* = 1,85 \frac{V}{d_3} \text{Re}_3^{0,75} (\xi / 2)^{0,25}, \quad (12)$$

где d_3 – эквивалентный диаметр насадки, м; Re_3 – число Рейнольдса эквивалентное; ξ – коэффициент гидравлического сопротивления в насадочном слое.

Тогда безразмерный комплекс Нуссельта для теплоотдачи в канале с насадками примет вид

$$\text{Nu}_3 = \frac{1,85 \text{Re}_3^{0,75} (\xi / 2)^{0,25} \text{Pr}^{0,333}}{1,65 \text{Re}_3^{0,125} / \xi^{0,25}}. \quad (13)$$

Определение энергетического коэффициента

Принято оценивать эффективность различных форм конвективных поверхностей при помощи энергетической эффективности [5,6]. Одними из наиболее широко применяемых являются методы, предложенные М.В. Кирпичевым и В.И. Антуфьевым, которые основаны на определении энергетического коэффициента E_d , характеризующего теплогидродинамическое совершенство проведения процесса теплообмена около некоторой поверхности [5].

Для решения поставленной задачи воспользуемся выражением энергетического коэффициента

$$E = \frac{8 \cdot \alpha}{u_{cp}^3 \cdot \xi \cdot \rho}, \quad (14)$$

где α – коэффициент теплоотдачи, Вт/(м²·К); u_{cp} – средняя скорость движения среды в насадке, м/с; ξ – коэффициент гидравлического сопротивления в канале (трубе); ρ – плотность среды, кг/м³.

Таким образом, поставленная задача решается путем определения безразмерного комплекса Нуссельса (т.е. коэффициента теплоотдачи) и коэффициента гидравлического сопротивления ξ .

В данной работе приведем выражения гидравлического сопротивления ξ для некоторых насадочных элементов [7]:

- для насадок кольцеобразной формы

$$\xi = \frac{16}{\text{Re}_3^{0,2}}, \quad (15)$$

- для насадок седлообразной формы

$$\xi = \frac{133}{\text{Re}_3} + 2,34. \quad (16)$$

Результаты расчета

Применим полученные выражения энергетического коэффициента E для выбора наиболее энергоэффективного интенсификатора, в виде хаотичных насадочных элементов.

В качестве интенсификаторов рассматривались следующие насадки [7]:

1) Кольца Палля (разм. $35 \times 35 \times 0,8$) с техническими характеристиками: $d_s = 0,021$ м, $a_v = 139$ м²/м³, $\varepsilon_{св} = 0,9$ м³/м³.

2) Мини кольцевая насадка (разм. $40 \times 4 \times 1$) с техническими характеристиками: $d_s = 0,019$ м, $a_v = 195,3$ м²/м³, $\varepsilon_{св} = 0,92$ м³/м³.

3) Седлообразная насадка Инталокс (разм. 38 мм) с техническими характеристиками: $d_s = 0,017$ м, $a_v = 195$ м²/м³, $\varepsilon_{св} = 0,81$ м³/м³.

Исходные данные:

среда – трансформаторное масло; коэффициент кинематической вязкости масла $\nu = 7,6 \cdot 10^{-6}$ м²/с; коэффициент теплопроводности масла $\lambda = 0,1082$ Вт/(м·К); плотность среды $\rho = 845,7$ кг/м³; число Прандтля $Pr = 111$; число Рейнольдса $Re_s = 320 \div 1050$.

На основе проведенных расчетов построим график зависимости энергетического коэффициента E от числа Рейнольдса Re_s .

На рисунке 1 представлена зависимость энергетического коэффициента E от числа Рейнольдса Re_s для рассматриваемых интенсификаторов.

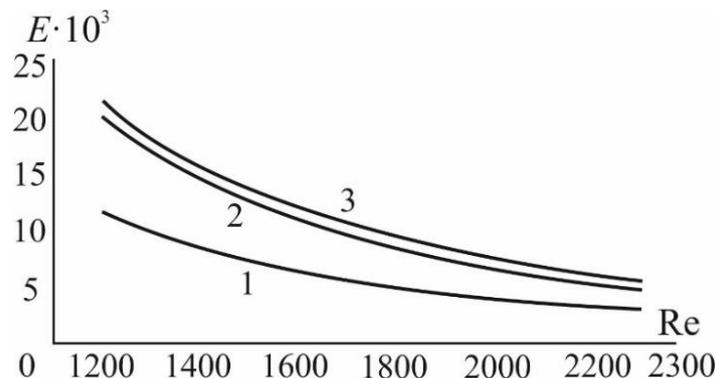


Рисунок 1 – Зависимость энергетического коэффициента E (выражение 14) от числа Рейнольдса Re_s : 1 – кольца Палля ($d_s = 0,021$ м); 2 – мини кольца ($d_s = 0,019$ м); 3 – седла Инталокс ($d_s = 0,017$ м).

Из проведенных расчетных данных следует, что с энергетической точки зрения наиболее выгодно применение, в качестве интенсификатора теплообмена, хаотичных насадочных элементов, состоящих из седел Инталокс. Элементы седлообразных насадок не имеют внутренней полости, но благодаря особенности формы их поверхность смачивается лучше остальных, что и приводит к высоким значениям энергетического коэффициента.

Заключение

Предложенный подход является оптимальным и позволяет выявить потенциальные преимущества того или иного интенсификатора в виде насадочных элементов, а также уточнить их режимные и конструктивные параметры с целью повышения эффективности процесса теплообмена.

Список литературы

1. Дударовская О.Г. Модели интенсифицированного теплообмена и смешения сред в каналах с хаотичными насадочными слоями: дис. канд. техн. наук. Казань, 2016. 202 с.
2. Дударовская О.Г. Анализ эффективности маслоохладителя с различными элементами интенсификации теплообмена // Международный журнал информационных технологий и энергоэффективности. 2018. Т.3, № 3(9). С. 26-31.
3. Lin C.S., Moulton R.W., Putnam G.L. Mass transfer between solid walls and fluid streams, Ind and Eng Chem. 1953. № 45, N 3. pp 636-640.
4. Лаптев А.Г., Фарахов Т.М., Дударовская О.Г. Эффективность явлений переноса в каналах с хаотичными насадочными слоями. Страта: Санкт Петербург, 2016. 202 с.
5. Антуфьев В.М. Эффективность различных форм конвективных поверхностей нагрева. М: Энергия, 1966. 184 с.
6. Кирпичев М.В., Михеев М.А. Моделирование тепловых устройств. М: Изд-во АН СССР, 1936. 320 с.
7. Каган А.М., Лаптев А.Г., Пушнов А.С., Фарахов М.И. Контактные насадки промышленных теплообменных аппаратов. Казань: Отечество, 2013. 454 с.

References

1. Dudarovskaya O.G. Models of intensified heat and mass transfer and mixing of media in channels with chaotic packed layers: dis. ... Cand. tech. sciences. Kazan, 2016.202 p.
 2. Dudarovskaya O.G. Analysis of the effectiveness of an oil cooler with various elements heat transfer intensification // International Journal of Information Technologies and energy efficiency. 2018. V.3, No. 3 (9). pp. 26-31.
 3. Lin C.S., Moulton R.W., Putnam G.L. Mass transfer between solid walls and fluid streams, Ind and Eng Chem. 1953. No. 45, No. 3. pp 636-640.
 4. Laptev A.G., Farakhov T.M., Dudarovskaya O.G. Efficiency of transport phenomena in channels with chaotic packed layers. Strata: St. Petersburg, 2016.202 p.
 5. Antufiev V.M. Efficiency of various forms of convective heating surfaces. M: Energy, 1966.184 p.
 6. Kirpichev M.V., Mikheev M.A. Simulation of thermal devices. M: Publishing house of the USSR Academy of Sciences, 1936.320 p.
 7. Kagan A.M., Laptev A.G., Pushnov A.S., Farakhov M.I. Contact nozzles industrial heat and mass exchangers. Kazan: Fatherland, 2013.454 p.
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.94

АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ К ПРОЕКТИРОВАНИЮ ИМИТАЦИОННЫХ МОДЕЛЕЙ

¹ Дук Г.В., ² Антонов А. А., ³ Чернышев С. А.

^{1,2} Санкт-Петербургский государственный университет аэрокосмического приборостроения, Россия, (190000, г. Санкт-Петербург, ул. Большая Морская 67), e-mail: ¹ger-duk@bk.ru, ²grand.antonov@yandex.ru

³ Санкт-Петербургский государственный университет промышленных технологий и дизайна, Россия, (191186, г. Санкт-Петербург, ул. Большая Морская 18), e-mail: chernyshev.s.a@bk.ru

³ Санкт-Петербургский государственный экономический университет, Россия, (191023, г. Санкт-Петербург, ул. Садовая 21), e-mail: chernyshev.s.a@bk.ru.

В статье рассматриваются основные подходы к проектированию имитационных моделей сложных систем. Проводится анализ существующих методов, областей их применения и этапов формирования модели, а также преимуществ и недостатков. Приводится реальный пример комбинирования двух отдельных методов для более точного моделирования областей, где различают локальные и глобальные события.

Ключевые слова: имитационное моделирование, агентное моделирование, Монте-Карло, системная динамика, дискретно-событийное моделирование

ANALYSIS OF EXISTING APPROACHES TO THE DESIGN OF SIMULATION MODELS

¹ Duk G.V., ² Antonov A.A., ³ Chernyshev S.A.

^{1,2} Saint Petersburg State University of Aerospace Instrumentation, Russian Federation, (190000, Saint-Petersburg, Bolshaya Morskaya str. 67), e-mail: ger-duk@bk.ru, grand.antonov@yandex.ru.

³ Saint Petersburg State University of Industrial Technologies and Design, Russian Federation (191186, Saint-Petersburg, Bolshaya Morskaya str. 18), e-mail: chernyshev.s.a@bk.ru.

³ Saint Petersburg state university of economics, Russian Federation, (191023, Saint-Petersburg, Sadovaya str. 21), e-mail: chernyshev.s.a@bk.ru.

The article discusses the main approaches to the design of simulation models of complex systems. The analysis of existing methods, scopes of their application and stages of model formation, as well as advantages and disadvantages are carried out. A real example of combining two separate methods for more accurate modeling of scopes where local and global events are distinguished is given.

Keywords: simulation modeling, agent-based modeling, Monte Carlo, system dynamics, discrete-event modeling

В настоящее время все компании нацелены на получение прибыли при наименьших затратах. Одним из таких способов является предсказание бизнес-событий. Яркий пример целевой аудитории - биржевые компании. Одним из наиболее популярных и зарекомендовавших себя подходов к представлению процессов и систем - имитационное

моделирование. Имитационная модель – универсальное средство исследования сложных систем, представляющее собой логико-алгоритмическое описание поведения отдельных элементов системы и правил их взаимодействия, отображающих последовательность событий, возникающих в моделируемой системе [1, с. 240]. Такие модели не ограничиваются временем и их можно использовать в различных испытаниях при любых условиях, поэтому они выступают основой имитационного моделирования. А двигателем такого моделирования являются возникающие случайные процессы. Такая модель может работать без вмешательства человека при условии, если в нее были заложены определенные зависимости. Конечный результат системы во многом зависит от используемого подхода к проектированию имитационных моделей. Поэтому очень важно выбрать подходящий подход при разработке системы.

Можно выделить 5 основных подходов [2], используемых для проектирования имитационных моделей:

1. Динамическое моделирование;
2. Системная динамика;
3. Дискретно-событийный подход;
4. Агентное моделирование;
5. Метод Монте-Карло.

Динамическое моделирование [3] позволяет описать возможные факторы и последствия определенного процесса, побуждающих к переходу системы в новое состояние. Цель такого подхода - исследование предприятия, как информационной системы с обратной связью, которая позволяет с каждым шагом совершенствовать формы организации и управления системой. Такие системы описываются алгебраическими и дифференциальными уравнениями, диаграммами и блок-схемами. Выделяют 6 основных этапов формирования имитационной динамической модели [4]:

1. Построения структурной схемы системы;
2. Сбор данных и детализация схемы;
3. Формирования математических моделей;
4. Компоновка и отладка модели;
5. Проверка работоспособности модели;
6. Ввод прогнозирования и других расчетов.

