

*Журнал «Научное обозрение.
Технические науки»
зарегистрирован Федеральной службой
по надзору в сфере связи, информационных
технологий и массовых коммуникаций.
Свидетельство ПИ № ФС77-57440
ISSN 2500-0799*

**Двухлетний импакт-фактор РИНЦ – 0,792
Пятилетний импакт-фактор РИНЦ – 0,350**

*Учредитель, издательство и редакция:
ООО НИЦ «Академия Естествознания»,
Почтовый адрес: 105037, г. Москва, а/я 47
Адрес редакции и издателя: 410056, Саратовская
область, г. Саратов, ул. им. Чапаева В.И., д. 56*

**Founder, publisher and edition:
LLC SPC Academy of Natural History,
Post address: 105037, Moscow, p.o. box 47
Editorial and publisher address: 410056,
Saratov region, Saratov, V.I. Chapaev Street, 56**

*Подписано в печать 31.10.2022
Дата выхода номера 30.11.2022
Формат 60×90 1/8*

*Типография
ООО НИЦ «Академия Естествознания»,
410035, Саратовская область,
г. Саратов, ул. Мамонтовой, д. 5*

**Signed in print 31.10.2022
Release date 30.11.2022
Format 60×90 8.1**

**Typography
LLC SPC «Academy Of Natural History»
410035, Russia, Saratov region,
Saratov, 5 Mamontovoi str.**

*Технический редактор Доронкина Е.Н.
Корректор Галенкина Е.С., Дудкина Н.А.*

*Тираж 1000 экз.
Распространение по свободной цене
Заказ НО 2022/5
© ООО НИЦ «Академия Естествознания»*

Журнал «НАУЧНОЕ ОБОЗРЕНИЕ» выходил с 1894 по 1903 год в издательстве П.П. Сойкина. Главным редактором журнала был Михаил Михайлович Филиппов. В журнале публиковались работы Ленина, Плеханова, Циолковского, Менделеева, Бехтерева, Лесгафта и др.

Journal «Scientific Review» published from 1894 to 1903. P.P. Soykin was the publisher. Mikhail Filippov was the Editor in Chief. The journal published works of Lenin, Plekhanov, Tsiolkovsky, Mendeleev, Bekhterev, Lesgaft etc.



М.М. Филиппов (M.M. Philippov)

**С 2014 года издание журнала возобновлено
Академией Естествознания
From 2014 edition of the journal resumed
by Academy of Natural History**

**Главный редактор: М.Ю. Ледванов
Editor in Chief: M.Yu. Ledvanov**

**Редакционная коллегия (Editorial Board)
А.Н. Курзанов (A.N. Kurzanov)
Н.Ю. Стукова (N.Yu. Stukova)
М.Н. Бизенкова (M.N. Bizenkova)
Н.Е. Старчикова (N.E. Starchikova)
Т.В. Шнуровозова (T.V. Shnurovozova)**

НАУЧНОЕ ОБОЗРЕНИЕ • ТЕХНИЧЕСКИЕ НАУКИ

SCIENTIFIC REVIEW • TECHNICAL SCIENCES

www.science-education.ru

2022 г.



***В журнале представлены научные обзоры,
статьи проблемного
и научно-практического характера***

***The issue contains scientific reviews,
problem and practical scientific articles***

СОДЕРЖАНИЕ

Технические науки (2.2. Электроника, фотоника, приборостроение и связь, 2.3. Информационные технологии и телекоммуникации, 2.4. Энергетика и электротехника)

СТАТЬИ

АВТОМАТИЗАЦИЯ РАСЧЕТА КРУГОВОГО ТЕРМОДИНАМИЧЕСКОГО ЦИКЛА <i>Ильичев В.Ю., Федин Р.Р.</i>	5
ОСНОВНЫЕ ПОКАЗАТЕЛИ ДОСТУПНОСТИ В KUBERNETES И СПОСОБЫ ЕЕ ОБЕСПЕЧЕНИЯ <i>Липатова С.Е., Белов Ю.С.</i>	10
ВЫБОР МАТЕРИАЛОВ РОТОРА И КОНСТРУКЦИОННЫХ ЧАСТЕЙ ИНТЕГРИРОВАННОГО СТАРТЕРА-ГЕНЕРАТОРА <i>Уразбахтин Р.Р.</i>	15
ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ИНТЕРФЕЙСА НА ПРИМЕРЕ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ-АССИСТЕНТА ДЛЯ НАСТОЛЬНОЙ ИГРЫ «ДРЕВНИЙ УЖАС» С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ <i>Маслова Ю.А., Белов Ю.С., Федоров В.О.</i>	20
ПОДХОД К АДАПТАЦИИ СООБЩЕНИЙ ДЛЯ ОБМЕНА МЕЖДУ АВИАЦИОННЫМИ ПРИЛОЖЕНИЯМИ <i>Солоделов Ю.А., Камалетдинова Г.Р., Кожанов К.Д., Альбицкий Д.В., Трубочанинов И.А.</i>	25

CONTENTS

Technical sciences (2.2. Electronics, photonics, instrumentation and communication, 2.3. Information technologies and telecommunications, 2.4. Energy and electrical engineering)

ARTICLES

AUTOMATION OF CIRCULAR THERMODYNAMIC CYCLE CALCULATION

Ilichev V.Yu., Fedin R.R. 5

THE MAIN INDICATORS OF AVAILABILITY IN KUBERNETES AND WAYS TO ENSURE IT

Lipatova S.E., Belov Yu.S. 10

MATERIALS SELECTION FOR THE INTEGRATED STARTER-GENERATOR ROTOR AND CONSTRUCTIVE PARTS

Urazbakhtin R.R. 15

GENERAL PRINCIPLES OF BUILDING THE INTERFACE ON THE EXAMPLE OF DEVELOPING A MOBILE ASSISTANT APPLICATION FOR THE BOARD GAME “ANCIENT HORROR” USING AUGMENTED REALITY TECHNOLOGY

Maslova Yu.A., Belov Yu.S., Fedorov V.O. 20

APPROACH TO UNIFIED MESSAGING IN AVIATION APPLICATIONS

Solodelov Yu.A., Kamaletdinova G.R., Kozhanov K.D., Albitskiy D.V., Trubchaninov I.A. 25

СТАТЬИ

УДК 621.1.016.7

**АВТОМАТИЗАЦИЯ РАСЧЕТА
КРУГОВОГО ТЕРМОДИНАМИЧЕСКОГО ЦИКЛА**

Ильичев В.Ю., Федин Р.Р.

*ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана»,
Калужский филиал, Калуга, e-mail: patrol8@yandex.ru*

Статья посвящена описанию исследований, направленных на создание средств автоматизированного определения параметров кругового термодинамического цикла для идеального газа. Вначале приводятся краткие сведения об особенностях круговых (замкнутых) циклов, используемых в тепловых двигателях. Далее сформулированы цель и задачи, решаемые в ходе исследования. Определены требования к программному продукту, с помощью которого должна осуществляться автоматизация расчетов. Для создания программы выбран свободно распространяемый язык Python, позволяющий производить вариативные расчеты циклов при разных исходных теплофизических параметрах и для различных газообразных рабочих тел. Программа написана на основе методики вычислений, приведенной в литературе, однако вычисление теплофизических величин осуществлено с помощью применения специальной библиотеки функций PropSI.CoolProp. Также использованы подключаемые модули: для создания пользовательского интерфейса программы Tkinter, заполнения массивов расчетными данными Numpy, представления результатов в числовом и графическом виде Threading и Pyplot.Matplotlib. Программный код отличается наглядностью и простотой изучения. Пользовательский интерфейс позволяет легко изменять исходные данные, быстро получать и верифицировать результаты, а также делать выводы о совершенстве термодинамических циклов на основе вычисления их термического КПД. Это приводит к существенному упрощению и ускорению расчетов циклов широко применяемых газотурбинных энергетических установок и тем самым позволяет повысить качество их проектирования. В заключение сделаны выводы по результатам представленной работы и даны рекомендации для ее дальнейшего использования.

Ключевые слова: круговой цикл, газотурбинная установка, теплофизические параметры, язык Python, библиотека CoolProp

AUTOMATION OF CIRCULAR THERMODYNAMIC CYCLE CALCULATION

Ilichev V.Yu., Fedin R.R.

Bauman Moscow State Technical University, Kaluga branch, Kaluga, e-mail: patrol8@yandex.ru

The article is devoted to the description of studies aimed at creating means of automated determination of parameters of a circular thermodynamic cycle for an ideal gas. First, brief information is given about the features of circular (closed) cycles used in thermal engines. Next, the purpose and tasks solved during the study are formulated. The requirements for the software product with which the automation of calculations should be carried out are determined. To create the program, a free Python language was chosen, which allows you to perform variable calculations of cycles at different initial thermophysical parameters and for various gaseous working bodies. The program is written on the basis of the calculation method given in the literature, however, the calculation of thermophysical values was carried out using a special library of functions PropSI.CoolProp. Plug-ins were also used: to create the user interface of the Tkinter program, fill the arrays with Numpy calculation data, present the results in numerical and graphical form Threading and Pyplot.Matplotlib. The program code is distinguished by clarity and ease of study. The user interface allows you to easily change the source data, quickly obtain and verify results, as well as draw conclusions about the perfection of thermodynamic cycles based on the calculation of their thermal efficiency. This leads to a significant simplification and acceleration of the calculation of cycles of widely used gas turbine power plants and thereby makes it possible to improve the quality of their design. Conclusions were made based on results of presented work and recommendations were given for its further use.

Keywords: circular cycle, gas turbine plant, thermophysical parameters, Python language, CoolProp library

Одним из основных предметов термодинамики является изучение протекания процессов по так называемым круговым или замкнутым циклам [1], в результате которых параметры рабочего тела постоянно возвращаются к исходным значениям. Для идеальных газов такие циклы состоят из следующих отдельных равновесных процессов: изобарного (при протекании которого давление газа $p = const$), изотермического (температура $T = const$), изохорного (удельный объем $v = const$), адиабатного (без подвода и отвода теплоты), изоэнтроп-

ного (подвод или отвод теплоты при неизменной энтропии S). Все эти процессы рассматриваются в предположении, что подвод или отвод теплоты (если он есть) является неизменным с течением времени (все процессы являются политропными) [2].

Круговой цикл является циклом теплового двигателя [3] при выполнении условий:
– последовательность протекания процессов в цикле, изображенном в $p-v$, $T-S$ (температура-энтропия) или $I-S$ (энтальпия-энтропия) диаграмме, осуществляется по часовой стрелке;

- присутствует подвод теплоты к рабочему телу от внешнего источника;

- совершается положительная работа.

Один из самых распространенных круговых циклов теплового двигателя, в котором рабочим телом является идеальный газ (теоретически любой), так и называется – цикл газотурбинного двигателя (ГТУ). Рассмотрим цикл ГТУ, изображенный в p - v координатах (рис. 1).

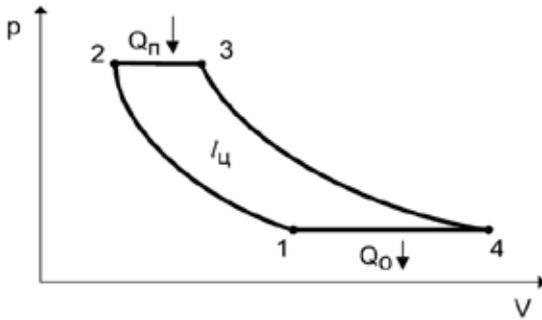


Рис. 1. p - v диаграмма кругового цикла газотурбинного теплового двигателя

Данный цикл состоит из следующих термодинамических процессов:

1–2 – политропное сжатие газа в компрессоре, при котором затрачивается работа;

2–3 – изобарный подвод теплоты Q_n в камере сгорания ГТУ;

3–4 – адиабатное расширение газа в турбине, совершающей полезную работу (идущую, например, на привод электрогенератора);

4–1 – изобарный отвод теплоты от газа Q_o в теплообменном аппарате.