Несмотря на то, что данный метод является одним из первых, он остается достаточно действенным. В основном применяется социально-экономическое развитие, которое позиционируется, как очень важная проблема в современном мире.

Системно-динамический подход (системная динамика) – это способ имитационного моделирования, своими методами и инструментами позволяющий понять структуру и динамику сложных систем [5]. Подход позволяет создавать уменьшенную копию реальных систем с модельным временем, в который есть возможность управлять временем (ускорять, замедлять или останавливать), что дает нам понять и более детально рассмотреть последствия принятых решений. Под модельным временем понимается «сжатое» время, в котором функционирует модель, или, иначе, это время, которое является имитацией времени реальной системы [6]. Формирование модели для такого подхода выполняется в 5 основных этапов:

1. Подготовка и обработка входных данных;
2. Определение и разработка схемы причинно-следственных связей;
3. Задание математической модели;
4. Внедрение баз данных;
5. Тестирование имитационной модели на исходных данных.

Затем выполняется поиск готовых решений или, иными словами, модель применяется на практике. Такой подход на основе причинно-следственных и обратных связей позволяет исследовать поведение сложных систем. У данной модели имеется множество достоинств, среди которых просчет множества вариантов будущего при изменении входных параметров, а также она позволяет выявлять наиболее слабые места системы. Поэтому данный подход нашел применение в различных сферах, таких как:

- экономическая динамика;
- бизнес-динамика;
- городская динамика;
- мировая динамика.

Дискретно-событийный подход [7] (дискретно-событийное моделирование) – это вид имитационного моделирования, при котором система представляется, как хронологическая последовательность событий, позволяющих переводить систему из одного состояния в другое. Основными направлениями являются моделирование бизнес-процессов, логистики, производства, здравоохранения и т.д [8]. Основное отличие от остальных подходов связано с понятием времени. Тут оно представляется в виде отдельных событий, которые несут определённое время и выступают синхронизирующим инструментом (например часы), который порождает все события модели. Можно выделить следующие этапы формирования модели:

1. Анализ предметной области системы;
2. Выявление ключевых процессов;
3. Формирования процессной диаграммы;
4. Подбор оптимального времени события;
5. Тестирование имитационной модели.

Такая модель позволяет пошагово просчитывать, принимать и отслеживать решения имитационной модели на каждом этапе, а также возвращаться и рассматривать другие решения.

Агентное моделирование – это вид имитационного моделирования, предназначенный для исследования поведения децентрализованных агентов и того, как индивидуальное поведение каждого агента влияет на глобальное поведение моделируемой системы в целом [9]. Основой агентного моделирования является понятие «агент» – некий объект, способный принимать решения на основе заложенных правил, взаимодействовать с другими сущностями системы, проявлять активность и самостоятельно действовать. Хорошими примерами агентов, можно выделить социальных агентов (люди, социальные группы, покупатели), экономические агенты (государство, фирмы, регионы), технические агенты (автомобили, роботы, самолеты), экосистемные агенты (облака, пожары, тайфуны) и т.д. В отличие от остальных методов,

глобальное поведение системы складывается из множества локальных поведений каждого агента. Выделяют следующие этапы разработки агентной модели:

1. Анализ предметной области и формирования агентов;
2. задания параметров агентам и их поведения;
3. Создание окружающей среды и добавление в нее агентов;
4. Установка взаимоотношений между агентами;
5. Тестирование системы.

Такой подход применяется в имитационных моделях распределенных систем, где важно влияние всех элементов по отдельности на систему в целом.

Метод Монте-Карло [10] – это вид имитационного моделирования, позволяющий строить модели, описывающие процессы на основе случайных величин. В данном подходе применяется генератор равномерно распределенных случайных чисел в связи с функцией распределения вероятностей изучаемого процесса. Основная суть метода заключается в математическом описании процесса с использованием генератора случайных величин, далее выполняются множество прогонов модели и в результате чего, получаются вероятностные характеристики процесса. Статический анализ вероятностных характеристик позволяет принимать наиболее верные решения. Выделяют следующие этапы формирования модели:

1. Определение и анализ входных параметров;
2. Выявление ключевых факторов риска;
3. Подсчет функции распределения случайных величин;
4. Проведение циклов имитации модели и корректировка параметров;
5. Анализ полученных результатов;
6. Тестирование модели.

Такой подход применяется в сферах, где необходимо просчитывать все возможные риски изучаемой системы, например: в физике, математике, экономике, оптимизации и теории управления.

При моделирование сложных систем, часто возникает ситуация, когда одного из перечисленных подходов недостаточно. В связи с этим, разработчики прибегают к их различным комбинациям. Пожалуй, самым ярким примером является сфера грузоперевозок, где чаще всего применяется комбинация из дискретно-событийного и агентного подхода. В этом случае, дискретно-событийная составляющая отвечает за мониторинг и глобальное взаимодействие элементов системы, что позволяет пошагово и детально ею управлять. А агентное моделирование предоставляет возможность ввести в процесс моделирования интеллектуальных агентов, которые действуют в соответствии с описываемой онтологией и способны путем переговоров найти оптимальное решение ставящихся перед ними задач. Такая комбинация позволяет в довольно короткие сроки находить решения проблемных моментов, при возникновении которых система выходит из равновесия, что делает её довольно мощным инструментом в системах поддержки принятия решений.

Заключение

В статье были рассмотрены подходы имитационного моделирования, а также основные этапы формирования моделей. Выделены основные области применения и подчеркнуты

положительные и отрицательные стороны каждого метода. Например, дискретно-событийный подход, который выполняет все действия в хронологической последовательности, хорошо подходит для формирования бизнес-моделей и производства. А при моделировании систем, где прослеживаются маленькие и большие группы, стоит применить комбинацию из нескольких методов, чтобы получить наилучший результат.

На основе анализа подходов имитационного моделирования, можно заметить, что все методы нашли применение в различных сферах жизни и необходимо серьезно подойти к подбору используемого метода, чтобы получить наилучшие результаты при различных событиях моделируемого процесса.

По результатам исследования можно сделать вывод, что имитационное моделирование является одним из самых эффективных и наглядных видов моделирования при котором исследуемая система заменяется моделью, используемой для предсказания, планирования и с достаточной точностью описания протекающих в системе процессов.

Список литературы

1. Т.И. Алиев. Основы моделирования дискретных систем: учеб. пособие – СПб: ИТМО, 2009. – 363 с.
2. Е.А. Рамзаева, В.Г. Смелов, В.В. Кокарева. Имитационное моделирование производственных систем предприятия *tesnomatixplantsimulaton*: методические указания – Самара: СГАУ, 2013. – 51 с.
3. Д.Форрестер. Основы кибернетики предприятия. Москва: ПРОГРЕСС, 1971. 277 с.
4. С.Г. Светуных, И.С. Светуных. Методы социально-экономического прогнозирования: учеб. пособие СПб: СПбГЭУ, 2010. – 103 с.
5. Л.А. Муравьева-Витковская. Основы распределенного моделирования: учеб. пособие – СПб: НИУ ИТМО, 2013. – 150 с.
6. А.С. Акопов. Имитационное моделирование: учебник и практикум для академического бакалавриата. Москва: Юрайт, 2017. – 389 с.
7. С.В. Чубейко. Дискретно-событийное моделирование сетевых компьютерных систем в условиях высоких нагрузок // *Фундаментальные исследования*. – 2014. – № 8-2. – с. 340-344.
8. В.Н. Сидоренко, А.В. Красносельский. Имитационное моделирование в науке и бизнесе: подходы, инструменты, применение // *Бизнес информатика*. – 2009. – №2. – с. 52-57.
9. Е.В. Стельмашенок. Моделирование процессов и систем: учебник и практикум для академического бакалавриата – Москва: Юрайт, 2018. – 289 с.
10. И.С. Крысанов. Имитационное моделирование на примере метода Монте-Карло симуляции // *Лаборатория фармэкономике ММА им. И.М. Сеченова*. – 2008. – №2. – с. 3-5.

References

1. T.I. Aliev. Osnovy modelirovaniya diskretnykh sistem: ucheb. posobie – St. Petersburg: ITMO, 2009. – 363 p.
 2. E.A. Ramzaeva, V.G. Smelov, V.V. Kokareva. Imitacionnoe modelirovanie proizvodstvennykh sistem predpriyatiya tecnomatixplantsimulaton: metodicheskie ukazaniya – Samara: SGAU, 2013. – 51 p.
 3. D.Forrester. Osnovy kibernetiki predpriyatiya. Moscow: PROGRESS, 1971. 277 p.
 4. S.G. Svetun'kov, I.S. Svetun'kov. Metody social'no-ekonomicheskogo prognozirovaniya: ucheb. posobie St. Petersburg: SPbGEU, 2010. – 103 p.
 5. L.A. Murav'eva-Vitkovskaya. Osnovy raspredelennogo modelirovaniya: ucheb. posobie – St. Petersburg: NIU ITMO, 2013. – 150 p.
 6. A.S. Akopov. Imitacionnoe modelirovanie: uchebnik i praktikum dlya akademicheskogo bakalavriata. Moscow: YUrajt, 2017. – 389 p.
 7. S.V. CHubejko. Diskretno-sobytijnoe modelirovanie setevykh komp'yuternykh sistem v usloviyah vysokih nagruzok // Fundamental'nye issledovaniya. – 2014. – № 8-2. – P.P. 340-344.
 8. V.N. Sidorenko, A.V. Krasnosel'skij. Imitacionnoe modelirovanie v nauke i biznese: podhody, instrumenty, primenenie // Biznes informatika. – 2009. – №2. – P.P. 52-57.
 9. E.V. Stel'mashonok. Modelirovanie processov i sistem: uchebnik i praktikum dlya akademicheskogo bakalavriata – Moscow: YUrajt, 2018. – 289 p.
 10. I.S. Krysanov. Imitacionnoe modelirovanie na primere metoda Monte-Karlo simulyatsii // Laboratoriya farmoekonomiki MMA im. I.M. Sechenova. – 2008. – №2. – P.P. 3-5.
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.058:004.93

КОДОГЕНЕРАЦИЯ НА ОСНОВЕ ФОРМИРУЕМОЙ ОНТОЛОГИИ ПРЕДМЕТНОЙ ОБЛАСТИ

¹ Нагорных М. Э., ² Быков А.Н., ^{3,4} Чернышев С. А.

^{1,2} Санкт-Петербургский государственный университет аэрокосмического приборостроения, Россия (190000, г. Санкт-Петербург, ул. Большая Морская, 67, лит. А), e-mail: ¹ nagornykh_max@mail.ru, ² alexey_bykovoff@mail.ru

³ Санкт-Петербургский государственный университет промышленных технологий и дизайна, Россия (191186, г. Санкт-Петербург, ул. Большая Морская, 18)

⁴ Санкт-Петербургский государственный экономический университет, Россия (191023, г. Санкт-Петербург, ул. Садовая, 21), e-mail: chernyshev.s.a@bk.ru

В статье рассмотрены и проанализированы различные способы и технологии формирования онтологии предметной области, такие как: ER-модель, RDF, OWL и XML-схема. XML-схема более подходит при создании системы кодогенерации поскольку позволяет избежать внесение некорректных или излишних данных посредством механизма валидации XML-документа, заданной XML-схеме. Также рассмотрен процесс кодогенерации на основе полученной онтологии.

Ключевые слова: онтология, кодогенерация, XML-схема, Python

CODE GENERATION BASED ON THE GENERATED ONTOLOGY OF THE SUBJECT AREA

¹ Nagornykh M.E., ² Bykov A.N., ^{3,4} Chernyshev S.A.

^{1,2} Saint Petersburg State University of Aerospace Instrumentation, Russia (190000, Saint Petersburg, Bolshaya Morskaya st., 67, letter A), e-mail: ¹ nagornykh_max@mail.ru, ² alexey_bykovoff@mail.ru

³ Saint Petersburg State University of Industrial Technology and Design, Russia (191186, Saint-Petersburg, Bolshaya Morskaya st., 18)

⁴ Saint Petersburg State University of Economics, Russia (191023, Saint Petersburg, Sadovaya st., 21), e-mail: chernyshev.s.a@bk.ru

The article discusses and analyzes various methods and technologies for forming a domain ontology, such as: ER-model, RDF, OWL and XML-schema. The XML schema is more suitable when creating a codogeneration system because it allows you to avoid entering incorrect or unnecessary data through the XML validation mechanism specified in the XML schema. Also, we reviewed the codogeneration process based on the ontology obtained.

Keywords: ontology, code generation, XML schema, Python.

Введение

В реалиях нашего времени происходит активное развитие IT-сферы. Системы становятся больше и сложнее, из-за чего возникает необходимость автоматизации определенных процессов: систем поддержки принятия решений, онлайн-ботов консультантов, транспортных систем и многих других, строящихся согласно онтологии предметной области. Такой подход позволяет повысить эффективность и мобильность работы, а также упростить процесс взаимодействия между различными элементами разрабатываемых систем.

Онтология — это попытка полного описания области знаний с помощью ее абстрактного представления.[1] Онтология определяется с помощью методов, которые позволяют четко определить ее семантику и содержание. Это дает возможность вычислительным системам корректно определять поведение в различных ситуациях.

Структурно, онтология состоит из классов, объектов-классов и их взаимосвязей, которые, благодаря их методу представления, легко воспринимаются пользователем.

Таким образом, можно сделать вывод о том, способ представления предметной области при помощи онтологии дает возможность сохранить точность используемых понятий, а также их взаимоотношений друг с другом. Это является одним из главных критериев при автоматизации механизмов поддержки принятия решений

Процесс формирования описания предметной области для больших систем очень трудоемкий. А еще более проблемным процессом является описание этой области на понятном компьютеру языке, что и послужило толчком для проводимого исследования – поиск способа упрощения процесса написания кода программных продуктов, в которых для описания предметной области используется онтологический подход.

1. Способы описания онтологии предметной области

В настоящее время, существует множество стандартов для формирования онтологии предметной области, среди которых можно выделить: ER-модель, RDF, OWL, XML-схема.

ER-модель позволяет выделять ключевые объекты, их связи и соответствующие им ограничения. Кроме того, ER-модель обеспечивает независимость модели от способа реализации. Однако, ER-модель является узконаправленной, она решает маленький спектр задач и, преимущественно, используется для баз данных.

Другой стандарт - RDF. Он позволяет работать с метаданными, обеспечивать компьютер семантической информацией и обрабатывать эту информацию автоматически. Все объекты в нем - отдельные сущности, поэтому этап определения объектов и их отношений выполняется с меньшим приоритетом.[2] Однако RDF имеет неудобный синтаксис и крайне громоздкую структуру, что может вызывать неудобство при создании больших систем.

Язык OWL повышает функциональные возможности RDF, обеспечивая при этом совместимость с другими приложениями.[3] Однако, системы, созданные при помощи OWL крайне неудобны при проектировании больших структур и медлительны при работе с ними.[4]

Еще один вариант стандарта для формирования онтологии предметной области - XML-схема. Она формально описывает, что может содержать данный XML-документ, точно так же, как схема базы данных описывает данные, которые могут содержаться в ней: структура таблицы, типы данных, ограничения и другие.

Схема XML определяет форму или структуру XML-документа, а также правила для содержимого данных и семантику, например, какие поля может содержать элемент, какие подэлементы он может содержать и сколько элементов может присутствовать. Она также может описывать тип и значения, которые могут быть помещены в каждый элемент или атрибут. Ограничения данных XML называются фасетами и включают такие правила, как минимальная и максимальная длина. [5]

Несмотря на то, что все перечисленные методы имеют ряд своих преимуществ, и отлично подходят при web-проектировании, стоит выделить, что XML-схема более подходит при создании системы кодогенерации для систем поддержки принятия решений, так как она позволяет избежать избыточность данных и более быстродейственна, благодаря особому виду представления данных - дереву знаний. Кроме того, она позволяет избежать внесение некорректных или излишних данных вследствие валидации XML-документа, заданной XML-схеме.

В общем, кодогенерация — это процесс конвертации какой-либо модели данных в исполняемый код автоматически. Данный процесс значительно оптимизирует и ускоряет процесс создания информационной системы. А главное - она исключает возможность появления неточности в полученном коде.[6]

В различных механизмах кодогенерации этот процесс происходит по-разному. Отличаются, как входные данные, так и сам процесс конвертации.

2. Кодогенерация на основе XML-схемы

Выбранный способ - использование XML-формата, использует в качестве входных данных XML-структуру, которая проходит валидацию, согласно заданной XML-схеме.[7]

Пример листинга используемой XML-схемы приведен ниже:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Actor" type = "ActorFile"/>
  <xs:complexType name = 'ActorFile'>
    <xs:sequence>
      <xs:element name="Description" type="xs:string"/>
      <xs:element name="States" type="StateT"/>
      <xs:element name="Characteristics" type="CharacteristicsT"/>
    </xs:sequence>
    <xs:attribute name="name" type = "xs:string"/>
  </xs:complexType>
<!--States-->
  <xs:complexType name = 'StateT'>
    <xs:sequence>
      <xs:element name="State" type="StateFile" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name = 'StateFile'>
    <xs:attribute name="name" type = "xs:string"/>
    <xs:attribute name="extra_options" type = "xs:string"/>
    <xs:attribute name="description" type = "xs:string"/>
  </xs:complexType>
<!--Characteristics-->
```

```
<xs:complexType name = 'CharacteristicsT'>
  <xs:sequence>
    <xs:element name="Characteristic" type="CharacteristicsFile"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name = 'CharacteristicsFile'>
  <xs:attribute name="name" type = "xs:string"/>
  <xs:attribute name="type" type = "xs:string"/>
  <xs:attribute name="default" type = "xs:string"/>
  <xs:attribute name="description" type = "xs:string"/>
</xs:complexType>
</xs:schema>
```

Как можно наблюдать по первому листингу кода, XML-схема описывает модель объекта с его внутренними параметрами, как: имя, описание, состояние и характеристики. После того, как данные характеристики задаются, формируется XML-файл описания получившегося объекта. Кроме того, следует отметить, что получившийся файл полностью соответствует заданной XML схеме:

```
<Actor name="Car">
  <Description>Human vehicle</Description>
  <States>
    <State name="Go" extra_options="forward, back" description="Car
      rides" />
    <State name="Stop" extra_options="on the handbrake, on transfer"
      description="Car stands" />
  </States>
  <Characteristics>
    <Characteristic name="Color" type="Text Sequence Type"
      default="Red" description="Car color" />
    <Characteristic name="Speed" type="Numeric Type" default="90"
      description="Car speed" />
  </Characteristics>
</Actor>
```

На следующем шаге происходит кодогенерация. Для примера рассматривается использование языка программирования Python. Процесс кодогенерации начинается с импортирования встроенного модуля enum, позволяющего создавать перечисления для описания внутренних состояний класса, которые содержатся в XML-файле внутри узла «States» (Состояния).

Далее идет создание класса, название которого соответствует, атрибуту «name» внутри узла «Actor». Также указывается и описание класса в виде docstring-строки, которое соответствует узлу «Description». Следующий этап - инициализация переменных класса и их описание. Это действие выполняется в функции «init» и отражает в себе значения атрибутов узла «Characteristics». По такой же схеме создается производный класс, наследуемый от класса «Enum», который хранит в себе возможные состояния и их описание из узла «States».

Получившийся в результате кодогенерации листинг кода на языке программирования Python приведен ниже:

```
from enum import Enum

class Car:
    '''Human vehicle'''

    def init(self):
        self.Color = 'Red' # Car color
        self.Speed = 90 # Car speed