Целью данной работы являлось создание расчетной программы для последовательного определения термодинамических параметров заданного газа в ключевых точках кругового цикла, которые необходимо вывести на экран компьютера. По результатам расчета необходимо определить:

- работу расширения (сжатия) газа во всех четырех политропных процессах;

- работу цикла l_w , равную площади внутри цикла в p - v координатах (рис. 1), которую также можно найти как сумму работ всех политропных процессов;

- теплоту для каждого политропного процесса (кроме адиабатного процесса 3–4);

- разность подведенной и отведенной теплоты, которую необходимо сравнить с работой цикла (должны совпадать):

$$l_w = Q_n - Q_o;$$

- термический КПД цикла;

$$\eta_t = l_w / Q_n.$$

После этого необходимо осуществить автоматизированное графическое построение цикла в p - v и I - S координатах.

Основными требованиями, предъявляемыми к расчетной программе, являются следующие:

- выбор методики расчета кругового цикла ГТУ для идеального газа;

- выбор свободно распространяемого языка программирования для наиболее простого осуществления расчетов и возможной последующей модернизации создаваемого программного продукта, а также для возможности изучения кода программы всеми желающими;

- создание удобного интерфейса пользователя для ввода исходных данных (с возможностью выбора разных газообразных рабочих тел) и проведения вычислений;

- визуализация результатов в наглядной численной и графической форме.

Материалы и методы исследования

В качестве основной методики для создания компьютерной программы использован алгоритм расчета, приведенный в книге [4]. Там же приведен пример, позволяющий проверить результат каждого расчетного действия.

Автор имеет большой опыт программирования на чрезвычайно распространенном языке Python [5], для которого к тому же существуют библиотеки функций (программные модули), позволяющие определять теплофизические свойства любых простых (однокомпонентных) газов, создавать и заполнять массивы данных результатами расчетов и осуществлять их вывод как в численной, так и в наглядной графической форме. Также язык Python славится простотой написания программного кода и широкими возможностями по созданию интерфейсов пользователя практически любой степени сложности. То есть этот язык программирования идеально подходит по всем сформулированным выше требованиям к расчетной программе.

Перечислим подключаемые библиотеки функций, использованные в программе:

1) модуль Tkinter [6], на основе которого создавался интерфейс пользователя, состоящий из программного окна, в котором размещаются поля ввода исходных данных, поле выбора газа (чекбоксы), текстовые данные (надписи), а также кнопки, при нажатии на которые осуществляется расчет с последующим выводом результатов в отдельные окна;

2) подмодуль PropsSI модуля CoolProp [7], позволяющий вычислять теплофизические параметры в системе СИ (температу-

ры, давления, удельные объемы, различные виды теплоемкости, энтальпии, энтропии и т.д.) для выбранного газа;

3) модуль NumPy [8] для создания массивов и заполнения их данными, полученными в результате расчетов, с целью дальнейшего их использования для построения графических зависимостей;

4) подмодуль Pyplot модуля Matplotlib, предназначенный для визуализации данных в виде графиков, оформленных согласно требованиям к представлению научной информации;

5) модуль Threading [9], позволяющий запускать одновременно несколько программных процессов, что необходимо для открытия дополнительного окна с выводом результатов расчета в численном виде.

После подключения в коде программы перечисленных библиотек функций, было осуществлено проектирование наиболее удобного интерфейса пользователя. Созданный интерфейс программы с введенными исходными данными для расчета примера, приведенного в книге [4], показан на рис. 2.

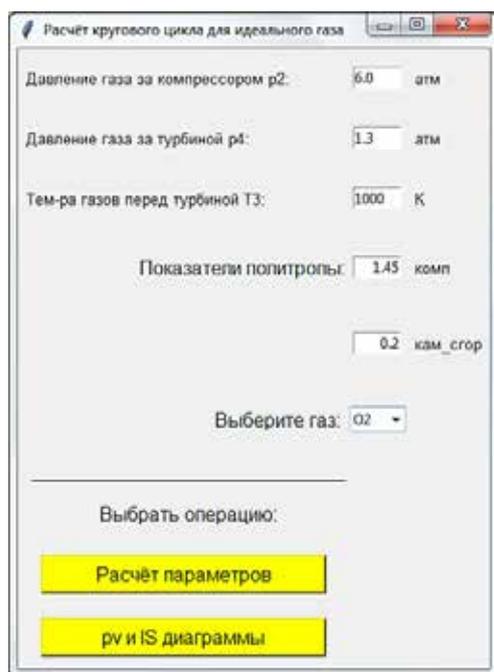


Рис. 2. Интерфейс программы расчета кругового цикла газотурбинного теплового двигателя

Интерфейс позволяет осуществлять ввод следующих данных: давление газа за компрессором p_2 (давление и температура газа перед компрессором в программе прописаны равными параметрам атмосферы – $p_1 = 1$ атм и $T_1 = 273,15$ К), температура газа перед турбиной T_3 , показатели

политропны для процессов в компрессоре и в камере сгорания. Также из всплывающего списка можно выбрать газ, используемый в качестве рабочего тела. Значения в полях ввода являются редактируемыми; таким образом можно осуществлять расчеты с любыми исходными параметрами.

При нажатии на программную кнопку «Расчет параметров» осуществляется присвоение соответствующим переменным значений, указанных в полях ввода интерфейса, и производится расчет термодинамических параметров цикла, перечисленных в задачах исследования. Затем результаты расчета выводятся в отдельном окне, благодаря использованию функций модуля Threading.

Особенностью реализации методики расчета в среде программирования Python является широкое использование модуля CoolProp для определения практически всех теплофизических параметров цикла (расчет любого необходимого параметра осуществляется по заданным двум известным параметрам и по приведенному в каждой команде PropsSI.CoolProp рабочему телу – в примере это кислород).

При нажатии на вторую программную кнопку « $p-v$ и $I-S$ диаграммы» также производится расчет параметров в узловых точках цикла 1–4 (рис. 1) и с помощью команд подмодуля Pyplot.Matplotlib на экран компьютера и в файлы выводятся последовательно графики рассчитанного цикла в $p-v$ и $I-S$ координатах.

Результаты исследования и их обсуждение

Показанные в интерфейсе программы на рис. 2 (заданные по умолчанию при запуске программы) исходные данные использовались для расчета примера по указанной выше методике. Рассмотрим результаты расчета, получаемые в численном и графическом виде после нажатия кнопок в интерфейсе.

На рис. 3 показаны результаты расчета, появляющиеся в специальном окне после нажатия на кнопку «Расчет параметров».

В окно выводятся только основные результаты – удельные объемы, температуры, давления в ключевых точках цикла, работа, произведенная в термодинамических процессах, и работа цикла в целом, а также соответствующее подведенное или отведенное количество теплоты. Из результатов видно, что работа и теплота цикла практически совпадают (разница составляет лишь 0,86%), что указывает на правильность выполненных расчетов. Также в окне показан вычисленный термический КПД цикла, составляющий 32%.

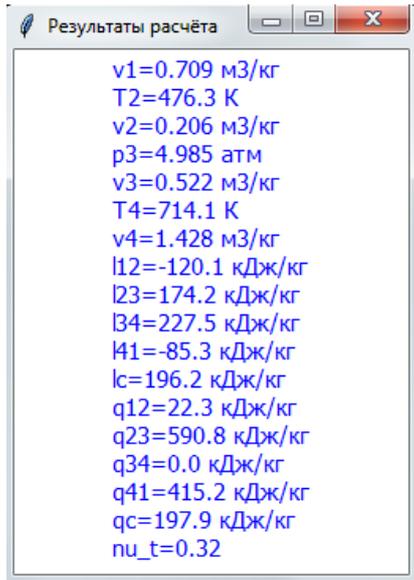


Рис. 3. Основные расчетные параметры цикла

На рис. 4 и 5 выведены графические изображения цикла, полученные при нажатии на кнопку « $p-v$ и $I-S$ диаграммы».

Правильность положения точек цикла на его $p-v$ диаграмме можно проверить по приведенным на рис. 3 численным значениям вычисленных параметров. Сравнение подтверждает точность выведения данных на график.

По $I-S$ диаграмме видно, что «подведение» работы в компрессоре (процесс 1–2) или теплоты в камере сгорания (процесс 2–3) приводит к увеличению энтальпии и энтропии (что объясняется активизацией теплового движения молекул газа) [10]. В турбине (процесс 3–4) нет подведения и отведения теплоты, так как процесс адиабатный, однако совершается работа, что приводит к уменьшению энтальпии. И наконец, в теплообменном аппарате (процесс 4–1) теплота отводится, поэтому энтальпия и энтропия уменьшаются.

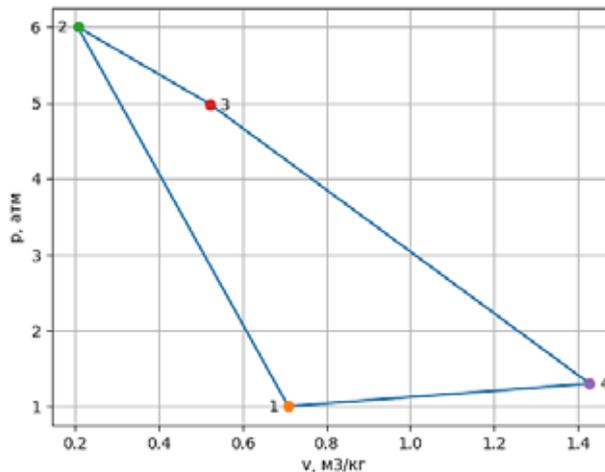


Рис. 4. $p-v$ диаграмма цикла, полученная по результатам его расчета

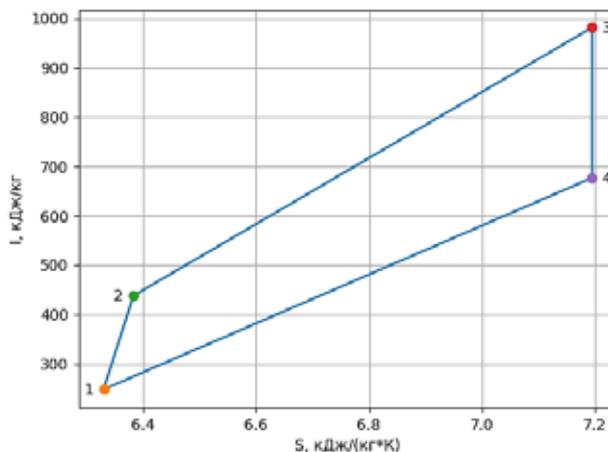


Рис. 5. $I-S$ диаграмма цикла, полученная по результатам его расчета

При других исходных данных можно получить результаты, оформленные подобным же образом, что позволяет оценить влияние входных параметров расчета на результаты и оптимизировать круговой цикл газотурбинной установки за счет максимизации его термического КПД.

Выполненная работа показала удобство и простоту использования языка Python для создания программы расчета кругового термодинамического цикла для идеального газа. Результаты расчета примера совпали с результатами расчета «вручную», что подтверждает правильность написания кода программы.

Python является интерпретируемым языком программирования, т.е. для использования кода нужно установить в систему Linux или Windows как дистрибутив пакета, так и все используемые библиотеки функций. Однако, благодаря использованию программы Pyinstaller, можно скомпилировать код программы вместе с дополнительными модулями в один файл и таким образом получить полноценную программу, не требующую установки, что и было произведено в результате выполнения работы. При этом сохраняется как возможность изучения кода программы (его можно открыть любым редактором текстовых файлов), так и ее запуска на любом компьютере без необходимости установки самого дистрибутива Python.

Очень интересным является изучение влияния различных рабочих газообразных тел на вид получаемого кругового цикла, что может пригодиться при обучении основам теплофизики.

Заключение

Таким образом, все сформулированные задачи работы, а также ее цель полностью достигнуты. Помимо этого, обоснован выбор языка Python для создания кода программы и рассмотрены использованные при этом программные средства.

Написанная программа позволяет производить исследование влияния исходных факторов на показатели термической эффективности цикла и таким образом подобрать его оптимальные параметры.

Автоматизация расчетов приводит к снижению времени расчетов и соответствующему уменьшению затрат на начальном этапе проектирования газотурбинных установок [11]. Созданная программа может использоваться учащимися школ и студентами вузов при изучении курсов «Термодинамика» и «Теплофизика», а также инженерами-проектировщиками газотурбинных установок. Изучение кода программы поможет освоить применение различных современных методов и средств программирования на снискавшем большую популярность языке Python.