class State(Enum):
    Go = 1 # Car rides
    Stop = 2 # Car stands
```

Сгенерированный класс описывает заданные характеристики и состояния объекта согласно его XML-файлу. При этом сам процесс генерации кода можно расширять с помощью добавления новых правил в XML-схему.

Заключение

Анализируя работу созданного кодогенератора, можно сделать вывод, что, благодаря данному способу описания возможно описать любую онтологию любой предметной области. Полученный код легко воспринимается человеком, не знакомым с проектом работы, поэтому дальнейшая разработка становится проще. А достаточно простым расширением XML-схемы можно увеличить границы описания предметной области. Кроме того, валидация полученного файла обеспечивает минимизацию ошибки и целостность данных.

Благодаря такому подходу можно ускорить создание информационных систем и добиться высокого уровня автоматизации трудоемких процессов проектирования.

Список литературы

1. Лапшин В. А. Онтологии в компьютерных системах. — М.: Научный мир, 2010.
2. James Hendler, Dean Allemang, Fabien Gandon. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. — 2-е изд., М.: Association for Computing Machinery and Morgan & Claypool Publishers, 2011.
3. Lee W. Lacy. OWL: Representing Information Using the Web Ontology Language. — М.: Trafford Publishing (uk) Ltd, 2006
4. Michael Uschold. Demystifying OWL for the Enterprise (Synthesis Lectures on Semantic Web: Theory and Technology). — М.: Morgan & Claypool Publishers, 2018
5. XML Documentation// Материалы с сайта <https://www.w3.org> [Электронный ресурс]: – Режим доступа: <https://www.w3.org/TR/xmlschema-0/>, свободный. – Загл. с экрана. – Яз. Англ. Дата доступа: 13 октября 2020
6. Альфред В. Ахо, Моника С. Лам, Рави Сети, Джеффри Д. Ульман. Компиляторы: принципы, технологии и инструментарий — 2-е изд., М.: Вильямс, 2008.
7. Python Documentation// Материалы с сайта <https://www.python.org> [Электронный ресурс]: – Режим доступа: <https://www.python.org/doc/>, свободный. – Загл. с экрана. – Яз. Англ. Дата доступа: 9 сентября 2020

References

1. Lapshin V. A. Ontologies in computer systems. — М.: Scientific world, 2010.
 2. James Hendler, Dean Allemang, Fabien Gandon. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. — 2nd ed., М.: Association for Computing Machinery and Morgan & Claypool Publishers, 2011.
 3. Lee W. Lacy. OWL: Representing Information Using the Web Ontology Language. — М.: Trafford Publishing (uk) Ltd, 2006
 4. Michael Uschold. Demystifying OWL for the Enterprise (Synthesis Lectures on Semantic Web: Theory and Technology). — М.: Morgan & Claypool Publishers, 2018
 5. XML Documentation. Available at: <https://www.w3.org/TR/xmlschema-0/> (accessed 13 October 2020)
 6. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Compilers: Principles, Techniques and Tools. — 2nd ed., М.: Williams, 2008.
 7. Python Documentation. Available at: <https://www.python.org/doc/> (accessed 9 September 2020)
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала: <http://www.openaccessscience.ru/index.php/ijcse/>



УДК 681.3.06

ПОДХОД К РАЗРАБОТКЕ ТЕХНОЛОГИИ АВТОМАТИЗИРОВАННОГО ПЛАНИРОВАНИЯ И ОПЕРАТИВНОГО УПРАВЛЕНИЯ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИМИ СИСТЕМАМИ

Балашов О.В.¹, Букачев Д.С.².

¹ Смоленский филиал АО «Радиозавод», Россия, (214027, г. Смоленск, улица Котовского, 2), e-mail: smradio@mail.ru

² ФГБОУ ВО Смоленский государственный университет, Смоленск, Россия (21400, г. Смоленск, ул. Пржевальского, 4), e-mail: dsbuka@yandex.ru

Обосновывается необходимость разработки технологии автоматизированного планирования и оперативного управления различными организационно-техническими системами. Раскрываются основные компоненты технологии. Рассматриваются программные модули ядра интеллектуальной системы, реализующие предлагаемую технологию.

Ключевые слова: компетенция, решение, лингвистическое описание.

APPROACH TO THE DEVELOPING TECHNOLOGIES FOR AUTOMATED PLANNING AND OPERATIONAL MANAGEMENT OF ORGANIZATIONAL AND TECHNICAL SYSTEMS

¹ Balashov O.V., ² Bukachev D.S.

¹ Smolensk branch of joint-stock company "Radio factory", Russia, (214027, Smolensk, street Kotovskogo, 2), e-mail: smradio@mail.ru

² Federal State Educational Institution of Higher Education Smolensk State University, Smolensk, Russia (21400, Smolensk, street Przewalski, 4), e-mail: dsbuka@yandex.ru

The necessity of developing a technology for automated planning and operational management of various organizational and technical systems is substantiated. The main components of the technology are revealed. The software modules of the core of an intelligent system that implement the proposed technology are considered.

Keywords: competence, decision, linguistic description.

К организационно-техническим системам относятся системы, на объектах которых производится принятие решений о выполнении действий, согласованных с действиями других объектов (систем) или направленных на координацию этих действий. Организационно-технические системы (ОТС) имеют иерархическую структуру. Функционирование ОТС (производственные, строительные, медицинские, военные системы и ряд других) происходит в условиях неопределенности информации, обрабатываемой при принятии решений. Неопределенность информации может быть вызвана её неполнотой, избыточностью, недоста-

верностью, нечёткостью и неточностью [1, 2]. Выделяются два вида условий неопределённости информации, соответственно, условия статистической и нестатистической неопределённости [3]. В качестве условий статистической неопределённости обрабатываемой информации рассматриваются такие условия функционирования ОТС, при которых возможна многократная реализация одних и тех же решений (при фиксированных условиях обстановки), а также сбор и обработка статистической информации о ходе реализации этих решений. Для оценки решений принимаемых в этих условиях используются методические средства теории вероятностей и теории полезности [4, 5, 6]. Условия функционирования ОТС при которых происходит непрерывное изменение обстановки, приводящее к уникальности принимаемых решений, соответствуют условиям нестатистической неопределённости обрабатываемой информации. Для оценки решений принимаемых в этих условиях используются методические средства теории возможностей [7, 8, 9].

Необходимо отметить, что на практике не существует организационно-технических систем функционирующих в условиях статистической неопределённости обрабатываемой информации. Данные условия больше соответствуют условиям функционирования автоматических систем управления. Основными условиями функционирования ОТС являются условия нестатистической неопределённости обрабатываемой информации (отсутствие или неполнота данных о действиях конкурирующей или противоборствующей системы, нечеткость заданной цели функционирования и ряд других).

Наиболее важной проблемой, возникающей при управлении ОТС, является оценка реализуемости принимаемых решений на выполнение действий как при планировании, так и при оперативном управлении системой. Корректное решение данных задач неавтоматизированным способом невозможно в силу больших объёмов и неопределённости обрабатываемой информации при дефиците времени на принятие решений. Автоматизированное решение задач планирования и оперативного управления сдерживается отсутствием на сегодняшний день сформировавшегося подхода к разработке моделей оценки реализуемости решений, принимаемых в условиях нестатистической неопределённости. Проведенные авторами исследования позволили разработать подход, который возможно положить в основу технологии автоматизированного управления ОТС, предусматривающего решение задач разработки плана предстоящих действий, оценку хода его реализации и коррекцию в случае необходимости. В основе подхода к разработке моделей оценки реализуемости решений лежит ситуационный синтез моделей оценки возможностей системы по реализации мероприятий, соответствующих рассматриваемой ситуации её функционирования.

Современные взгляды на проектирование автоматизированных информационных систем (АИС) предполагают решение задач разработки планов предстоящих действий, оценку хода их реализации и коррекцию. Однако, на сегодняшний день разговор о практической реализации данных задач в АИС не более чем декларация, что вызвано отсутствием сформировавшегося теоретического подхода к их решению и как следствие отсутствием технологии автоматизированного управления ОТС. Анализ функционала современных АИС для различных ОТС позволяет сделать следующие выводы:

- наиболее автоматизированными являются процессы доступа к данным и их первичной обработки, под которой понимаются процессы получения, передачи, хранения и оценки достоверности данных;
- открытость АИС различных ОТС имеет место на уровне данных и предполагает наличие жесткого программного обеспечения, предназначенного для первичной обработки данных по заранее оговоренным правилам;
- отсутствует сформировавшийся подход к автоматизированной разработке планов предстоящих действий, обеспечивающий их «понимание» не только АИС объектов рассматриваемой системы, но и АИС других ОТС;
- отсутствует сформировавшийся подход к автоматизированному решению задачи оценки хода реализации мероприятий ранее разработанного плана, и как следствие к решению задачи оперативного управления объектами ОТС в режиме жёсткого реального времени;
- отсутствует сформировавшийся подход к решению задач интеграции различных ОТС и их самоорганизации в критических ситуациях.

В основе существующего подхода к разработке АИС лежит, как правило, использование программных модулей, состоящих из комплекса информационно-расчётных задач. Перечень этих задач формируется таким образом, чтобы максимально автоматизировать работу руководителей на всех уровнях управления рассматриваемой ОТС. Сами задачи реализуются в виде «жёсткого» программного обеспечения, предназначенного для решения задач организации доступа к данным, их первичной обработке, а также для моделирования прогнозируемых состояний ОТС и её объектов.

Для решения задач моделирования прогнозируемых состояний ОТС предлагается использование экспертных систем, объединенных в системы поддержки принятия решений [10, 11]. Однако на сегодняшний день не существует теоретических разработок, обеспечивающих создание системы, управляющей работой СППР, а также отсутствует подход к оценке реализуемости принимаемых человеком решений в условиях нестатистической неопределённости обрабатываемой информации. Отсутствие данных разработок является основной причиной того, что существующие подходы к автоматизации процессов управления в ОТС не ориентированы на реализацию таких функций АИС, как разработка планов предстоящих действий, оценка хода их реализации и коррекция.

Теоретические средства существующих подходов обеспечивают построение информационных систем, обеспечивающих решение первичной обработки информации и лишь частично её вторичной и третичной обработки. Вторичная обработка данных включает в себя процессы формализованного представления текущей или прогнозируемой ситуации функционирования ОТС (или её объектов). Третичная обработка данных включает в себя процессы разработки планов предстоящих действий, оценки хода их реализации и коррекции.

В существующих подходах к проектированию и функционированию АИС большое место уделяется задачам доступа к данным и их первичной обработке. Современные сетевые технологии позволяют решить проблему информационно-технической совместимости, а также переносимости программного обеспечения в АИС (и между ними!) различных ОТС.

Современные технологии управления базами данных, ориентированные на сетевые технологии, позволяют решить проблемы репликативности и иноперабельности данных.

Анализ современных информационных технологий (web-технологии; технологии создания приложений, реализующих программные компоненты для интернет- и интранет-сетей средствами языка Java; объектно-ориентированные технологии обработки данных и т.д.) показывает, что возможности по решению задач первичной обработки данных явно превышают потребности, определяемые существующими подходами к автоматизированному управлению ОТС. Прежде всего, речь идёт о том, что современные информационные технологии не ориентированы на решение задач автоматизированной разработки планов предстоящих действий, оценки хода их реализации и коррекции. Практическим выходом всех существующих информационных технологий является организация оперативного доступа к данным и их первичная обработка. Несомненна высокая ценность этих технологий для пользователей в различных ОТС, однако эта ценность была бы еще выше, если бы эти технологии позволяли бы решать проблемы вторичной и третичной обработки информации. Можно выделить две основные причины сдерживающие развитие информационных технологий в указанном направлении. Во-первых, на сегодняшний день все вопросы связанные с решением задач разработки планов предстоящих действий, оценки хода их реализации и коррекции являются технологическим ноу-хау фирм-разработчиков АИС, что сдерживает решение проблем «открытости», «интеграции» и «самоорганизации», возникающие при автоматизированном управлении. Второй причиной является ранее упоминавшаяся проблема разработки моделей оценки реализуемости решений, принимаемых в условиях нестатистической неопределённости обрабатываемой информации.

Вычислительные ресурсы современных ЭВМ позволяют автоматизировать процессы принятия решений при разработке планов предстоящих действий, а также оценке хода их реализации и коррекции. Формализация указанных задач предполагает обработку сложных типов данных (хронологические ряды, графические образы, мультимедийные данные, а также объединение различных типов данных), что позволяет с требуемым уровнем детализации представлять текущее или прогнозируемое состояние ОТС, а также выполняемые ею действия в памяти ЭВМ. План предстоящих действий отрабатываемый на некотором объекте ОТС представляет собой множество данных, находящихся в различных базах данных, которые создаются в процессе разработки плана. Логика размещения данных в базах данных определяется перечнем и приоритетностью управляющих решений, выполнение которых прогнозируется в каждой из множества ситуаций образующих структуру плана предстоящих действий. Количество и структура баз данных заранее неизвестны, что определяется нестатистичностью решений, принимаемых при разработке планов предстоящих действий.

Предлагаемая технология автоматизированного управления должна обеспечивать разработку планов предстоящих действий, оценку хода их реализации и коррекции по алгоритмам, единым для объектов различных ОТС. Для этого требуется наличие на объектах ОТС интеллектуальных информационных систем (ИИС) с типовым ядром, состоящим из средств вторичной и третичной обработки данных, которые работают по единым правилам. Основными элементами ядра предлагаемой ИИС является ряд программных модулей, показанных на рисунке 1. К ним относятся:

- библиотека типовых программных модулей (ПМ);
- модуль идентификации текущей ситуации,
- модуль планирования;
- модуль оперативного управления;
- система обработки логики управления.

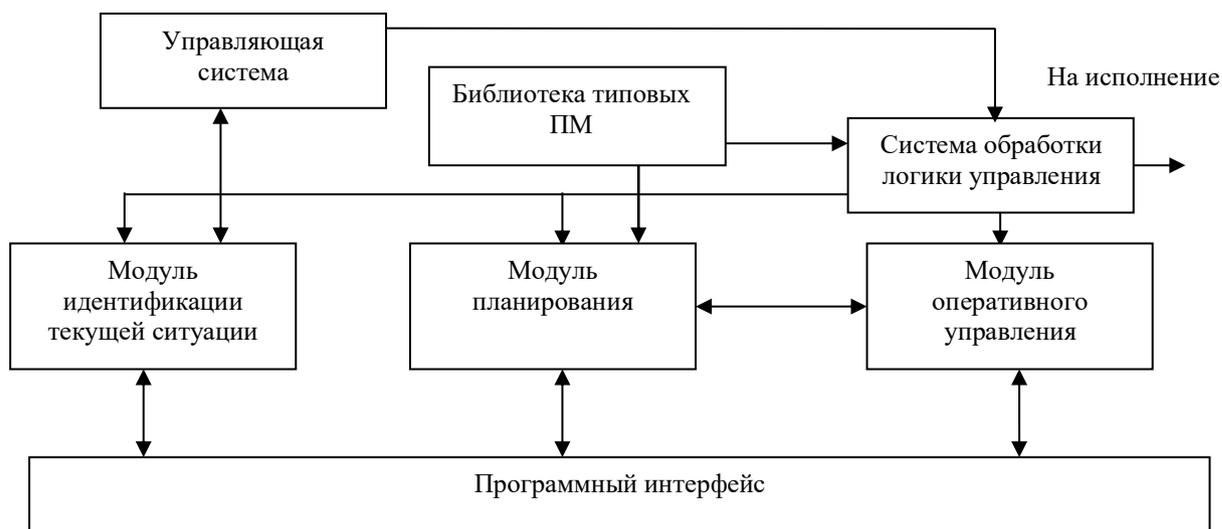


Рисунок 1 – Основные элементы ядра предлагаемой ИИС

Модуль идентификации текущей ситуации осуществляет распознавание текущей ситуации и генерацию управленческих решений о выполнении действий, соответствующих этой ситуации. Основной задачей модуля идентификации текущей ситуации является управление механизмами идентификации текущих значений ситуационных признаков (МИТЗСП), которые являются элементами прикладного программного обеспечения. Основными задачами каждого такого механизма является идентификация текущего значения рассматриваемого ситуационного признака и генерация управляющих решений, соответствующих этому значению. Под ситуационным признаком рассматривается некоторый фактор обстановки для идентификации текущего значения которого необходим анализ предметной области, соответствующей этому фактору. Проведенные исследования позволили выделить три группы ситуационных признаков: соответствующих внешним и внутренним факторам обстановки, а также функциональным возможностям системы. К ситуационным признакам, характеризующим внешние факторы обстановки, относятся признаки, значения которых характеризуют состояние и действия, совершаемые взаимодействующими и конкурирующими системами, а также признаки, значения которых характеризуют состояние окружающей среды.

К ситуационным признакам, характеризующим внутренние факторы обстановки, относятся признаки, значения которых характеризуют состояние рассматриваемой системы и действия, совершаемые её объектами. К возможностным ситуационным признакам относятся признаки, значения которых характеризуют текущие значения функциональных возможностей системы, исходя из текущих значений ситуационных признаков первых двух групп. Перечень возможностных ситуационных признаков определяется функциональным предназначением системы.

Каждый МИТЗСП представляет собой экспертную систему, в задачу которой входит сбор и оценка неопределённости информации, необходимой для идентификации текущего значения ситуационного признака, идентификация текущего значения и генерация управляющих решений, соответствующих этому значению. Оценка неопределенности информации состоит в ее проверке на непротиворечивость, достоверность и полноту. Множество идентифицированных значений ситуационных признаков в фиксированный момент времени образует текущую ситуацию. Иными словами, под текущей ситуацией понимается формализованное представление состояния рассматриваемой ОТС, процессов и условий её функционирования в фиксированный момент времени. Количество управляемых МИТЗСП может меняться, при этом одной из задач рассматриваемого механизма является предоставление человеку возможностей по компоновке структуры ситуации (определение количества рассматриваемых ситуационных признаков). С целью обработки в памяти ЭВМ ситуационных признаков, значения которых имеют различный физический смысл и размерность, ситуации представляются в виде нечётких множеств второго уровня [1, 12]. Под управлением МИТЗСП понимается активизация каждого из множества данных механизмов в соответствии с текущим значением его периода обновления.

Модуль планирования предназначен для формирования вариантов целевой ситуации и вариантов планов. Основной задачей данного модуля является разработка рационального плана предстоящих действий и его формализованное представление в памяти ЭВМ.

Необходимым условием разработки плана является наличие текущей и целевой ситуаций. Идентификация текущей ситуации непрерывно производится соответствующим модулем ядра ИИС. Целевая ситуация представляет собой множество требуемых значений ситуационных признаков, соответствующих состоянию системы, в которое её необходимо перевести. Наличие текущей и целевой ситуаций делает возможным формирование структуры плана предстоящих действий, которое осуществляется механизмом формирования стратегии управления. Под стратегией управления понимается множество базовых (промежуточных) ситуаций, отделяющих текущую ситуацию от целевой. В задачу пользователя должны входить только выбор объекта, его работ и ввод значений их параметров (рисунок 2).

Таким образом, стратегия управления рассматривается как структура плана предстоящих действий. При формировании стратегии управления может быть получено несколько ее вариантов. Каждый из этих вариантов может рассматриваться как структура одного из вариантов плана предстоящих действий.

Разработка плана предстоящих действий состоит в последовательной отработке планов перехода системы из ситуации в ситуацию на «пути» из текущей ситуации в целевую. Для отработки таких планов активизируется механизм разработки плана перехода системы из одной ситуации в другую. Каждый такой план представляет собой виртуальную модель прогнозируемых действий системы при ее переходе из одной ситуации в другую. Виртуальность модели состоит в том, что она представляет собой всевозможные комбинации различного рода реальных моделей, используемых при моделировании прогнозируемых действий системы и условий ее функционирования при переходе из одной ситуации в другую.

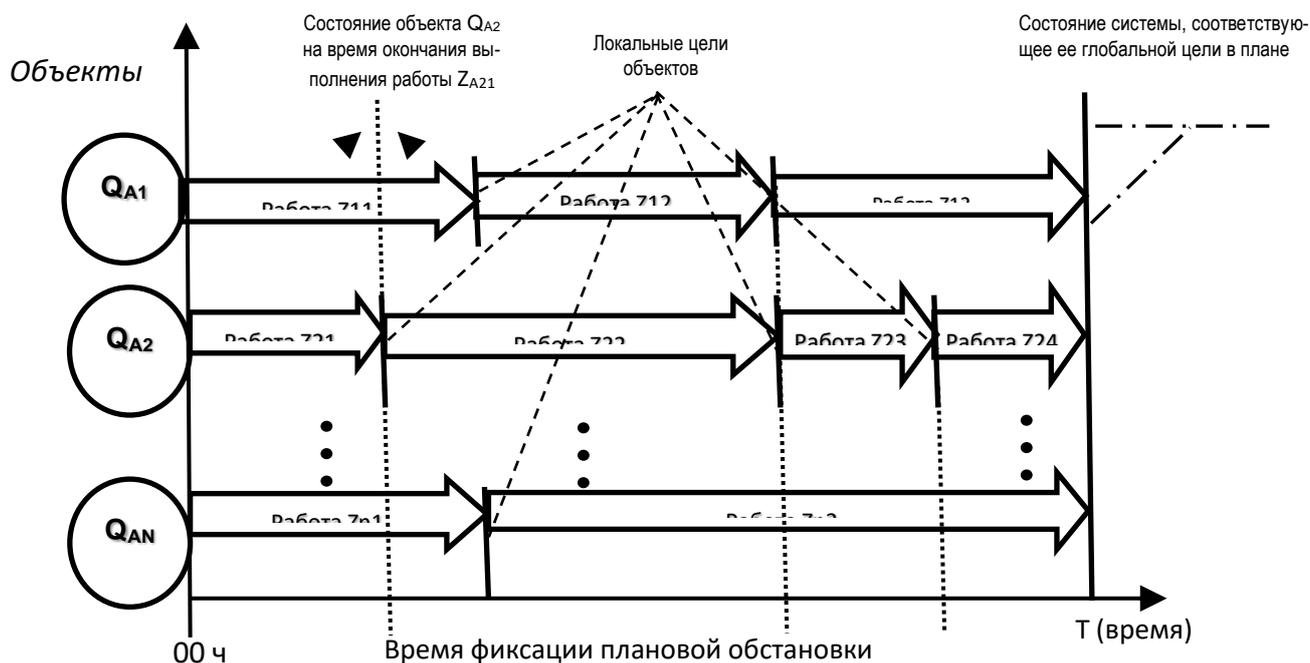


Рисунок 2 – Графическое представление плана работ

Логические связи между реальными моделями определяются содержанием рассматриваемой ситуации, а также содержанием той ситуации в которую необходимо перейти. Так как функционирование организационно-технических систем происходит в условиях нестатистической неопределенности обрабатываемой информации о состоянии системы и условий её функционирования, то эти связи заранее неизвестны и формируются непосредственно в процессе разработки плана. Исходя из этого, до начала разработки плана предстоящих действий неизвестны модели реализации мероприятий, образующих содержание этого плана, а также количество, структура и объем баз данных, соответствующих этим моделям.

Таким образом, количество и объем баз данных соответствующих некоторому варианту плана предстоящих действий определяется количеством ситуаций, образующих структуру рассматриваемого варианта, а также количеством и технологической сложностью управляющих решений, обеспечивающих перевод системы из текущей ситуации в целевую. Структура используемых баз данных определяется содержанием рассматриваемых действий и формируется из структур эталонных баз данных, создаваемых на этапе проектирования ИИС и корректируемых в процессе её эксплуатации. Для обработки различных вариантов плана предстоящих действий в состав механизма его разработки должны быть включены средства идентификации баз данных, соответствующих тому или иному варианту плана.

Каждому из множества планов перехода системы из одной ситуации в другую, образующих содержание плана предстоящих действий, соответствуют значения показателей, характеризующих возможности и необходимость этого перехода. Таким образом, в результате разработки плана формируются распределения значений возможностей и необходимости перехода системы из текущей ситуации в целевую. Обработка данных распределений позволяет получить значения показателей, характеризующих возможность и необходимость достижения целевой ситуации. Предлагаемый подход к разработке планов предстоящих действий

предполагает реализацию механизма перепланирования по ситуациям, ситуационным признакам и действиям с целью достижения требуемых значений возможности и необходимости достижения целевой ситуации.

Один вариант плана может отличаться от другого прогнозируемыми результатами действий, стратегией управления и целевой ситуацией. Выбор наиболее рационального варианта производится в результате анализа значений показателей, характеризующих возможность и необходимость достижения целевой ситуации. Анализ этих же показателей лежит в основе алгоритмов оценки целесообразности сохранения стратегии управления и целевой ситуации в процессе оперативного управления системой.

Модуль оперативного управления работает в ходе реализации плана и проводит оценку соответствия текущей ситуации соответствующей ей ситуации в ранее сформированном плане, и вырабатывает корректирующие решения. Он предназначен для решения следующих основных задач: сравнение текущей ситуации с соответствующей ей ситуацией в ранее сформированном плане; активизация модели ранее сформированного плана перехода системы из одной ситуации в другую; ситуативный синтез виртуальной модели плана перехода системы из текущей ситуации в требуемую; оценка целесообразности сохранения стратегии управления и целевой ситуации; интеграция системы с другими системами.