Список литературы

1. Афанасьева Е.В., Корнеев С.Д. Теоретические основы теплотехники. Круговые процессы. М.: Московский государственный индустриальный университет, 2014. 38 с.
2. Овезгелдиевакурс О.Б. Адиабатный процесс. Политропный процесс // Научные механизмы решения проблем инновационного развития: сборник статей Международной научно-практической конференции: в 3 частях. 2017. С. 114-117.
3. Янченко В.С. Моделирование обобщенного термодинамического цикла тепловых двигателей // Транспорт на альтернативном топливе. 2011. № 4 (22). С. 41-44.
4. Карышев А.К., Лапин Ю.Д., Симонов В.П. Теплофизика: учебное пособие / Под ред. А.К. Карышева. М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. 107 с.
5. Ильичев В.Ю. Создание параметрических конечно-элементных трехмерных объектов с использованием функций Python // Системный администратор. 2021. № 5 (222). С. 82-85.
6. Радионов С.В. Создание интерфейса для программы Python3 с помощью библиотеки Tkinter // Постулат. 2019. № 1-1 (39). С. 41.
7. Ильичев В.Ю. Разработка программных продуктов с использованием модуля Python CoolProp для исследования эффективности утилизации тепла продуктов сгорания газообразных топлив // Системный администратор. 2020. № 11 (216). С. 80-83.
8. Ильичев В.Ю., Качурин А.В. Создание программ на языке Python для исследования множества Мандельброта // E-Scio. 2021. № 5 (56). С. 362-371.
9. Арьков В.Ю. Параллельные потоки и Hyper-Threading: возможности ускорения // Системный администратор. 2020. № 6 (211). С. 72-77.
10. Данеев А.В., Ванчиков В.Ц., Мишарина О.С. Об истории вероятностного подхода к исследованию теплового движения молекул // Информационные технологии и проблемы математического моделирования сложных систем. 2015. № 14. С. 33-38.
11. Трошенькин Б.А., Трошенькин В.Б. Совершенствование термодинамических циклов газотурбинных установок // Современная наука: исследования, идеи, результаты, технологии. 2014. № 2 (15). С. 45-48.

УДК 004.457

ОСНОВНЫЕ ПОКАЗАТЕЛИ ДОСТУПНОСТИ В KUBERNETES И СПОСОБЫ ЕЕ ОБЕСПЕЧЕНИЯ

Липатова С.Е., Белов Ю.С.

*Московский государственный технический университет имени Н.Э. Баумана,
Калужский филиал, Калуга, e-mail: fn1-kf@mail.ru*

Kubernetes – один из популярнейших инструментов оркестровки контейнеров, обеспечивающий возможность гибкого управления системой. Постепенно внедряясь в компании разработки по всему миру, одним из ключевых аспектов данной технологии остается доступность. Kubernetes позволяет обеспечить более высокую степень доступности приложений по сравнению с аналогами в различных сценариях сбоев. В данном исследовании будут рассмотрены как особенности работы Kubernetes и его внутреннее устройство для полного понимания процессов, так и метрики доступности, позволяющие проанализировать состояние системы и её поведение в различных ситуациях. Помимо этого, будут рассмотрены уровни проверки работоспособности Kubernetes, исходя из которых можно выделить основные сценарии сбоев и способы их моделирования для проведения анализа эффективности системы, оптимальность настроек для обеспечения доступности, а также самостоятельного исследования степени защищенности и стабильности системы, её поведения во внештатных ситуациях. Всё это играет важную роль как для небольших проектов, так и для крупных и сложных систем, уровень доступности является одним из ключевых параметров стабильности сервисов, что напрямую влияет на техническую среду, экономические показатели, а также уровень доверия и одобрения пользователей. Чем выше скорость восстановления системы после сбоев и отказоустойчивость сервисов, тем выше показатели работы приложения в разрезе различных сфер и компонентов.

Ключевые слова: Kubernetes, доступность, метрики доступности, сценарии сбоев

THE MAIN INDICATORS OF AVAILABILITY IN KUBERNETES AND WAYS TO ENSURE IT

Lipatova S.E., Belov Yu.S.

*Bauman Moscow State Technical University, Kaluga branch, Kaluga,
e-mail: sonya_lipatova@list.ru*

Kubernetes is one of the most popular container orchestration tools, providing flexible system management. Gradually being introduced into development companies around the world, one of the key aspects of this technology remains accessibility. Kubernetes allows you to provide a higher degree of application availability compared to analogues in various failure scenarios. This study will examine both the features of Kubernetes and its internal structure for a complete understanding of processes, and availability metrics that allow analyzing the state of the system and its behavior in various situations. In addition, the levels of Kubernetes health check will be considered, based on which it is possible to identify the main failure scenarios and ways to model them for analyzing the effectiveness of the system, the optimality of settings to ensure availability, as well as an independent study of the degree of security and stability of the system, its behavior in emergency situations. All this plays an important role both for small projects and for large and complex systems, the level of availability is one of the key parameters of service stability, which directly affects the technical environment, economic indicators, as well as the level of trust and approval of users. The higher the speed of system recovery after failures and the fault tolerance of services, the higher the performance of the application in the context of various spheres and components.

Keywords: Kubernetes, availability, availability metrics, failure scenarios

В связи с ростом популярности технологии Kubernetes, а также микросервисной архитектуры, имеющей множество преимуществ в сравнении с монолитной, важно уделить внимание одному из ключевых факторов использования этого стека: доступности микросервисов внутри кластеров, поскольку от этого напрямую зависит как стабильность сервисов, реакция пользователей, имеющая ведущую роль в работе приложений и систем, а также прибыль компаний-разработчиков. Однако перед тем, как рассмотреть критерии доступности и её составляющие, необходимо рассмотреть саму технологию Kubernetes и внутреннее устройство кластера для понимания особенностей работы данной тех-

нологии, а также определить, что именно относится к понятию «доступность».

Цель исследования – выделить основные показатели доступности в Kubernetes на уровне метрик, уровней проверки работоспособности и сценариев сбоев, а также выявить способы её обеспечения.

Понятие Kubernetes. Kubernetes (K8s) – это портативная расширяемая платформа с открытым исходным кодом для управления контейнеризованными рабочими нагрузками и сервисами, которая облегчает как декларативную настройку, так и автоматизацию [1].

В настоящий момент всё больше компаний по всему миру переходят к использованию данной технологии ввиду того, что это

позволяет сократить повторяющиеся ручные процессы, связанные с развертыванием контейнеров и управлением ими. С ростом популярности технологии растёт и её экосистема, что позволяет быстро найти ответы на интересующие вопросы по настройке конфигурации системы и т.д.

Являясь одним из самых популярных инструментов оркестровки контейнеров с открытым исходным кодом, Kubernetes стремительно проникает в стек технологий крупнейших компаний мира. Однако это не просто система оркестровки. Фактически он устраняет необходимость в оркестровке, поскольку включает в себя набор независимых, составляемых процессов управления, которые непрерывно приводят текущее состояние к заданному желаемому состоянию [2].

По своей сути Kubernetes представляет собой один из стратегических компонентов всего DevOps-процесса [3], именно поэтому большое внимание уделяется как кибербезопасности, так и доступности, поскольку именно эти два параметра имеют решающее значение как для функционирования сервисов в целом, так и для экономических и бизнес-показателей.

Иными словами, Kubernetes – это система гибкого управления инфраструктурой контейнеризации с возможностью балансировки нагрузки, обеспечивающая скорость, гибкость технологии и экономическую эффективность [4].

Устройство кластера Kubernetes. Рассмотрим устройство кластера Kubernetes

для полного понимания внутренних процессов и компонентов.

Основные компоненты кластера Kubernetes представлены на рис. 1.

Компоненты панели управления отвечают за основные операции кластера (например, планирование), а также обрабатывают события кластера (например, запускают новый под, когда поле replicas развертывания не соответствует требуемому количеству реплик) [1].

Рассмотрим подробнее компоненты управления K8s:

1) kube-apiserver – сервер API – компонент Kubernetes панели управления, который представляет API Kubernetes. API-сервер – это клиентская часть панели управления Kubernetes. Он предназначен для горизонтального масштабирования. Есть возможность запуска нескольких экземпляров kube-apiserver и балансировки трафика между ними;

2) etcd – распределённое и высоконадежное хранилище данных в формате «ключ-значение», которое используется как основное хранилище всех данных кластера в Kubernetes;

3) kube-scheduler – компонент плоскости управления, который отслеживает созданные поды без привязанного узла и выбирает узел, на котором они должны работать;

4) kube-controller-manager – компонент Control Plane запускает процессы контроллера;

5) cloud-controller-manager – запускает контроллеры, которые взаимодействуют с основными облачными провайдерами [1].

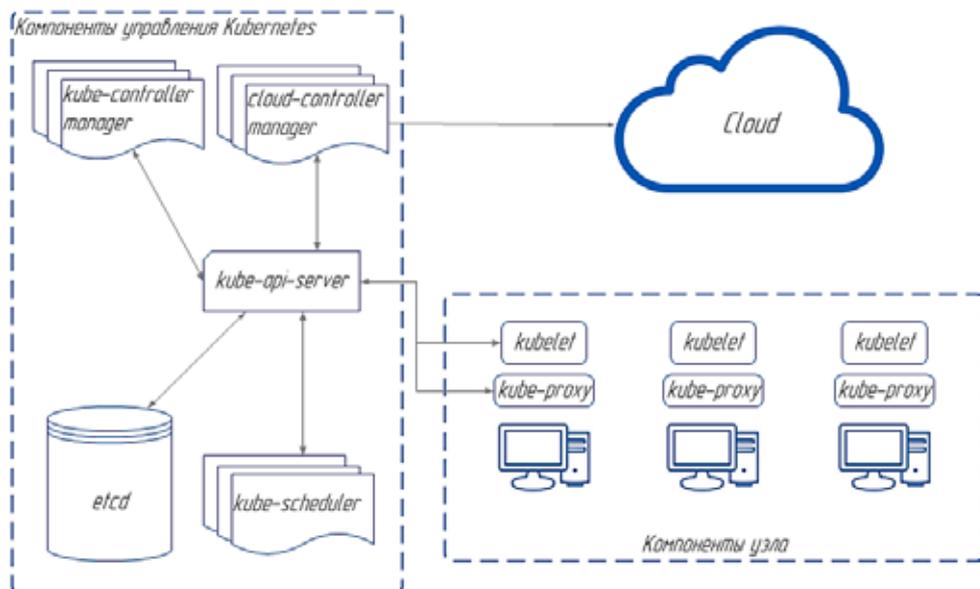


Рис. 1. Компоненты кластера Kubernetes

Рассмотрим компоненты узла, поддерживающие работу подов и среды выполнения Kubernetes:

1) kubelet – утилита на каждом кластере, следящая за тем, чтобы контейнеры были запущены в поде;

2) kube-proxy – сетевой прокси, работающий на каждом узле в кластере и реализующий часть концепции сервиса. Он конфигурирует правила сети на узлах;

3) среда выполнения контейнеров [1].

К дополнениям, расширяющим возможности K8s, относят DNS, Dashboard (веб-интерфейс), мониторинг ресурсов и логирование кластера.

Доступность. Доступность услуги – это нефункциональное требование, определяемое как процент времени предоставления услуги [5]. Высокая доступность достигается, когда система доступна не менее 99,999% времени, т.е. общее допустимое время простоя высокодоступных систем – около 5 минут в год [2].

Важной особенностью Kubernetes является перезапуск, замена или переназначение неисправных контейнеров в случае сбоя их хостов. Объявление контейнеров снова работоспособными происходит лишь после полного цикла их подготовки. Но эти меры могут оказаться недостаточными для поставщиков услуг операторского класса, и доступность в качестве важного нефункционального требования по-прежнему вызывает у них беспокойство [6]. Для этого требуется дополнительный анализ каждой системы, поскольку в зависимости от приложений, их назначений и сервисов настройки K8s будут различаться, а, следовательно, показатели доступности могут различаться.

Метрики доступности. Рассмотрим метрики, которые используются при анализе

доступности Kubernetes и их зависимость между собой. Схематично они представлены на рис. 2.