Целью сравнения текущей ситуации с соответствующей ей ситуацией в ранее сформированном плане предстоящих действий является определение степени соответствия рассматриваемых ситуаций и определения тех управляющих решений и ситуационных признаков которые вызывают это несоответствие. В случае выявления несоответствия производится коррекция плана перехода из текущей ситуации в следующую. В случае необходимости может производиться коррекция всего плана предстоящих действий или его отдельных фрагментов, а также корректироваться стратегия управления и целевая ситуация. В случае, если коррекция не нужна производится активизация модели ранее сформированного плана этого перехода.

Под активизацией модели ранее сформированного плана перехода системы из одной ситуации в другую понимается формирование в памяти ЭВМ виртуальных моделей реализации управляющих решений, образующих содержание этого плана, и их последовательное выполнение в соответствии с логикой плана. При этом под выполнением понимается не только выполнение моделей, но и практическое управление объектами системы (выдача команд управления, информационный обмен и т.д.).

Ситуативный синтез виртуальной модели плана перехода системы из текущей ситуации в требуемую имеет место при выявлении несоответствия между текущей ситуацией и соответствующей ей ситуацией в ранее сформированном плане предстоящих действий. В зависимости от степени несоответствия может быть произведена коррекция моделей реализации одного или нескольких управляющих решений, образующих содержание плана перехода из текущей ситуации в другую, или заново разработан этот план. Процесс коррекции осуществляется человеком посредством активизации механизма разработки плана перехода системы из одной ситуации в другую. Основным отличием ситуативного синтеза подобных планов в процессе функционирования системы от выполнения этого же процесса в период планирования предстоящих действий системы является дефицит времени, который приводит к невозможности формирования альтернативных вариантов плана перехода из текущей ситуации в требуемую и выбора рационального варианта. В случае выявления несоответствия между текущей

ситуацией и соответствующей ей ситуацией в ранее сформированном плане предстоящих действий, помимо коррекции плана перехода системы из текущей ситуации в требуемую, производится оценка целесообразности сохранения стратегии управления и целевой ситуации. Оценка целесообразности сохранения стратегии управления производится посредством активизации механизма формирования стратегии управления. Суть процесса оценки состоит в выявлении возможного изменения стратегии управления, как множества базовых ситуаций, вызванного несоответствием текущей ситуации с соответствующей ей ситуацией в ранее сформированном плане. Необходимо выделить два крайних случая приводящих к необходимости изменения стратегии управления. Первый случай может быть вызван низкими значениями возможностей системы по ее переходу из текущей ситуации в требуемую при условии совпадения значения ситуационных признаков в текущей ситуации со значениями этих же признаков в соответствующей ситуации ранее сформированного плана. Второй случай имеет место когда текущая ситуация отличается от соответствующей ей ситуации в ранее сформированном плане по значениям ситуационных признаков. В результате оценки целесообразности сохранения стратегии управления формируется стратегия управления соответствующая текущей ситуации. Формирование этой стратегии управления производится с использованием механизма формирования стратегии управления. Изменение стратегии управления вызывает необходимость разработки нового плана предстоящих действий непосредственно в процессе функционирования системы. Необходимость изменения целевой ситуации может иметь место когда текущая стратегия управления не обеспечивает достижение целевой ситуации в виду отсутствия или низких возможностей системы по переходу из текущей ситуации в целевую.

Система обработки логики управления предназначена для решения следующих задач: кодирование и декодирование логики обработки объектов данных; кодирование и декодирование логики плана предстоящих действий. В основе работы данной системы лежит объектно-ориентированный подход к представлению данных. Выделяются три типа объектов данных:

- данные, необходимые для использования реальных моделей, а также формализованное представление логики обработки этих данных;
- атрибуты реальных моделей, количество и сложность которых определяется функциональным назначением объекта или системы (для ее пункта управления);
- логика формализации виртуальных моделей (управляющий код).

Любое данное рассматривается как объект данных первого типа. В случае, если объект данных первого типа является комбинированным (совокупность данных различного типа) этому объекту ставится в соответствие код логики обработки этих данных. Код представляет собой совокупность данных как объектов и их атрибутов, связанных определенными отношениями, характеризующими способ объединения этих данных. Данный код также представляется в виде объекта данных первого типа.

Объект данных второго типа имеет соответствующие атрибуты (могут рассматриваться, как объекты), позволяющие использовать некоторую реальную модель (как объект) при синтезе виртуальных моделей. В качестве атрибутов объекта данных второго типа могут рассматриваться: имя реальной модели, ее размещение во внешней памяти, объем кода, имена и расположение подключаемых баз данных и ряд других.

Объект данных третьего типа представляет собой управляющий код, связывающий объекты данных первого и второго типа определенными отношениями, формализующими

последовательность и условия манипуляции этими объектами при синтезе виртуальных моделей. Помимо этого, объекты данных третьего типа имеют ряд атрибутов (объектов данных первого типа), формализующих основные элементы плана: план; объект-разработчик плана; перечень ситуационных признаков, необходимых для синтеза виртуальной модели плана предстоящих действий; виртуальная модель плана перехода системы из ситуации в ситуацию; виртуальная модель плана реализации управляющего решения; приоритетность, последовательность и условия реализации управляющих решений; и ряд других. Синтез управляющего кода производится в процессе разработки плана предстоящих действий и его реализации.

В задачу системы обработки логики управления входит отслеживание процессов разработки плана предстоящих действий и его реализации, формализация логики этих процессов, а также представление этой логики в виде управляющего кода (объекта данных третьего типа).

Таким образом, виртуальный план предстоящих действий системы состоит из некоторого множества реальных моделей и множества объектов-данных первого типа, необходимых для создания виртуальных моделей реализации управляющих решений, образующих содержание плана, а также управляющего кода определяющего последовательность и условия выполнения управляющих решений. Иными словами, во внешнюю память ЭВМ записывается не программный код виртуальной модели плана (хотя, в принципе, это возможно), а сравнительно небольшой управляющий код и данные, необходимые для формирования виртуальной модели плана в оперативной памяти ЭВМ.

Код логики обработки объектов данных первого типа, атрибуты объектов данных второго типа, отношения и атрибуты объектов данных третьего типа являются элементами одного языка описания логики управления. Данный язык предназначен для формализованного представления логики разработки и реализации плана предстоящих действий. Таким образом, данный язык позволяет формализовать логику процессов планирования и оперативного управления. В основу разработки данного языка могут быть положены основные положения теории формальных грамматик и, в частности, КС-грамматик [13].

Управляющая система предназначена для управления работой ИИС. Основными ее задачами являются управление процессами, оценка их приоритетности, контроль и управление вычислительными ресурсами. Под процессами понимается выполнение интеллектуальной системой определенных действий, связанных с разработкой и реализацией плана предстоящих действий, обработкой данных, управлением вычислительной аппаратурой, информационным

обменом и рядом других действий. Управляющая система должна представлять собой совокупность логических механизмов, обеспечивающих решение указанных задач. Алгоритмы работы управляющей системы должны быть адаптивными к технической оснащенности объекта (прежде всего конфигурацией и характеристиками вычислительной системы и системы передачи данных), а также к содержанию системного и прикладного программного обеспечения.

Таким образом, по сравнению с существующей, предлагаемая технология направлена на автоматизацию процессов планирования и оперативного управления. Целью разработки методологии является создание типовых программных компонент комплексов средств автоматизации управления, позволяющих автоматизировать и унифицировать процессы управления в АИС систем различного назначения.

Основной идеей предлагаемой технологии является предоставление пользователю

возможностей по манипулированию действиями объектов в процессе планирования и оперативного управления действиями объектов. В задачу пользователя должны входить только выбор действия объекта (его названия) и ввод значений его параметров. В задачу ЭВМ должно входить: отображение моделируемой обстановки, выполнение необходимых расчётов, оценка и выдача данных о прогнозируемом качестве выполнения действия, формирование и выдача рекомендаций по изменению значений параметров или обработку действий других объектов и ряд других задач.

Список литературы

1. Аверкин А.Н., Батыршин И.З., Блишун А.Ф. Нечеткие множества в моделях управления и искусственного интеллекта/ Под ред. Поспелова Д.А. - М.: Наука. Гл. ред. физ.-мат. лит., 1986. -312 с.
2. Борисов А.Н., Алексеев А.В., Меркурьева Г.В. и др. Обработка нечеткой информации в системах принятия решений. - М.: Радио и связь, 1989. - 304 с.
3. Поспелов Д.А., Сильдмяэ И.Я. Рольевые структуры в представлении знаний и в диалоговых системах. - Изв. АН СССР. Техническая кибернетика. 1985, № 5.
4. Кайберг Г. Вероятность и индуктивная логика. -М.: Прогресс, 1978. - 375 с.
5. Козелецкий Ю. Психологическая теория решений. - М.: Прогресс, 1979. - 504 с.
6. Фишберн П.К. Теория полезности для принятия решений. - М.: Наука, 1978. -352 с.
7. Борисов А.Н., Аппен Е.П. Оценка возможностей характеристик при анализе альтернатив. - В кн.: Методы принятия решений в условиях неопределенности. Рига: РПИ, 1980.
8. Дюбуа Д., Прад А. Теория возможностей//Приложения к представлению знаний в информатике. - М.: Радио и связь, 1990. -287 с.
9. Ягер Р.Р. Нечёткие множества и теория возможностей. Последние достижения. Пер.с англ. М.: Радио и связь, 1986.
10. Борисов А.Н., Вилломс Э.Р., Сукур Л.Я. Диалоговые системы принятия решений на базе мини-ЭВМ//Информационное математическое и программное обеспечение. Рига. Зинатне. 1986.
11. Трахтенгерц Э.А. Методы генерации, оценки и согласования решений в распределенных системах поддержки принятия решений. - Автоматика и телемеханика. 1995, № 4.
12. Мелихов А. Н., Берштейн Л. С., Коровин С. Я. Ситуационные советующие системы с нечеткой логикой. М., Наука, 1990. -272 с.
13. Клыков Ю.И. Ситуационное управление большими системами М., "Энергия", 1974.

References

1. Averkin A.N., Batyrshin I.Z., Blishun A.F. Nechetkie mnozhestva v modelyah upravleniya i iskusstvennogo intellekta/ Pod red. Pospelova D.A. - M.: Nauka. Gl. red. fiz.-mat. lit., 1986. - 312 p.
2. Borisov A.N., Alekseev A.V., Merkur'eva G.V. i dr. Obrabotka nechetkoj informacii v sistemah prinyatiya reshenij. - M.: Radio i svyaz', 1989. - 304 p.
3. Pospelov D.A., Sil'dmyae I.YA. Rolevyeh struktury v predstavlenii znaniy i v dialogo-vyh sistemah. - Izv. AN SSSR. Tekhnicheskaya kibernetika. 1985, № 5.

4. Kajberg G. Veroyatnost' i induktivnaya logika. -M. :Progress,1978. - 375 p.
 5. Kozeleckij YU. Psihologicheskaya teoriya reshenij. - M. :Progress, 1979. - 504 p.
 6. Fishbern P.K. Teoriya poleznosti dlya prinyatiya reshenij. - M. :Nauka, 1978.-352 p.
 7. Borisov A.N., Appen E.P. Ocenka vozmozhnostnyh harakteristik pri analize al'terna-tiv. - V kn.: Metody prinyatiya reshenij v usloviyah neopredelennosti. Riga: RPI, 1980.
 8. Dyubua D., Prad A. Teoriya vozmozhnostej//Prilozheniya k predstavleniyu znaniy v informatike. - M. :Radio i svyaz', 1990. -287 p.
 9. YAger R.R. Nechyotkie mnozhestva i teoriya vozmozhnostej. Poslednie dostizheniya. Per.s angl. M. : Radio i svyaz', 1986.
 10. Borisov A.N., Vilyums E.R., Sukur L.YA. Dialogovye sistemy prinyatiya reshenij na ba-ze mini-EVM//Informacionnoe matematicheskoe i programmnoe obespechenie. Riga. Zi-natne. 1986.
 11. Trahtengerc E.A. Metody generacii, ocenki i soglasovaniya reshenij v raspredelen-nyh sistemah podderzhki prinyatiya reshenij. - Avtomatika i telemekhanika. 1995, № 4.
 12. Melihov A. N., Bershtejn L. S., Korovin S. YA. Situacionnye sovetuyushchie sistemy s nechetkoj logikoj. M., Nauka, 1990. -272 p.
 13. Klykov YU.I. Situacionnoe upravlenie bol'shimi sistemami M., "Energiya", 1974.
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала: <http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.522

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВИРТУАЛЬНЫХ АССИСТЕНТОВ ДЛЯ РАЗРАБОТКИ ПРОГРАММ

¹Федулов Я.А., ¹Тычинская А.М., ¹Якушев В.А., ²Федулова А.С.

¹Филиал Национального исследовательского университета «МЭИ» в г. Смоленске, Смоленск, Россия (214013, г. Смоленск, Энергетический пр-д, 1), e-mail: fedulov_yar@mail.ru

²Национальный исследовательский университет «МЭИ», Москва, Россия (111250, г. Москва, ул. Красноказарменная, 14), e-mail: voitsitskay@mail.ru

В работе представлены распространенные виртуальные ассистенты, позволяющие упростить ввод запроса для поиска информации или запуска приложений. Отражена статистика использования виртуальных помощников с функциями необходимыми для разработки новых программ. Приведены положительные и отрицательные стороны использования виртуальных ассистентов, а также выделены основные требования к данным помощникам. Тестирование каждого из способов ввода запросов, позволяет сделать вывод о необходимости создания модели более обобщенного и интеллектуального виртуального ассистента, выполняющего распознавание звукового сигнала с использованием статистической модели для оптимальной работы приложения, что позволит существенно облегчить ввод запросов. Сделаны выводы о возможностях использования разработанных моделей для создания реальных виртуальных ассистентов, которые позволят облегчить использование поисковых систем и приложений.

Ключевые слова: виртуальный ассистент, интерфейс прикладного программирования, поиск информации, голосовое управление.

COMPARATIVE ANALYSIS OF VIRTUAL ASSISTANTS FOR PROGRAM DEVELOPMENT

¹Fedulov Ya.A., ¹Tychinskaya A.M., ¹Yakushev V.A., ²Fedulova A.S.

¹Branch of the National Research University "MEI" in Smolensk, Smolensk, Russia (214013, Smolensk, Energetichesky pr-d, 1), e-mail: fedulov_yar@mail.ru

²National Research University "MEI", Moscow, Russia (111250, Moscow, Krasnokazarmennaya st., 14), e-mail: voitsitskay@mail.ru

The paper presents common virtual assistants that make it easier to enter a query to search for information or launch applications. The statistics of the use of virtual assistants with the functions necessary for the development of new programs is reflected. The positive and negative aspects of using virtual assistants are given, and the basic requirements for these assistants are highlighted. Testing each of the methods for entering queries allows us to conclude that it is necessary to create a model of a more generalized and intelligent virtual assistant that performs audio signal recognition using a statistical model for optimal operation of the application, which will significantly facilitate the input of queries. Conclusions are made about the possibilities of using the developed models to create real virtual assistants, which will facilitate the use of search engines and applications.

Keywords: virtual assistant, application programming interface, information retrieval, voice control.

В современном мире большое количество информации хранится в сети Интернет, для получения доступа к которой существует ряд способов, таких как ввод запросов вручную с помощью устройств ввода, или использование виртуальных голосовых ассистентов [1]. Применение виртуальных помощников встречается в различных областях человеческой жизни, как игровой индустрии, так и в повседневной деятельности для установки или запуска приложений.

Актуальность рассматриваемой в данной работе темы обуславливается необходимостью учета технических и программных особенностей персональных компьютеров или смартфонов для установки и использования виртуальных ассистентов при решении различных прикладных и научных задач.

Целью исследования является изучение распространенных и малоизвестных виртуальных ассистентов, а также экспериментальное подтверждение преимуществ виртуальных помощников с открытым исходным кодом, которые подходят для решения широкого класса частных задач и предоставляют возможности внедрения улучшений.

В качестве программно-аппаратной платформы для проведения экспериментов с использованием реальных виртуальных ассистентов выступал ноутбук, на котором установлены параллельно две операционные системы Windows 8 и Ubuntu 2020. Различные операционные системы и установленные браузеры позволили провести необходимые эксперименты и сделать выводы о популярных и малораспространенных виртуальных ассистентах, а также осветить их особенности. В данной статье проводится анализ существующих виртуальных ассистентов, рассматриваются их основные функции, в том числе возможности определения частей для модификации и программного использования.

Виртуальные ассистенты – это программы, работающие на основе искусственных нейронных сетей, при этом запросы могут быть представлены в письменном виде или распознавания голоса и дальнейшей обработки полученного сообщения [2]. Использование искусственных нейронных сетей позволяет данным программам вести диалог, выполнять простые задачи. Чаще всего виртуальные ассистенты представлены голосовыми помощниками, разработанные с использованием искусственного интеллекта, технологий машинного обучения, а также распознавания голоса.

Идея распознавания голосовых сообщений с использованием информационных технологий зародилась еще в середине XIX века, ученые разрабатывали машину для распознавания одночленных звуков. В XXI веке вычислительные машины позволяют находить целые предложения, это стало возможным после проведения обучения и автоматизации, что показывает высокий уровень развития технологий, которые позволяют искусственным-нейронным сетям проводить анализ контекста и определять источник звукового сообщения.

Использование виртуальных помощников может быть различным, как представлено на рисунке 1.

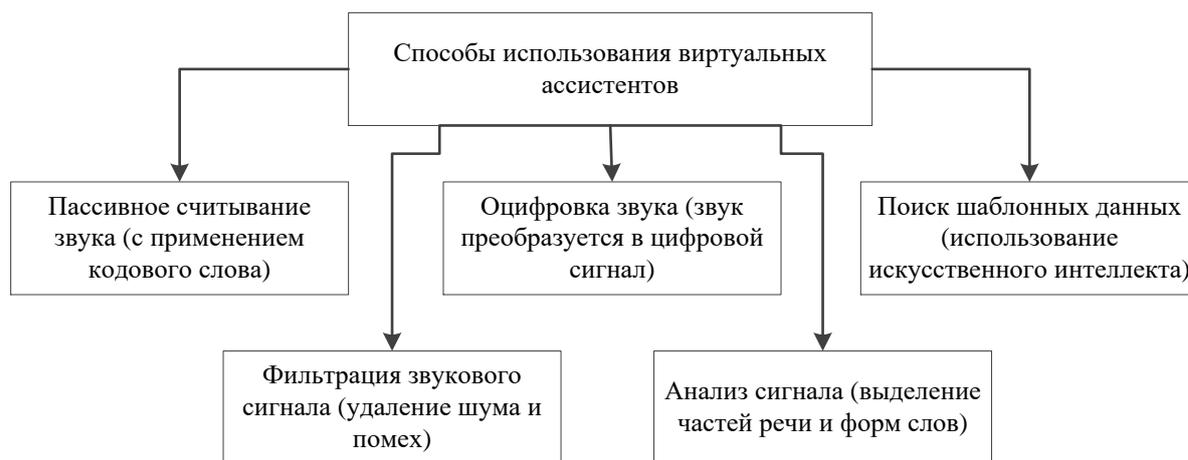


Рисунок 1 – Способы использования виртуальных ассистентов с пояснениями

Для реализации виртуального ассистента может быть выбран любой из существующих языков программирования, и на сегодняшний момент одним из самых распространенных является Python, поскольку он ориентирован на повышение производительности программиста и читаемости кода, а также его стандартная библиотека имеет множество полезных функций.

Для получения необходимых данных о возможности применения готовых решений и дальнейшей разработки программного обеспечения на их основе были проведены следующие эксперименты и сделаны соответствующие выводы. Была поставлена цель установки голосового ассистента на компьютер, для тестирования его свойств.

Описание экспериментов

Для работы были установлены последние версии браузеров, а также скачаны и установлены необходимые для экспериментов виртуальные помощники: Алиса (Yandex), Siri (Apple), Google-ассистент (для телефонов на Android), CMU Sphinx и Julius (Linux).

Голосовой помощник Алиса (Yandex)

Алиса – виртуальный русский голосовой ассистент, разработан программистами компании «Яндекс» в 2017 году. Алиса считается одной из самых популярных приложений среди виртуальных ассистентов, так как имеет большое количество функций: распознавание естественной речи, имитация живого диалога, может решать прикладные задачи.

Работа приложения представлена следующими пунктами:

- 1) речь пользователя транслируется на сервер распознавания;
- 2) сервер преобразует полученный звуковой сигнал в текстовый;
- 3) полученный текст отправляется в сервис в определенную сущность – классификатор интенгов;
- 4) каждому интенгу соответствует специальная модель (семантический теггер).

Основная задача модели – выделение необходимой информации для себя из сказанной пользователем фразы. Обработанная информация в модуль dialog manager, в котором хранится контекст диалога, а также что предшествовало данному запросу. В данный модуль поступают результаты разбора сказанной реплики, и система должна принять решение о дальнейших действиях с ним.

Выполнение запроса с использованием виртуального ассистента «Алиса» можно представить схематично, как показано на рисунке 2.



Рисунок 2 – Обработка звукового запроса виртуальным ассистентом

Данный виртуальный помощник функционирует на смартфонах и компьютерах, а также на информационно-развлекательной платформе для автомобилей (Яндекс-Авто), в «умной» колонке со встроенным голосовым помощником «Алиса».

Установка и работа с данным виртуальным ассистентом проста и понятна для пользователя смартфона и компьютера, что можно считать основным достоинством данного приложения, но привязанность к определенному браузеру делает его негибким и ограничивает возможности для дальнейшего использования.

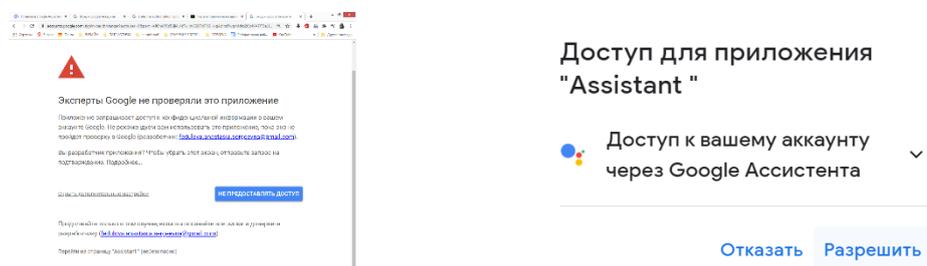
Интерфейс прикладного программирования (Application Programming Interface, API) в приложении Алиса позволяет разработать собственный навык и опубликовать его в специализированном сервисе «Яндекс.Диалоги» и он будет доступен для дальнейшей работы с ассистентом. Разработка навыка начинается с определения задачи, которая должна быть решена с помощью него, при этом задействуются определенные инструменты для интеграции навыка в Алису. Выделяются следующие типы навыков: навыки общего типа, навыки общего типа, навыки «Синтезатор».

Google-ассистент (на платформе Android)

Google Assistant – виртуальный голосовой ассистент, разработан программистами компании «Google» в 2016 году. Предпочтительнее работает на смартфонах с операционной системой Android, но для компьютеров разработан ассистент, устанавливаемый с помощью Google Cloud Platform - Tools & Modern Applications, но для этого необходимо иметь Google-аккаунт.

Алгоритм установки [3] данного приложения на компьютер достаточно трудоемкий. Далее представлен процесс установки Google Assistant на персональный компьютер (рисунок 3). Как видно из рисунка, необходимо давать полный доступ к компьютеру, чтобы приложение

начало функционировать на компьютере. Далее представлен результат работы с установленным приложением, рисунок 4.