Рассмотрим подробнее каждую из метрик:

1) время реакции Kubernetes на сбой – период между сбоем системы и первой реакцией Kubernetes, отражающей его обнаружение;

2) время исправления модуля – период между первой реакцией Kubernetes на сбой и моментом восстановления модуля;

3) время восстановления службы – период между первой реакцией Kubernetes на сбой и моментом возобновления доступности службы;

4) время простоя – общая продолжительность времени реакции и времени восстановления доступности, в течение которого услуга была недоступна [7].

Анализ данных метрик на этапах настройки, изменения или отладки конфигурации способен помочь добиться максимальных условий для обеспечения высокого уровня доступности приложений, развернутых в K8s.

Уровни проверки работоспособности Kubernetes. Рассматривая возможные сценарии сбоев, важно отметить, что Kubernetes предлагает три уровня проверки работоспособности и действий по исправлению для управления доступностью развернутых приложений:

1) прикладной уровень. На данном уровне Kubernetes гарантирует исправность программных компонентов, выполняющихся внутри контейнера, с помощью проверки работоспособности процесса или предопределенных тестов. В случае, если Kubelet обнаружит сбой, онотреагирует в соответствии с определенной политикой перезапуска;

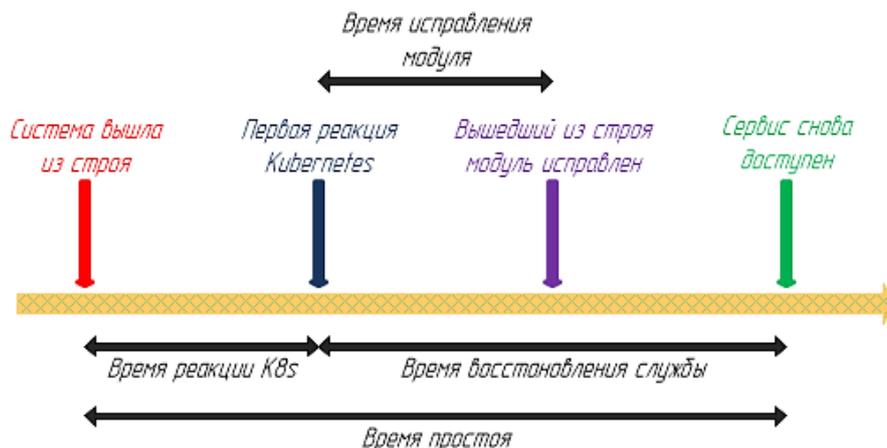


Рис. 2. Метрики доступности K8s при сбое

2) уровень процесса пода. Здесь Kubernetes отслеживает сбои внутри пода, являющегося средой, предоставляемой для запуска контейнеров приложений, путем предоставления общего хранилища и сети для них.

3) уровень узла (ноды). В данном случае Kubernetes отслеживает состояние узлов кластера через свой компонент node controller. Если узел, на котором размещен модуль, выходит из строя, модуль переносится на другой исправный узел.

Данные уровни показывают уязвимые места системы, что важно учитывать при настройке K8s.

Сценарии сбоев. Исходя из уровней проверки работоспособности, можно выделить следующие сценарии сбоев и возможности их воспроизведения для настройки конфигурации:

1) отключение службы из-за сбоя контейнера приложения – для имитации данного сбоя достаточно завершить процесс работы контейнера приложения из операционной системы.

2) сбой в обслуживании из-за сбоя процесса пода – моделирование сбоя возможно с помощью завершения процесса пода из операционной системы (ОС), поскольку при развертывании пода вместе с контейнерами приложений, указанными в его шаблоне, создается один дополнительный контейнер, который является процессом пода. Поскольку модуль сам по себе является процессом в ОС, есть вероятность, что он выйдет из строя [7];

3) сбой в обслуживании из-за нарушения работы узла – узел, на котором размещен модуль, выходит из строя. Можно смоделировать командой перезагрузки Linux или путем выключения узла.

Все эти сбои имеют место в работе любой системы, именно поэтому важно знать их роль в доступности приложения на основе K8s и рассмотреть встроенные возможности по их устранению.

Способы обеспечения доступности с помощью Kubernetes. Рассмотрим основные блоки возможных действий Kubernetes при сбое, которые способны помочь системе вернуться к работоспособности:

1) действия по восстановлению с использованием конфигурации Kubernetes по умолчанию. Сама технология Kubernetes повышает доступность системы путем настройки конфигурации при разворачивании технологии;

2) добавление избыточных экземпляров. Зачастую некоторые разработчики приходят к решению добавить избыточ-

ные экземпляры для повышения доступности, однако важно проанализировать данное решение на этапе разворачивания системы, поскольку это может негативно сказаться на ресурсах и спровоцировать новые риски;

3) восстановление путем использования наиболее гибкой конфигурации K8s.

При настройке Kubernetes есть возможность для указания параметров, способных повлиять на восстановление, например ускорить его. Здесь важно верно учесть ресурсы, внутренние тайминги системы, специфику приложений и микросервисов. Таким образом, система будет использовать наиболее гибкую конфигурацию, что положительно скажется на доступности сервисов.

Помимо этого, важно учитывать средние показатели внутренних таймингов Kubernetes:

1) сигнал завершения, отправляемый в контейнер приложения в сценарии сбоя процесса пода, занимает более 30 с;

2) при сценарии сбоя узла частота отправки статуса узла Kubelet ведущему устройству составляет 10 с, а количество разрешенных пропущенных обновлений статуса перед пометкой узла как неработоспособного равно четырем, что составляет время реакции от 30 до 40 с [8];

3) при выходе модуля из строя и перед созданием нового Kubernetes ожидает около 260 с [8].

Данные тайминги необходимо учитывать, как и остальные аспекты настройки, сценарии, риски и специфики, для настройки K8s, способной обеспечить максимальные показатели доступности.

В настоящий момент Kubernetes является одной из ключевых технологий оркестровки приложений. Одним из факторов роста количества его внедрений является возможность повышения доступности сервисов, контроль их работы. Внутри K8s можно выделить 3 уровня работоспособности, напрямую коррелирующие с возможными сценариями сбоев: прикладной уровень, процесса пода и процесса ноды. При этом разработчик может влиять на процесс восстановления работоспособности сервисов на данных уровнях путем настройки конфигурации по умолчанию, добавления избыточных экземпляров или использования более гибкой конфигурации K8s, но важно отметить, что каждое из решений имеет свои риски, зависящие от приложения, которые должны быть проанализированы перед настройкой системы и в процессе отладки конфигурации.

Список литературы

1. Документация по Kubernetes. [Электронный ресурс]. URL: <https://kubernetes.io/ru/docs/home/> (дата обращения: 25.05.2022).
2. Sebrechts M., Borny S., Wauters T., Volckaert B. Service Relationship Orchestration: Lessons Learned From Running Large Scale Smart City Platforms on Kubernetes // IEEE Access. 2021. Vol. 9. P. 133387–133401.
3. Липатова С.Е., Белов Ю.С. Практики обеспечения кибербезопасности в Kubernetes // E-Scio. 2022. № 1. С. 490–498.
4. Han J., Hong Y., Kim J. Refining Microservices Placement Employing Workload Profiling Over Multiple Kubernetes Clusters. IEEE Access. 2020. Vol. 8. P. 192543–192556.
5. Sebastio S., Ghosh R., Mukherjee T. An Availability Analysis Approach for Deployment Configurations of Containers. IEEE Transactions on Services Computing. 2021. Vol. 14. No. 1. P. 16–29.
6. Липатова С.Е., Белов Ю.С. Принцип работы контроллера statefulset Kubernetes для управления доступностью приложений с отслеживанием состояния на основе микросервисов // Высокие технологии и инновации в науке: сборник избранных статей Международной научной конференции, 2022. С. 112–115.
7. Qi S., Kulkarni S.G., Ramakrishnan K.K. Assessing Container Network Interface Plugins: Functionality, Performance, and Scalability. IEEE Transactions on Network and Service Management. 2021. Vol. 18. No. 1. P. 656–671.
8. Taherizadeh S., Stankovski V., Cho J. Dynamic Multi-level Auto-scaling Rules for Containerized Applications. The Computer Journal. 2019. Vol. 62. No. 2. P. 174–197.

УДК 621.313.8

ВЫБОР МАТЕРИАЛОВ РОТОРА И КОНСТРУКЦИОННЫХ ЧАСТЕЙ ИНТЕГРИРОВАННОГО СТАРТЕРА-ГЕНЕРАТОРА

Уразбахтин Р.Р.

ФГБОУ ВО «Уфимский государственный авиационный технический университет», Уфа,
e-mail: urr98@mail.ru

Актуальной задачей современного авиастроения является повышение топливной эффективности и экологичности летательных аппаратов. Эта задача должна быть решена не только в долгосрочной перспективе, но и на текущем уровне развития технологий и материалов. Поэтапное совершенствование летательных аппаратов происходит в рамках концепции более электрического самолета. Единственным видом энергии на борту летательного аппарата при этом станет электрическая энергия. Ключевой технологией в концепции более электрического самолета является интегрированный стартер-генератор, позволяющий осуществлять запуск авиационного двигателя и обеспечивать электроэнергией потребителей на борту летательного аппарата. Применение интегрированного стартера-генератора позволяет отказаться от отбора пневматической и механической мощностей от авиационных двигателей при условии электрификации агрегатов авиационного двигателя и летательного аппарата. Одним из вопросов, который необходимо решить при проектировании и дальнейшем производстве интегрированного стартера-генератора, является подбор материалов для интегрированного стартера-генератора. В статье рассматривается зона размещения интегрированного стартера-генератора в двухконтурном турбореактивном двигателе. Рассматриваются различные материалы постоянных магнитов, формируются рекомендации по выбору постоянных магнитов для интегрированного стартера-генератора. Рассматриваются материалы конструктивных частей интегрированного стартера-генератора. Рассматриваются подшипниковые опоры, которые могут применяться в интегрированном стартере-генераторе, если это предусмотрено конструкцией. На основе анализа современной научно-технической литературы формируются рекомендации по выбору вышеперечисленных материалов в интегрированном стартере-генераторе, делается вывод о реализуемости интегрированного стартера-генератора на текущем уровне развития материалов и технологий.

Ключевые слова: более электрический самолет, авиационный двигатель, двухконтурный турбореактивный двигатель, интегрированный стартер-генератор, постоянные магниты, подшипниковые опоры

MATERIALS SELECTION FOR THE INTEGRATED STARTER-GENERATOR ROTOR AND CONSTRUCTIVE PARTS

Urazbakhtin R.R.

Ufa State Aviation Technical University, Ufa, e-mail: urr98@mail.ru

An urgent task of modern aircraft industry is to increase the aircraft fuel efficiency and environmental friendliness. This task must be solved not only in the long term, but also at the current technologies and materials development stage. The phased aircraft improvement is taking place as part of the more electric aircraft concept. In this case, the only type of energy on board the aircraft will be electrical energy. A key technology in the more electric aircraft concept is the integrated starter-generator, which allows starting the aircraft engine and providing electricity to consumers on board the aircraft. The integrated starter-generator usage makes it possible to refuse from the pneumatic and mechanical power take off from aircraft engines, provided that the aircraft engine accessories and aircraft systems are electrified. One of the issues that need to be addressed in the design and further production of an integrated starter-generator is the materials selection. The paper considers the integrated starter-generator placement area in a bypass turbojet engine. Various permanent magnets materials are considered, recommendations for the permanent magnet's choice for an integrated starter-generator are formed. The materials of the integrated starter-generator structural parts are considered. Bearing supports are considered, which can be used in an integrated starter-generator, if this is provided for by the design. Based on the analysis of modern scientific and technical literature, recommendations are formed on the choice of the above materials for an integrated starter-generator, a conclusion is made about the integrated starter-generator feasibility at the current materials and technologies development stage.

Keywords: more electric aircraft, aircraft engine, bypass turbojet engine, integrated starter-generator, permanent magnets, bearings

Сегодня активно ведется проектирование интегрированных стартеров-генераторов (ИСГ), выполняющих функции запуска авиационного двигателя (АД) и генерирования электроэнергии в бортовую сеть летательного аппарата (ЛА). ИСГ для различных АД проектируются в работах [1–4]. Основным недостатком этих работ является недостаточное рассмотрение взаимного влияния параметров ИСГ и АД. Например, размеры ИСГ выбираются исходя из размеров об-

ласти его размещения в АД, упоминается возможное воздействие на него высоких температур. При этом не рассматриваются колебания валов АД, особенности пуска АД, равенство ресурсов ИСГ и АД и т.д. В связи с этим существует необходимость формирования подходов к проектированию ИСГ, учитывающих все особенности его расположения непосредственно внутри АД. Научно-исследовательским коллективом ФГБОУ ВО УГАТУ проектируется ИСГ

для двухконтурного турбореактивного двигателя (ТРДД). ИСГ представляет собой сложное техническое устройство, задача обеспечения отказоустойчивости которого является комплексной. В статье рассматривается выбор материалов постоянных магнитов, конструктивных частей и подшипников опор ИСГ.

В связи с большим объемом информации, касающейся выбора материалов для ИСГ или обеспечения их температурной устойчивости, выбор материалов для ИСГ производится по двум большим группам. В настоящей работе рассматриваются материалы для ИСГ, относящиеся к группе ротора – материалы постоянных магнитов (ПМ), материалы конструктивных частей и подшипниковые опоры для ИСГ.

Целью исследования является выявление особенностей размещения ИСГ непосредственно в ТРДД, обусловленных высокими температурами в области размещения ИСГ, и выбор материалов, которые могут применяться при проектировании и дальнейшем производстве ИСГ ТРДД на основе анализа современной научно-технической литературы.

Материалы и методы исследования

Рассмотрим конструкцию ТРДД. Упрощенный вид ТРДД в разрезе с распределением температур изображен на рис. 1. ИСГ размещается в зоне 2 согласно рис. 1.

Температуры, приведенные на рис. 1, являются ориентировочными и приведены для оценки параметров окружающей ТРДД среды. Точное определение температур возможно только для конкретной модели ТРДД. Для каждой модели ТРДД температуры в различных зонах будут отличаться.

При проведении предварительных расчетов на этапе проектирования ИСГ

для определения температуры в зоне его размещения можно воспользоваться формулой

$$T_{03is} = T_{02} \cdot P_K^{\frac{k-1}{k}}, \quad (1)$$

где P_K – степень повышения давления в компрессоре; k – показатель адиабаты.

Показатель адиабаты обычно принимают $k = 1,4$, а степень повышения давления P_K в компрессоре низкого давления (КНД) варьируется от 1,5 до 6 [5, 6]. Для некоторых ТРДД давление в КНД может повышаться в 8 раз. Температура на входе в ТРДД принимается $T_{02} = 288,15K$ исходя из того, что расчет ТРДД производится при условиях старта ЛА в нормальных атмосферных условиях.

На ИСГ длительно может действовать температура около $350^\circ C$, что накладывает серьезные ограничения на материалы, применяющиеся в ИСГ. Поэтому требуется подобрать материалы, которые могут функционировать длительное время под воздействием высокой температуры, и разработать методы, позволяющие применять материалы.

ИСГ представляет собой бесконтактную электрическую машину (ЭМ), которая способна работать как в двигательном, так и в генераторном режиме. Исходя из того, что ИСГ представляет собой ЭМ с ПМ, решается задача выбора материалов или обеспечения температурной устойчивости материалов для ИСГ.

Облик пакета ротора ИСГ определяется исходя из того, что ИСГ представляет собой ЭМ с ПМ. Пакет ротора ИСГ крепится непосредственно на РВД ТРДД. Пакет ротора ИСГ с поверхностным монтажом ПМ представлен на рис. 2, а. Такой пакет ротора состоит из магнитопровода, ПМ и бандажа.

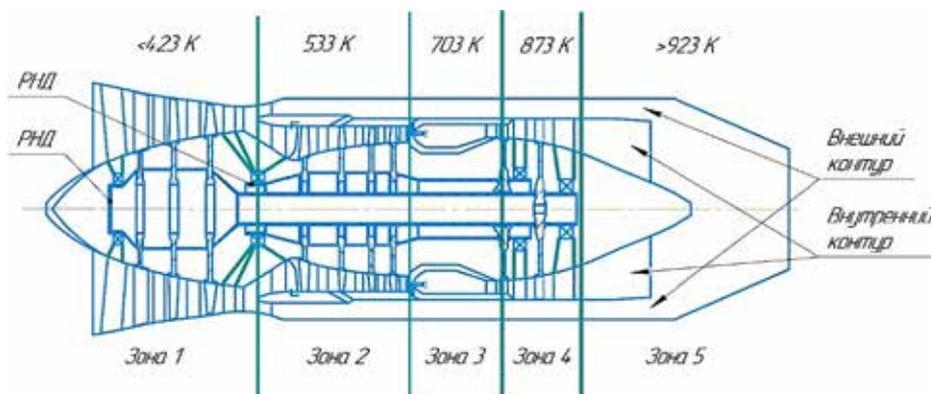


Рис. 1. Конструкция ТРДД с распределением температур:
РНД – ротор низкого давления, РВД – ротор высокого давления

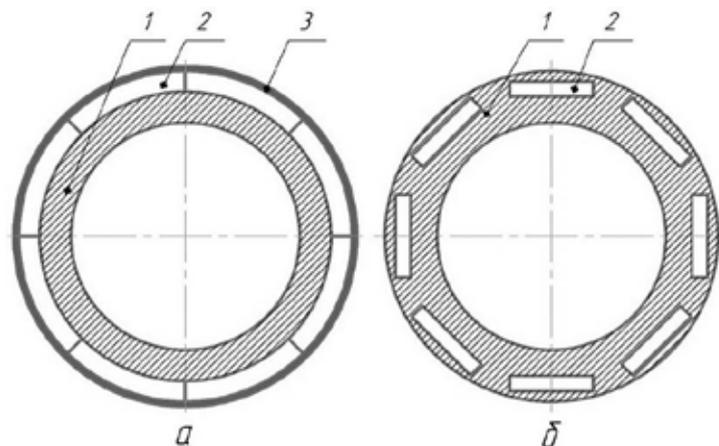


Рис. 2. Пакет ротора ИСГ: 1 – магнитопровод, 2 – ПМ, 3 – бандаж

Пакет ротора ИСГ с инкорпорированными ПМ представлен на рис. 2, б. Такой пакет ротора состоит только из магнитопровода и ПМ. Возможные конструкции пакета ротора ИСГ представлены на рис. 2.

Выполнение магнитопровода будет рассмотрено подробнее в будущей работе автора. В настоящей работе отметим, что магнитопровод необходим для обеспечения замыкания магнитного потока и обеспечения механической прочности ротора ИСГ. ПМ образуют постоянный магнитный поток. Бандаж необходим в случае поверхностного монтажа ПМ, чтобы обеспечить их надежное крепление на роторе. В случае поверхностного монтажа ПМ на них действуют центробежные силы, под действием которых они могут оторваться от магнитопровода. Соответственно, бандаж предотвращает отрывание ПМ от магнитопровода. В конструкции ротора ИСГ с инкорпорированными ПМ фиксируются за счет конструкции магнитопровода, в связи с чем бандаж не применяется.

Группа материалов ротора ИСГ формируется из материалов магнитопровода пакета ротора ИСГ, ПМ, бандажа, материалов конструктивных частей (например, стопорных шайб). Дополнительно в группе материалов ротора рассматриваются подшипниковые опоры, которые могут подразумеваться конструкцией ИСГ. Выполнение магнитопровода будет рассмотрено в будущей работе в связи с ограниченностью объема статьи.

Далее в статье решается задача выбора или обеспечения работоспособности материалов для ИСГ.

Постоянные магниты для ИСГ

Характеристикой ПМ, позволяющей оценить их способность функционировать при высоких температурах, является темпе-

ратура Кюри. ПМ могут функционировать только при температурах ниже температуры Кюри, при этом по мере приближения к температуре Кюри свойства ПМ ухудшаются. Для ИСГ необходимо выбирать ПМ, температура Кюри которых выше 300 °С. При этом необходимо учитывать, что ПМ будут нагреваться не только от окружающей среды, но и от наводящихся в них вихревых токов. Охлаждение ПМ ИСГ затруднено, так как они крепятся на роторе высокого давления (РВД) ТРДД.

Свойства ПМ с изменением температуры могут меняться двумя способами: единоразово или динамически [4]. Разовое изменение может произойти при уменьшении коэрцитивной силы, обусловленном ростом температуры, или из-за структурных изменений материала ПМ (окисления, увеличения размера зерен). В первом случае свойства ПМ можно восстановить за счет перемагничивания ПМ, во втором случае свойства ПМ ухудшаются безвозвратно.

Динамические изменения свойств ПМ описываются коэффициентами α и β для остаточной магнитной индукции и коэрцитивной силы соответственно. За начальную температуру ПМ примем температуру T_1 , за конечную температуру – температуру T_2 . Тогда коэффициенты α и β при изменении температуры от T_1 до T_2 определяются как

$$\alpha = \frac{B_r(T_2) - B_r(T_1)}{B_r(T_1) \cdot (T_2 - T_1)},$$

$$\beta = \frac{H_c(T_2) - H_c(T_1)}{H_c(T_1) \cdot (T_2 - T_1)},$$

где B_r – остаточная магнитная индукция при соответствующей температуре; H_c – коэрцитивная сила ПМ при соответствующей температуре.

Характеристики высокотемпературных ПМ для ИСГ [7–10]

Марка ПМ	Остаточная индукция B_r , Тл, не менее	Коэрцитивная сила по индукции, H_{cb} , кА/м, не менее	Коэрцитивная сила по намагниченности, H_{cp} , кА/м, не менее	Максимальная рабочая температура, °С
КС 25 ДЦ	0,92	617	1440	600
N35АН	1,17	860	2785	240
S32C	1,12	820	2070	350
ЮНДКТ5БА	1,06	119	–	550
S26H	1	716	1989	330

При проектировании ИСГ необходимо предупредить возможность возникновения разового изменения свойств ПМ и учитывать динамическое изменение свойств ПМ на различных режимах работы ИСГ и ТРДД.

Наиболее высокими характеристиками обладают ПМ на основе сплава NdFeB, но они обладают невысокой температурой Кюри (до 310 °С). Для высокотемпературных применений активно исследуются ПМ на основе сплава SmCo, температура Кюри которых доходит до 900 °С. Также перспективным является применение в ИСГ ПМ на основе сплавов на основе сплава Sm-Co-Fe-Cu-Zr. В таблице приводятся свойства ПМ, потенциально претендующих на роль ПМ ИСГ.

Для ПМ, функционирующих при высоких температурах, возможно повышение температурной устойчивости за счет применения специальных покрытий, выполненных из никеля или вольфрама [11]. При высоких температурах происходит ускоренное окисление ПМ, что может приводить к образованию трещин на ПМ. Для роторов ИСГ с поверхностным монтажом ПМ эта проблема может быть решена путем применения бандажа. Здесь необходимо учитывать, что применение металлического бандажа приведет к повышению потерь на вихревые токи, а бандажи из композитных материалов менее устойчивы к длительному воздействию высоких температур. Коэффициент теплового расширения бандажа должен быть приблизительно равен коэффициенту теплового расширения ПМ.

Материалы конструктивных частей ИСГ

Как любая ЭМ, ИСГ содержит в своей конструкции конструктивные и активные части. Конструктивные части участвуют в формировании финального облика ИСГ, участвуют в обеспечении его работы, но при этом не принимают непосредственного участия в процессах преобразования электрической энергии. Конструктивными частями ИСГ могут являться корпус, втул-

ка под пакет ротора, каркасы для крепления трубопроводов системы охлаждения и т.д.

Температуры в некоторых областях ТРДД составляют 800–900 °С, при этом отдельные элементы конструкции ТРДД способны длительное время функционировать при этих температурах. Для выполнения конструктивных элементов ТРДД существует ряд жаропрочных сталей и высокотемпературных сталей.

Для изготовления роторов ТРДД могут применяться, например, стали перлитного класса 20Х3МВФ и 26ХН3М2ФАА, характеризующиеся высокой температурной стабильностью при длительной эксплуатации. Для изготовления крепежа в ТРДД применяются релаксационностойкие стали, например, марок 25Х2МФА и 20Х1М1Ф1ТР. Для изготовления компрессоров ТРДД могут применяться коррозионностойкие стали, например, 14Х17Н2 и 20Х13.

Очевидно, что температуры конструктивных частей ИСГ не превышают температуры конструктивных частей ТРДД. Поэтому сортамент современных материалов позволяет выбрать стали, удовлетворяющие требованиям к конструктивным частям ИСГ.