```
Please visit this URL to authorize this application:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=371305482358-9cdeak2aodbmvehkao54bujk563ig6vl.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Aawg%3Aoauth%3A2.0%3Aaob&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fassistant-sdk-prototype+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fassistant-sdk-prototype&state=WhrizNlQDwJgell2pV6bIsPOLdMmT4&prompt=consent&access_type=offline
Enter the authorization code: 4/1AY0e-g7D-7g7W1Xu8CyViCbe6R-DrKWZALiC6gXUCLJqt8zZAURNvDydKME

googlesamples-assistant-devicetool --project-id primal-context-299110 register-model --manufacturer "Assistant SDK developer" --product-name "Assistant SDK light" --type LIGHT --model "371305482358-9cdeak2aodbmvehkao54bujk563ig6vl.apps.googleusercontent.com_OAuth"
py -m googlesamples.assistant.grpc.pushtotalk --device-model-id "371305482358-9cdeak2aodbmvehkao54bujk563ig6vl.apps.googleusercontent.com_OAuth" --project-id primal-context-299110
```

Рисунок 3 – Установка Google-Assistant на компьютере с Windows

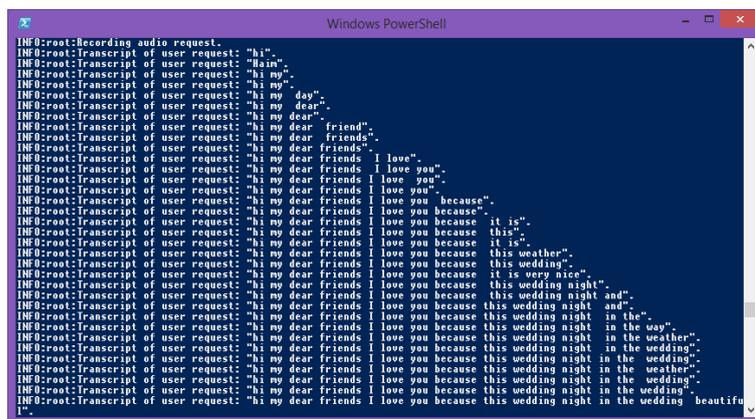


Рисунок 4 – Работа приложения на компьютере

Виртуальный помощник Google Assistant используется в смартфонах, включен в приложение для мгновенного обмена сообщениями, в умный голосовой Wi-Fi динамик для управления умным домом, в программное обеспечение умных часов от Google. Основная проблема в установке на персональный компьютер, виртуальный помощник на компьютере воспринимает только английские слова.

Код данного виртуального ассистента частично открыт для модификаций. Для выполнения и добавления запросов и интеграции с Ассистентом, разработчики предоставляют Google доступ к определенным данным, таким как:

- 1) информация необходимая, чтобы установить связь между аккаунтами (пользователи могут взаимодействовать со сторонними функциями через виртуального помощника);
- 2) библиотеки данных, а также коллекции контента (позволяют приложению выполнять запросы с использованием сервиса разработчика);

3) данные для взаимодействия с пользователями (информация передается в Google разработчиками, использующими Actions on Google).

4) пользовательские запросы и данные, которые связаны через запрос (информация отправляется в Google устройствами внешних производителей со встроенным виртуальным ассистентом (колонки, телевизоры) для того, чтобы помощник смог осуществлять дальнейшую работу).

Для разработки собственных приложений с использованием предоставляемых голосовым ассистентом возможностей, были рассмотрены возможности работы с использованием Google API Assistant. Служба Google Assistant предоставляет низкоуровневый API, который позволяет вам напрямую управлять байтами звука в запросе и ответе Assistant. Привязки для этого API могут быть созданы для таких языков, как Node.js, Go, C ++, Java, для всех платформ, поддерживающих gRPC. Python предоставляется справочный код для захвата звука, воспроизведения звука и управления состоянием разговора.

Голосовой помощник Siri (Apple)

Siri – виртуальный голосовой ассистент, разработан программистами Международного центра искусственного интеллекта (SRI) в 2007 году, названного самым большим проектом, в котором задействован искусственный интеллект [**Ошибка! Источник ссылки не найден.**], который в 2010 был выкуплен компанией Apple.

Особенности данного виртуального ассистента [5]:

1) использует обработку естественной речи, для ответа на вопросы, а также для построения рекомендации;

2) приспосабливается индивидуально под каждого пользователя, изучает предпочтения в течение долгого периода времени.

Распознавание речи возможно с использованием голосовых технологий компании Nuance Communications [6].

Действия и ответы формируются с использованием разработок партнёров:

- Citysearch, BooRah, Yelp Inc, Yahoo Local, Yandex, ReserveTravel – решение деловых вопросов;
- Eventful, StubHub и LiveKick – события и концертная информация;
- MovieTickets.com, Rotten Tomatoes, The New York Times информация о фильмах;
- Yahoo Weather для информации о погоде;
- Bing, Yahoo, Google для веб-поиска.

Минусы использования данного ассистента.

1. Использование только на продуктах компании Apple.

2. Запросы могут быть не обработаны.

3. Настройка может быть сделана на прием запросов, также при заблокированном телефоне (иные лица могут использовать данный телефон для отправки СМС-сообщений, также сервиса СМС-банкинга для получения баланса карты владельца телефона или кражи денег).

Для понимания работы приложения Siri необходимо рассмотреть API – SiriKit, которое работает с использованием набора доменов, которые предоставляют связанные области функциональности, такие как обмен сообщениями. В каждом домене есть набор намерений, которые представляют конкретные задачи, решаемые пользователем с помощью Siri. Обработка

языка в приложении сводится к тому, что SiriKit решает, какое намерение и приложение запрашивает пользователь, и код проверяет, что пользователь просит, а затем выполняет это.

Системы распознавания речи CMU Sphinx и Julius (Linux)

CMU Sphinx – группа систем распознавания речи, разработанных программистами в Университете Карнеги-Меллона. Данный набор содержит в себе различные распознаватели речи, такие как Sphinx 2–4, а также тренажер акустических моделей – SphinxTrain. В 2000 году группа Sphinx открыла исходный код некоторых компонентов распознавателя речи, речевые декодеры поставляются с примерами приложений [7]. Для обучения акустической модели необходимо дополнительно программное обеспечение, компилятор языковой модели, а также общедоступный словарь произношения – *cmudict*.

CMU Sphinx имеет следующие особенности: современные алгоритмы распознавания речи для эффективного распознавания речи, инструменты CMU Sphinx разработаны специально для платформ с низким уровнем ресурсов, гибкий дизайн, большое количество инструментов для функций, связанных с распознаванием речи.

Julius – высокопроизводительное двухпроходное программное обеспечение (ПО) для исследователей и разработчиков, связанных с речью, используется, чтобы проводить декодирование непрерывной речи, для этого имеет обширный словарь.

Выполняет декодирование в реальном времени на большинстве современных персональных компьютерах, с использованием триграммы слов и контекстно-зависимой скрытой марковской модели. В рассматриваемую систему включены основные методы поиска.

Основной платформой является Linux, а также иные рабочие станции Unix, отдельные версии ассистента поддерживают работу на операционной системе Windows . Julius – бесплатное ПО с открытым исходным кодом.

В Julius версии 3.4 встроен грамматический анализатор распознавания Julian . Julian – модификация Julius, где языковая модель – разработанный вручную тип конечного автомата, называемый грамматикой детерминированного конечного автомата, который может использоваться для создания системы голосовых команд со своим словарем или для различных задач голосовой диалоговой системы.

API имеет особенности: программное обеспечение с открытым исходным кодом, высокоскоростное и точное распознавание в реальном времени на основе стратегии двух проходов, низкие требования к памяти.

Для дальнейшей работы необходимо получить информацию об использовании голосовых ассистентов в России [8]: «Алиса» – осведомлены 96% респондентов и используется 58,79%. Google Assistant – 20,13% участников исследования используют, а осведомлены 67%. О Siri знают 61% опрошенных, из них 16,29% активно используют. CMU Sphinx и Julius (Linux) знают 15% участников, используют 4,79%, статистика представлена на рисунке 5.

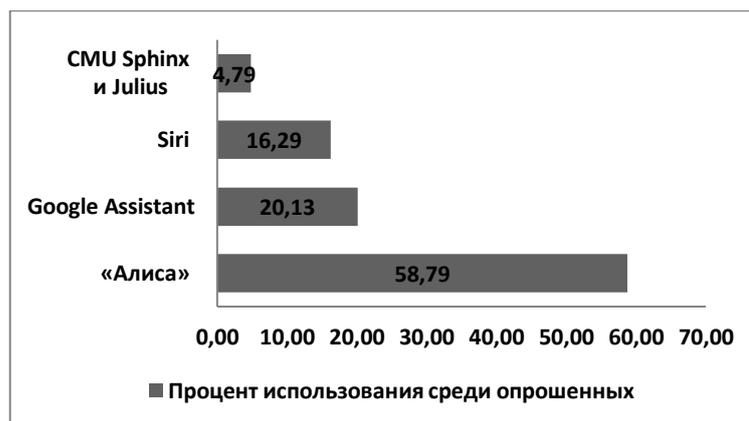


Рисунок 5 – Статистика использования виртуальных ассистентов

По результатам проведенного анализа с учетом потребностей как пользователей, так и разработчиков программного обеспечения были выделены основные характеристики виртуальных помощников, представленные в таблице 1.

Таблица 1 – Характеристики виртуальных помощников

№ п/п	Характеристики	Алиса	Google Assistant	Siri	CMU Sphinx и Julius
1	Операционная система	Microsoft Windows, Android и iOS	Android, Chrome OS, iOS, iPadOS, KaiOS, Linux, Wear OS, Android Auto	iOS 5-13, macOS Sierra, macOS High Sierra, macOS Mojave	Linux, Mac OS, Windows, Android
2	Приветствие или активация	Привет, Алиса	Ок, Google	Привет, Siri	Задается пользователем (ок, компьютер)
3	Понимание произвольной речи и команд	Да	Да	Да	Настраиваемое
4	Применение	Распознавание речи, запуск приложений, реальное общение	Поиск фраз, работа с умными устройствами	Управление телефонными коммуникациями	Задается пользователем
5	Открытость кода	Smart Home API	Python client library or generated bindings for languages like Go, Java, C#, Node.js, and Ruby	API SiriKit	Консольный, API
6	Распознавание	Wake-Up-Word ASR	Wake-Up-Word ASR	Wake-Up-Word ASR	Алгоритм Витерби, bushderby
7	Реализация настраиваемых функций	Нет	Нет	Нет	Да

Исходя из проведенного сравнительного анализа основных голосовых помощников и их API, применительно к задачам распознавания звукового сообщения и разработки программных средств на их основе были сделаны следующие выводы. В случае возникновения необходимости использования виртуального голосового помощника, который будет легко установлен на требуемой платформе, подойдут готовые популярные решения Алиса, Google

Assistant, Siri, которые предоставляют достаточный набор предустановленных функциональных возможностей.

В случае использования предоставляемого популярными решениями средств разработки API приходится сталкиваться с рядом существенных ограничений, связанных с жесткостью и привязанностью к определенным программно-аппаратным платформам. Возможностей популярных готовых решений голосовых ассистентов не хватает для решения нетипичных задач, что приводит к необходимости создания собственного виртуального ассистента, работающего на персональных компьютерах или мобильных платформах. Для решения такого типа задач голосового управления требуется работать с виртуальными ассистентами с открытым кодом, дописывать и тонко настраивать их работу, и в данном случае открытые проекты CMU Sphinx и Julius предоставляют гораздо более широкие возможности.

На основе проведенного исследования ведется построение собственного усовершенствованного голосового ассистента, позволяющего не только осуществлять запрос по голосовому сообщению, но также осуществлять анализ полученного текста, проводить его распознавание и выдавать ответ. Разработка проводится на основе группы систем распознавания речи, для системы Linux с использованием структуры системы с применением кодового слова, что позволит использовать приложения не только для вывода информации, а также для запуска определенных приложений.

Список литературы

1. Смыслова Л. В. Чат-бот как современное средство интернет-коммуникаций // Молодой ученый. 2018. №9. С. 36-39.
2. Поначугин А.В., Пичужкина Д.Ю., Смекалова Е.С. Голосовой помощник как технология обработки данных. // НАУКА БЕЗ ГРАНИЦ. М.: НГПУ им. К. Минина, 2020 С. 96-100.
3. Установка Google Ассистента в Windows. URL: <https://g-ek.com/ustanovka-google-assistenta-v-windows-10>. Дата обращения: 12 декабря 2020.
4. M. Pinola, Speech recognition through the decades: how we ended up with Siri, [Электронный ресурс] // PCWorld, 2011. URL:http://www.techhive.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html?page=0.
5. Jain M., Kota R., Kumar P., Patel S.N. Convey: Exploring the Use of a Context View for Chatbots // Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. 2018.
6. Janarthanam S. Hands-On Chatbots and Conversational UI Development: Build chatbots and voice user interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills. Packt Publishing, 2017.
7. Поляков Е.В., Мажанов М.С., Качалова М.В., Поляков С.В. Разработка интеллектуального голосового ассистента и исследование обучающей способности алгоритмов распознавания естественного языка // Системный администратор. 2017. №12.
8. Наумов М.Ю., Чистяков А.С. Применение систем искусственного интеллекта в различных сферах деятельности // Постулат. 2017. №5.

References

- 1 Smyslova LV Chatbot as a modern means of Internet communications // Young scientist. 2018. No. 9. pp. 36-39.
 - 2 Ponachugin A.V., Pichuzhkina D.Yu., Smekalova E.S. Voice assistant as a data processing technology. // SCIENCE WITHOUT BORDERS. М.: NGPU im. K. Minina, 2020 pp. 96-100.
 - 3 Installing Google Assistant in Windows. URL: <https://g-ek.com/ustanovka-google-assistent-a-v-windows-10>. Date of treatment: December 12, 2020.
 - 4 M. Pinola, Speech recognition in decades: how we came to Siri, [Electronic resource] // PCWorld, 2011. URL: http://www.techhive.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html? Page = 0.
 - 5 Jain M., Kota R., Kumar P., Patel S.N. Convey: Exploring the Use of Context View for Chatbots // Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. 2018.
 - 6 Janarthnam S. Hands-On Chatbots and Conversational UI Development: Build chatbots and voice user interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills. Packt Publishing, 2017.
 - 7 Polyakov E.V., Mazhanov M.S., Kachalova M.V., Polyakov S.V. Development of an intelligent voice assistant and study of the learning ability of natural language recognition algorithms // System Administrator. 2017. No. 12.
 - 8 Naumov M.Yu., Chistyakov A.S. Application of artificial intelligence systems in various fields of activity // Postulate. 2017. No. 5.
-