Подшипниковые опоры ИСГ

Задача выбора подшипниковых опор, обладающих высокой температурной стабильностью, может быть неактуальна при проектировании ИСГ, так как возможно конструктивное исполнение ИСГ на РВД ТРДД, при котором ИСГ не будет иметь собственных подшипниковых опор. Ранее было отмечено, что роторы ТРДД подвержены колебаниям. Расположение пакета ротора ИСГ на РВД ТРДД теоретически может усилить колебания РВД. Соответственно, для стабилизации РВД в месте размещения ИСГ может потребоваться введение дополнительных подшипниковых опор.

С учетом высоких температур в зоне размещения ИСГ перспективным является применение керамических подшипников.

Для обеспечения высокой температурной устойчивости детали керамических подшипников могут покрываться, например, фосфатом марганца. Для подшипниковых опор, функционирующих при высоких температурах, также перспективно применение твердой смазки на основе, например, дисульфида молибдена.

Результаты исследования и их обсуждение

Результаты проведенного исследования показали, что для ИСГ возможно подобрать, при необходимости, подшипниковые опоры, которые будут функционировать при длительном воздействии на них высоких температур. Также результаты проведенного исследования показали, что существуют материалы, из которых возможно изготовить конструктивные части ИСГ.

С ПМ ситуация является более сложной. С одной стороны, существуют ПМ, способные функционировать при температурах до 650°C. С другой стороны, такие ПМ обладают невысокими характеристиками, что заставляет проектировщиков рассматривать магниты SmCo на роль ПМ для ИСГ. При этом необходимо решать задачу охлаждения ПМ ИСГ с учетом их расположения на РВД ТРДД. Разработка ПМ, устойчивых к высоким температурам и сохраняющих при этом высокие характеристики, является одной из важных задач современного материаловедения. Также перспективным является улучшение характеристик ПМ на основе сплава Sm-Co-Fe-Cu-Zr.

Выбор ПМ – сложная инженерная задача для исследователей, вовлеченных в проектирование ИСГ, так как при ее решении надо учитывать необходимость исключения возможности единоразового изменения свойств ПМ, динамическое изменение свойств ПМ и решить вопрос охлаждения ПМ.

Заключение

Реализация ИСГ с точки зрения ПМ, материалов конструктивных частей ИСГ и подшипниковых опор возможна на современном этапе развития технологий. При этом совершенствование свойств ПМ приведет к повышению эффективности ИСГ, что в перспективе дополнительно повысит целесообразность его разработки.

В следующей части статьи автором будет рассматриваться выбор материалов для проектирования и дальнейшего изготовления

активных частей ИСГ. Также будет предложена конструкция обмотки для ИСГ ТРДД.

Результаты исследования в области ПМ для ИСГ показали, что важной инженерной задачей при проектировании ИСГ является проектирование системы охлаждения ИСГ. Эта задача усложняется тем, что пакет ротора ИСГ расположен непосредственно на РВД ТРДД, а возможности жидкостного охлаждения ограничены резервуарами с жидкостями на борту ЛА. Поэтому в будущих работах автора будет решаться задача проектирования системы охлаждения ИСГ ТРДД.

Работа выполнена в рамках гранта в форме субсидий в области науки из бюджета Республики Башкортостан для государственной поддержки молодых ученых, тема «АП-ЭМ-02-21-ГБ».

Список литературы

1. Bhangu B.S., Rajashekara K. Electric Starter Generators: Their Integration into Gas Turbine Engines. IEEE Industry Applications Magazine. 2014. Vol. 20. No. 2. P. 14–22.
2. Zhang G.H., Ma C.Q., Sun H.Y., Zhang L.F., Liu L., Wang K. Optimization design of interior PM starter-generator. 20th International Conference on Electrical Machines and Systems (ICEMS). 2017. P. 1–5.
3. Bojoi R., Cavagnino A., Tenconi A., Vaschetto S. Control of Shaft-Line-Embedded Multiphase Starter / Generator for Aero-Engine. IEEE Transactions on Industrial Electronics. 2016. Vol. 63. No. 1. P. 641–652.
4. Cavagnino A., Li Z., Tenconi A., Vaschetto S. Integrated Generator for More Electric Engine: Design and Testing of a Scaled-Size Prototype. IEEE Transactions on Industry Applications. 2013. Vol. 49. No. 5. P. 2034–2043.
5. Кулагин В.В., Кузьмичев В.С. Теория, расчет и проектирование авиационных двигателей и энергетических установок. М.: Инновационное машиностроение, 2013. 336 с.
6. Мамедов Б.Ш., Штанько П.К. Расчет степени повышения давления осевых компрессоров воздушно-реактивных двигателей // Технологический аудит и резервы производства, 2016. № 2 (1). С. 13–18.
7. ООО «ПОЗ-Прогресс». Продукция. Постоянные магниты. [Электронный ресурс]. URL: <http://www.poz-progress.ru/index.php?page=products&pid=15> (дата обращения: 28.07.2022).
8. ООО «ПОЗ-Прогресс» / XVIII Международная конференция по постоянным магнитам // Металловедение и термическая обработка металлов. 2013. № 2. С. 16–17.
9. ООО «АМТ&С». Постоянные магниты промышленного применения. [Электронный ресурс]. URL: <http://www.amtc.ru/production/magnit/> (дата обращения: 28.07.2022).
10. НПО «ЭРГА». Продукция. Постоянные магниты. Самарий-кобальтовые магниты SmCo. [Электронный ресурс]. URL: <https://erga.ru/catalog/postoyannye-magnity/samariy-kobaltovye-magnity-smco/> (дата обращения: 28.07.2022).
11. Chen C., Walmer M., Liu S. Thermal stability and the effectiveness of coatings for sm-co 2: 17 high-temperature magnets at temperatures up to 550c. IEEE Transactions on Magnetics. 2004. Vol. 40. No 4. P. 2928–2930.

УДК 004.5

ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ИНТЕРФЕЙСА НА ПРИМЕРЕ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ- АССИСТЕНТА ДЛЯ НАСТОЛЬНОЙ ИГРЫ «ДРЕВНИЙ УЖАС» С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Маслова Ю.А., Белов Ю.С., Федоров В.О.

ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана»,
Калужский филиал, Калуга, e-mail: fn1-kf@mail.ru

В данной статье будут рассмотрены фундаментальные и психологические принципы построения игрового интерфейса мобильного приложения на примере наиболее частых ошибок разработки. При разработке мобильного приложения-ассистента для настольной игры, в которой используется технология дополненной реальности, предполагается правильное с точки зрения основных принципов построения создание интерфейса. К фундаментальным принципам построения интерфейса относятся баланс и единообразие элементов, наличие обратной связи, необходимое и достаточное количество текстовой информации на экране, акцентирование внимания пользователя и удобство. А психологическими принципами создания интерфейса являются: движение взгляда пользователя, восприятие изображения лучше текста, разбиение множества элементов на группы, использование привычных обозначений, оптимальное количество информации. Используя комбинацию данных принципов, геймдизайнеры и разработчики приложений могут создавать приложения в том виде, в котором мы привыкли их видеть. Приступая к построению интерфейса мобильного приложения, сначала продумывается вся внутренняя и внешняя логика взаимодействия приложения, создается прототип экранов. Данная статья может иметь интерес для тех, кто хочет заняться непосредственно созданием игровых мобильных приложений, изучать данную область деятельности.

Ключевые слова: принципы построения, мобильное приложение, интерфейс, разработка приложения-ассистента, фундаментальные принципы, психологические принципы

GENERAL PRINCIPLES OF BUILDING THE INTERFACE ON THE EXAMPLE OF DEVELOPING A MOBILE ASSISTANT APPLICATION FOR THE BOARD GAME “ANCIENT HORROR” USING AUGMENTED REALITY TECHNOLOGY

Maslova Yu.A., Belov Yu.S., Fedorov V.O.

Moscow State Technical University named after N.E. Bauman, Kaluga branch, Kaluga,
e-mail: fn1-kf@mail.ru

This article will consider the fundamental and psychological principles of building a game interface for a mobile application using the most common development errors as an example. When developing a mobile assistant application for a board game that uses augmented reality technology, the creation of an interface is assumed to be correct from the point of view of the basic principles of construction. The fundamental principles of building an interface include the balance and uniformity of elements, the presence of feedback, the necessary and sufficient amount of textual information on the screen, focusing the user's attention and convenience. And the psychological principles of creating an interface are: the movement of the user's gaze, the perception of an image better than text, the division of many elements into groups, the use of familiar designations, the optimal amount of information. Using a combination of these principles, game designers and app developers can build apps the way we're used to seeing them. Starting to build the interface of a mobile application, first, all the internal and external logic of the application's interaction is thought out, a screen prototype is created. This article may be of interest to those who want to directly engage in the creation of gaming mobile applications, to study this area of activity.

Keywords: principles of construction, mobile application, interface, development of an assistant application, fundamental principles, psychological principles

Любая разработка интерфейса начинается после того, как сформирован базовый функционал проекта. Благодаря чему игра разбивается на логические части (например, фазы игры, боевой процесс и др.). После определения логических частей выделяют конкретные экраны в игре. При их разработке следует обращать внимание на связи между экранами, их функционал, а также размеры и расположение основных элементов. Такая концепция разработки предполагает использование функциональных и психологических принципов интерфейса.

Цель исследования – изучить основные проблемы и принципы построения и разработки интерфейса мобильного приложения на примере игрового проекта в жанре ролевых игр. Одной из основных проблем, требующих решения, является проблема выбора принципов построения интерфейса мобильного приложения [1]. Существует множество принципов. Однако все они содержат в себе общие концепции [2]. В данной работе были выбраны для описания функциональные и психологические принципы разработки интерфейса мобильного приложения.

Материалы и методы исследования

Для выполнения данной работы использовались различные методы исследования: теоретический анализ, метод обобщения, сравнительный анализ. Материалом исследования являются статьи, опубликованные в открытой печати в электронной версии и доступные бесплатно в полнотекстовом варианте.

Фундаментальные принципы построения интерфейса

Принципы, касающиеся логической структуры приложения, относят к фундаментальным [3].

Достаточно часто можно встретить интерфейсы [4], обладающие большим количеством кнопок, картинок и многого другого. Такое построение объясняется тем, что разработчик не до конца понимает ключевую задачу игрока в выбранном интерфейсе. Он считает, что абсолютно все эти элементы нужны для взаимодействия с данным аспектом игры. Данная проблема определяет ключевой функциональный принцип построения интерфейсов как баланс элементов на игровом экране. Демонстрация экрана, содержащего необходимое и достаточное количество функциональных элементов игрового процесса, представлена на рис. 1.

Следующим принципом является единообразии самих элементов. При разработке интерфейса необходимо осознавать, что практически любой элемент, который необходимо ввести, должен быть универсальным. Например, при добавлении кнопки будьте готовы к тому, что форму и размер этой кнопки придется использовать и в дальнейшем. Нельзя перерисовать новую кнопку для конкретного экрана. Иначе это приведет к тому, что в процессе игры пользователь будет путаться и совершать ошибки, тратить больше времени на выполнение игровой задачи, поскольку в каждом новом окне за привычные функции будут отвечать визуально разные элементы или расположенные в разных местах. Пример использования данного принципа разработки интерфейса приведен на рис. 2, 3.

Третий фундаментальный принцип разработки игрового интерфейса определяет наличие обратной связи. Часто бывает так, что нажатие какой-либо кнопки недоступно в определенные игровые периоды или наполнение какого-то окна отсутствует по какой-либо причине. При обращении пользователя приложения к определенному элементу или экрану игры необходимо предоставить дополнительную информацию, объяснить, почему данный элемент или экран в конкретный момент недоступен.



Рис. 1. Экран главного меню мобильного приложения-ассистента для настольной игры



Рис. 2. Единообразие кнопок



Рис. 3. Демонстрация нового экрана приложения с сохранением созданных ранее кнопок



Рис. 4. Пример подсказок для игрока

Серьезной ошибкой считается показ игроку пустого окна или неработающей кнопки. Решается эта проблема обычно подсказками, которые в зависимости от конкретного интерфейса отображаются в самом окне или при нажатии на какой-либо элемент. Пользователь должен видеть и понимать, что происходит в приложении в данный момент. Можно сообщать о том, что игрок забыл выполнить какое-либо действие. Пример использования таких подсказок на экране рабочего игрового мобильного приложения-ассистента показан на рис. 4. В данном случае используется дополненная реальность для определения игрового персонажа игрока [5, 6].

Небольшое количество текстовой информации на экране составляет четвертый фундаментальный принцип проектирования интерфейса. Лучшим решением будет создание «интуитивно понятного интерфейса» – т.е. такого, в котором можно избавиться от надписей на кнопках или иконках. Любое чтение в момент быстрой навигации внутри окна затрудняет восприятие и делает взаимодействие между игроком и приложением-ассистентом к настольной игре. Если вы делаете иконку, при нажатии на которую открывается карта, то нужно сделать иконку в форме карты и не нужно подписывать ее текстом. Однако большое количество ико-

нок и полный отказ от текста может привести к тому, что интерфейс перестанет быть читабельным.

Следующий принцип – красота не может быть важнее удобства. Геймдизайнеры, создавая интерфейсы, обладающие многообразием украшений, множеством артов, в погоне за красотой часто забывают об эргономике и функциональных особенностях элементов.

Шестым принципом является наличие акцентов. Акцент представляет собой выделение элементов интерфейса цветом, анимацией или размером. Они нужны для концентрации внимания игрока. Поскольку игра – это динамичная, изменяющаяся структура, в которой могут меняться правила, то могут появляться новые возможности и исчезать старые. На такие моменты необходимо обращать внимание игрока.

Психологические принципы построения интерфейса

Это общие принципы, которые непосредственно связаны с тем, как человеческий глаз и мозг воспринимает информацию. Такие принципы связаны с нашими привычками и способами обработки информации, выделением и усвоением главного [7].

Первым принципом построения интерфейса с точки зрения психологии человека является движение взгляда. Общеизвестно,

что при чтении движение глаз осуществляет перемещение слева направо, сверху вниз [8]. Так людям проще воспринимать текстовую информацию. Поэтому при разработке интерфейса для лучшей читабельности следует располагать самые важные элементы на диагонали из левого верхнего угла в правый нижний. Также для лучшего прочтения необходимо использовать крупный шрифт. На рис. 5 представлено описание одного из выбранных игроком действия.

Второй психологический принцип разработки интерфейса основан на том, что любой человек сначала обращает внимание на картинку и только потом на текст, который ей сопутствует. Это связано с тем, что человеческий мозг зрительно воспринимает яркие визуальные образы, а затем, почерпнув оттуда максимальное количество информации, принимается за текст.

Третьим принципом является разбиение большого объема данных на группы. Человек плохо воспринимает большой объем информации сразу, он гораздо лучше усваивает сущности, разбитые на группы. Информацию лучше всего подавать блоками и списками, группируя по общему признаку или задаче, которую выполняют выбранные элементы. Для показа такого принципа обратимся к рис. 6.

В качестве четвертого принципа выступает использование привычных эле-

ментов. В любом типе интерфейсов существуют устоявшиеся и принятые человечеством элементы. Это совершенно не означает, что все интерфейсы как две капли воды похожи один на другой. Хотя, сталкиваясь с незнакомым новым интерфейсом, человек будет стараться «зацепиться» за знакомый и понятный принцип расположения информации или знакомый элемент. Именно по этой причине большинство инвентарей имеют похожую структуру. А элемент закрытия окна или какого-то предмета представляет собой общеизвестный крестик.

Пятым психологическим принципом выступает количество информации. При столкновении пользователя с новым интерфейсом на человека направлено большое количество новой информации, призванной для знакомства пользователя с новыми функциями и обучения игровому процессу. Сложные и многофункциональные экраны необходимо разбить на простые. Данное правило также относится не только к интерфейсам, но и к новой механике игры. Поэтому в данном игровом приложении-ассистенте было выбрано использовать «памятку», которая находится на экране мобильного телефона в любой фазе игрового процесса и на любом игровом экране приложения. Использование памятки представлено на рис. 7.



Рис. 5. Информационный текст, описывающий определенное действие игрового процесса

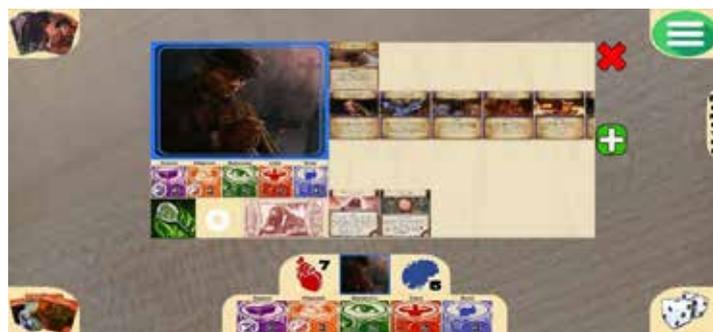


Рис. 6. Разбиение карточек состояний игрока на основные типы и работа с ними

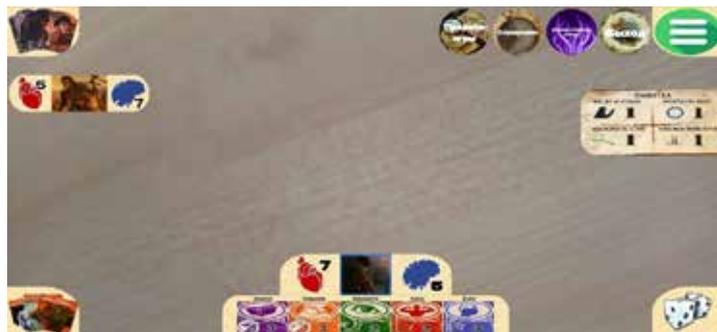


Рис. 7. Раскрытие кнопок игрового экрана приложения и отображение информации, хранящейся в кнопке «Памятка»



Рис. 8. Соединение реальных элементов игры с интерфейсом приложения

Созданный интерфейс приложения ассистента к настольной игре обязан сохранять общий стиль реальных объектов: карточек, фишек и др. Демонстрация игрового процесса с применением мобильного приложения ассистента представлена на рис. 8.

Заключение

Проектирование интерфейсов – это сложная задача, которая требует умения смотреть на проблему в комплексе и очень обдуманно принимать любые решения. В этой статье рассматривается психологический аспект того, как игрок неосознанно воспринимает информацию, а также универсальные законы построения интерфейса, делающие его удобным.

Список литературы

1. Дэрси Л., Кондер Ш. Разработка приложений для Android-устройств. Т. 1: Базовые принципы. М.: Лори, 2014. 402 с.
2. Кулиев Э.В., Григорьева Н.В., Кравченко Ю.А., Довгалев М.А. Разработка интерфейса мобильного приложе-

ния // Известия Кабардино-Балкарского научного центра РАН. 2018. Вып. 6–2. С. 232–239.

3. Хабр. Общие принципы в создании мобильных приложений для начинающего UX/UI-дизайнера. [Электронный ресурс]. URL: <https://habr.com/ru/company/e-legion/blog/275831/> (дата обращения: 17.10.2022).

4. Интерфейс. Виды и типы пользовательского интерфейса. [Электронный ресурс]. URL: http://interfyeis.blogspot.ru/2012/03/blog-post_07.html (дата обращения: 17.10.2022).

5. Маслова Ю.А., Белов Ю.С. Технологии дополненной реальности // E-scio. 2022. № 2. С. 314–323. URL: http://e-scio.ru/wp-content/uploads/2022/03/E-SCIO-2_2022.pdf (дата обращения: 17.10.2022).

6. Маслова Ю.А., Белов Ю.С. Создание кроссплатформенных приложений дополненной реальности в среде unity с помощью .net // Фундаментальные и прикладные исследования. Актуальные проблемы и достижения: сб. статей. СПб.: ГНИИ «НАЦРАЗВИТИЕ», 2022. С. 25–27.

7. Steven Hooper. How Do Users Really Hold Mobile Devices. [Электронный ресурс]. URL: <http://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php> (дата обращения: 17.10.2022).

8. Как двигаются глаза при чтении и почему одни читают быстрее других: рассказывают психолингвисты. [Электронный ресурс]. URL: <https://journal.bookmate.com/chtenie-s-tochki-zreniya-fiziologii-rasskazyvayut-psiholingvist/> (дата обращения: 17.10.2022).

УДК 004.41

ПОДХОД К АДАПТАЦИИ СООБЩЕНИЙ ДЛЯ ОБМЕНА МЕЖДУ АВИАЦИОННЫМИ ПРИЛОЖЕНИЯМИ

^{1,2}Солоделов Ю.А., ^{1,2}Камалетдинова Г.Р., ¹Кожанов К.Д.,
¹Альбицкий Д.В., ¹Трубчанинов И.А.

¹ФАУ «Государственный научно-исследовательский институт авиационных систем»,
Москва, e-mail: grkamaletdinova@2100.gosnias.ru;
²ФГБОУ ВО «Московский авиационный институт
(национальный исследовательский университет)», Москва

Бортовые комплексы управления самолетов состоят из большого числа аппаратно-программных компонентов, взаимодействующих между собой, причем все большее число функций реализуется не отдельными вычислительными системами, а отдельными вычислительными процессами в одном вычислителе под управлением операционной системы. Взаимодействие вычислительных процессов осуществляется через обмен сообщениями. При этом возникает проблема согласованности форматов этих сообщений. Особенно актуальной является эта проблема в случае, когда функция, используемая повторно, уже была квалифицирована. В этом случае ее доработка с целью формирования необходимого вида сообщений потребует повторной квалификации, а значит, дополнительных затрат. Такой подход нецелесообразен. В статье рассматривается альтернативный подход к решению задачи унификации обмена между авиационными приложениями, который позволит упростить интеграцию повторно используемых функций, разработанных для использования операционной системой, поддерживающей стандарт ARINC 653 (Avionics Application Software Standard Interface). Для этого рассматриваются решения, предлагаемые в стандарте FACE (Future Airborne Capability Environment), и анализируются исходные данные. В работе предлагается использование специального программного обеспечения, которое будет выступать в качестве Адаптера обмена и обеспечит информационный обмен между пользовательскими приложениями. В статье описаны предпосылки перехода к такому решению и этапы развития проекта.

Ключевые слова: моделирование, унифицированный обмен, сервисное ПО информационного обмена (Адаптер), операционная система реального времени, бортовые системы, ОСРВ JetOS

APPROACH TO UNIFIED MESSAGING IN AVIATION APPLICATIONS

^{1,2}Solodelov Yu.A., ^{1,2}Kamaletdinova G.R., ¹Kozhanov K.D.,
¹Albitskiy D.V., ¹Trubchaninov I.A.

¹GosNIAS, Moscow, e-mail: grkamaletdinova@2100.gosnias.ru;
²Moscow Aviation Institute, Moscow

On-board aircraft control systems consist of a large number interacting hardware and software components. At the moment, the number of functions implemented as separated computing processes in the frame of one onboard computer controlled by the operating system is growing, comparing to the number of functions implemented on separated computing systems. Computing processes are intercommunicating by messaging. Hence the conformity of messages formats plays a crucial role. This problem becomes especially relevant in terms of functions qualification. E.g., if the function being reused has been already qualified, it cannot be easily modified in order to accept or transmit messages in another format due to complexity of the re-qualification procedures. Thereby this approach is inappropriate. The article discusses an alternative solution for unified messaging in aviation applications, which will simplify the integration of reusable functions designed for ARINC 653 standard (Avionics Application Software Standard Interface) operating systems. Discussed solutions are based on FACE (Future Airborne Capability Environment) standard and initial data analysis. The paper proposes to use special software that acts as an exchange adapter and implements unified messaging between functions. The article describes the idea of this approach and the project development.

Keywords: modeling, unified messaging, service software for information exchange (Adapter), real-time operating system, onboard systems, RTOS JetOS

Бортовой комплекс управления самолетом должен отвечать высоким требованиям по безопасности. Это достигается соблюдением нормативных требований и внешним контролем сертифицирующих органов [1]. При этом весь процесс разработки программного обеспечения строго регламентирован и отступление от него влияет на процесс сертификации, а значит, на возможность дальнейшего использования.

Выполнение целевых функций в комплексах бортового оборудования воздуш-

ных судов обеспечивается строгим соблюдением протоколов информационного взаимодействия между отдельными системами и/или функциональными программными приложениями. В связи с этим при появлении новых систем и/или функций возникает проблема обновления протоколов информационного взаимодействия [2] и их согласованности.

Разработка и квалификация новых функциональных программных приложений и систем является сложной, затратной

процедурой, поэтому усилия направляются на реализацию возможности повторного использования уже квалифицированных приложений: новый функционал может достигаться, к примеру, за счет их комбинации. В этом случае протоколы информационного взаимодействия не будут совпадать с исходными.

Отсюда возникает проблема не просто обновления протоколов (что потребует изменения приложений), а их унификации [3, 4], чтобы формат сообщений мог быть адаптирован под любые взаимодействующие приложения.

При условии неизменности приложений требуется разработка дополнительного программного средства, которое будет трансформировать сообщения необходимым образом.

В связи с этим была изучена практика подхода консорциума FACE [5], разрабатывающего стандарт, который направлен на упрощение интеграции повторно используемого программного обеспечения, как в новые проекты, так и в существующие рабочие проекты [6, с. 1–12].

В рамках данной работы рассматривался бортовой комплекс интегрированной модульной авионики и взаимодействие между приложениями [7–9].

Для моделирования процессов использовались архитектурные решения, предложенные FACE [6, с. 4–12], в рамках которых предлагается выделить сегмент транспортных служб, отвечающий за передачу сообщений [3; 6, с. 49–81]. Основываясь на этих данных и учитывая выбор стандарта ARINC 653 [10, с. 49–75], были сформулированы требования на разработку программного обеспечения, которое позволит реализовать унификацию протоколов информационного взаимодействия [11].

Программное обеспечение было реализовано, а затем включено в комплекс-демонстратор, позволяющий оценить работу программного обеспечения на моделируемых входных и выходных данных и проводить тестирование информационного взаимодействия функциональных программных приложений.

Данная работа направлена на поиск решения по адаптации обмена сообщениями между функциональными приложениями в условиях сертификационных ограничений на модификацию уже квалифицированных программных модулей. В условиях узкоспециализированной среды поиск решения ограничен, однако, анализируя опыт консорциума [5], а также опыт смежных специалистов [3; 4; 12], решение по уни-

фикации формата сообщений посредством специального программного обеспечения представляется наиболее логичным.

Следовательно, целью данной работы стала разработка программного обеспечения, которое служило бы для устранения имеющихся различий в форматах передаваемых/принимаемых сообщений. Для достижения данной цели разработано специальное программное обеспечение, которое включено в комплекс-демонстратор [13].

Материалы и методы решения

В программировании наиболее широко используемым способом организации данных является структура, которая включает в себя логически связанные наборы данных, в том числе и другие структуры. Для обмена информацией между приложениями чаще всего используются сообщения, представляющие собой структуры данных, где смысл данных обычно определяется местом в структуре [9, 12].

Порты передачи данных определяются ARINC 653 [10, с. 4–12], который был выбран для моделирования за удобство формирования сообщений.

Стандарт FACE требует, чтобы все структуры, которыми обмениваются приложения, строго соответствовали определенной модели данных [6, с. 82–86]. Реализация такого подхода требует либо создания приложений с учетом стандарта [6, с. 57–81], либо внесения изменений в уже созданные приложения.

В рамках данной работы требуется увязка уже существующего и квалифицированного функционального программного обеспечения, что накладывает существенные ограничения на возможность изменения этих модулей [1; 2].

Таким образом, необходимо обеспечить преобразование содержимого передаваемых сообщений так, чтобы оно соответствовало ожиданиям принимающего приложения без изменения самого приложения.

Данная задача решается, благодаря разработке специального программного обеспечения – Адаптера [3; 4; 12], который циклически получает и выдает информацию через порты ARINC 653 [10, с. 4–12], используя сервисы операционной системы реального времени [13] и обеспечивает трансформацию сообщений, не затрагивая структуру самого функционального программного обеспечения.

Таким образом, основным методом, используемым для решения поставленной задачи, является преобразование [3; 6, с. 49–73].

Возможными вариантами преобразований являются:

1. Преобразование размещения данных. Данный тип преобразования является наиболее очевидным при обмене структурами. Для реализации этого решения необходимо располагать информацией о порядке расположения данных внутри сообщений. Возможен вариант, когда необходимые выходные данные могут быть получены с помощью нескольких входных или быть частью сообщения, т.е. нет прямого соответствия между выходными и входными сообщениями и/или их частями.

2. Преобразование порядка выдачи данных.

3. Преобразования формата представления данных у источника и приемника. Различия могут быть существенными. Например, часто различается представление вещественных чисел – как по занимаемому размеру, так и по внутренней структуре представления мантиссы и порядка. Стандарт FACE полагает, что все вещественные числа должны быть представлены в формате IEEE 754 [14, с. 6–14] одинарной точности (32 разряда), двойной точности (64 разряда) или расширенной точности (80 разрядов). Может различаться и представление целых чисел в части числа занимаемых разрядов.

4. Преобразование единиц измерения в случае различных представлений данных на выход и прием.

5. Линейное преобразование чисел.

6. Преобразование начала и направления отсчета.

7. Преобразование кодировки (например, в случае обмена текстовыми данными).

8. Преобразование пересчета. Например, пересчет электрических показаний датчика в физические значения.

9. Преобразование систем отсчета (для систем координат или географических данных).

Все необходимые преобразования для настройки корректного информационного взаимодействия Адаптер будет исполнять, используя конфигурационные данные, сформированные заранее.

Адаптер должен допускать изменение значительного объема своих рабочих параметров – в первую очередь размера и числа портов и их характеристик, периодичность процессов, предназначенных для преобразования сигналов, а также состава структур, типов данных переменных в указанных структурах и алгоритмов преобразования [3; 6, с. 49–73]. Изменение рабочих параметров должно осуществляться без изменения исходного кода, с помощью конфигурационного файла.

Кроме декларации единого формата модели данных, стандарт FACE [5; 6, с. 63] определяет требование к обеспечению адаптации между собой различных способов передачи пакетов: Publish/Subscribe (Публикация/Подписка) и Request/Response (Запрос/Ответ), что также требует рассмотрения.

Результаты исследования и их обсуждение

В концепции FACE [6, с. 4–12] предлагается следующая архитектура:

- сегмент операционной системы (Operating System Segment – OSS),

- сегмент перемещаемых компонентов (Portable Components Segment – PCS),

- сегмент служб, специфичных для платформы (Platform-Specific Services Segment – PSSS),

- сегмент транспортных служб (Transport Services Segment – TSS),

- сегмент ввода/вывода (I/O Services Segment – I/O SS).

В настоящее время в российской авиационной промышленности отсутствует стандарт, аналогичный FACE, и разрабатываемые приложения учитывают только стандарт ARINC 653 [9; 10]. Поэтому предложено реализовать задачи преобразования информации от/для устройств ввода/вывода, решаемые в PSSS и TSS с помощью Адаптера [3; 8].

На данном этапе специальная программа-Адаптер берет на себя функции унификации и адаптации, что позволяет на данном этапе иметь адаптеру требуемый функционал без создания дополнительных программных средств.

В зависимости от задач, которые планируется решать с помощью Адаптера, необходимо формирование расписания вызова и определение порядка обслуживания портов. В частном случае, Адаптер, привязанный к конкретному приложению, может быть вызван непосредственно перед самим приложением для обработки его входящих сообщений.

Входные и выходные порты Адаптера увязываются с входными и выходными портами функциональных приложений, так же как входные порты одного функционального приложения и выходные порты другого увязываются через Адаптер.

Для апробации возможностей Адаптера решено построить его в процесс информационного обмена между функциональным программным обеспечением «автопилота» (AFA), системой самолетовождения (Flight Management System – FMS) и внешним окружением [3; 8; 9]. Стоит отметить,

что Адаптер позволяет работать и с другими функциональными приложениями. Для этого необходимо произвести настройку конфигурационных файлов.

Программное обеспечение Адаптера реализовано для запуска под операционной системой реального времени (ОСРВ) JetOS [9; 11; 13], разработанной ФАУ «ГосНИИАС» и используемой для моделирования других компонентов.

При подготовке запуска информационного обмена с использованием адаптера необходимо включить в состав интеграционного проекта те модули, между которыми должен быть настроен обмен, а также подготовить и настроить конфигурационные и настроечные файлы.

Важной особенностью реализации является возможность реализовать наличие внешней среды, что является актуальным на этапе включения моделей в комплекс-демонстратор.

Для упрощения создания нового проекта информационного обмена на основе шаблона разработан скрипт, заполняющий изменяемые данные на основе пользовательского ввода и дающий дальнейшие инструкции по заполнению конфигурационных и настроечных файлов.

Процесс работы скрипта в окне терминала представлен на рис. 1. Настройка

производится в соответствии с рекомендациями материалами, доступными пользователям.

В частности, вводятся имена взаимодействующих модулей (приложения, участвующие в обмене), ограничения на размер сообщения без очереди, длительность, задается файл с заголовками функций трансформации, добавляется файл с определением функций трансформации, где функции соответствуют определенным сигнатурам, подключается библиотека поддержки функций, обновляется, если необходимо, атрибут процесса «Период».

При запуске такой конфигурации информационного взаимодействия функционального программного обеспечения системы FMS и системы AFA в терминале может отслеживаться процесс информационного обмена.

Пример отладочной печати приведен на рис. 2.

Дальнейшим развитием данного решения является его работа в рамках демонстратора информационного обмена, где устанавливается взаимодействие с внешней средой. При этом оценивается корректность работы Адаптера и приложений, запущенных в операционной системе реального времени JetOS.

```

user@isp-vm:/media/sf_CommonFolder/other-works/TSS-project$ ./tss_launcher
*****
*          TRANSPORT SERVICES SEGMENT          *
*****
*   Введите 1 - Запустить существующий проект   *
*   Введите 2 - Очистить данные сборки и запуска *
*   Введите 3 - Создать новый проект           *
*****

3
Введите имя нового проекта
AFA_FMS
Введите частоту [Гц] отправки сообщений каждым приложением
Возможно выбрать частоту 5, 10, 20, 40, 50 или 100 Гц
50
Введите имя первого приложения, участвующего в информационном обмене (P1):
AFA
Введите имя второго приложения, участвующего в информационном обмене (P2):
FMS
Введите максимальный (наибольший) размер сообщения в порту без очереди:
1024
Введите имя заголовочного файла (без расширения), который содержит объявление функций трансформации и описание типов передаваемых данных:
AFA_FMS.h
Новый проект создан. Все введенные данные учтены. Осталось добавить:
- Файлы с исходным кодом общающихся приложений
- Файлы с описанием передаваемых данных и функций трансформации
- Подробнее смотреть пункты 3.4 - 3.6 MANUAL.txt
Для запуска (после добавления файлов):
1) выполните повторно команду ./tss_launcher
2) выберите вариант под цифрой 1
3) затем укажите имя созданного проекта
user@isp-vm:/media/sf_CommonFolder/other-works/TSS-project$

```

Рис. 1. Интерфейс создания нового проекта информационного обмена

```

COM0,single,USR>
COM0,single,USR> -----_start_connection_-----
COM0,single,USR> stubAFA: sending msg:
COM0,single,USR> stubAFA: HDG_TRK_ref = 4086.000000
COM0,single,USR> stubAFA: ALT_ref = 2749.000000
COM0,single,USR> stubAFA: VS_ref = 12767.000000
COM0,single,USR> stubAFA: FPA_ref = 9084.000000
COM0,single,USR> stubAFA: AT_THR_TGT_L = 0.000000
COM0,single,USR> stubAFA: AT_THR_TGT_L_V = 0
COM0,single,USR> stubAFA: AT_THR_TGT_R = 25089.000000
COM0,single,USR> stubAFA: AT_THR_TGT_R_V = 1
COM0,single,USR> stubAFA: FGCP_SPD = 26966.000000
COM0,single,USR> stubAFA: FGCP_MACH = 4978.000000
COM0,single,USR> stubAFA: FGCP_HDG = 20495.000000
COM0,single,USR> stubAFA: FGCP_ALT = 10311.000000
COM0,single,USR> stubAFA: FGCP_VS = 11367.000000
COM0,single,USR> stubAFA: FGCP_FPA = 30054.000000
COM0,single,USR> stubAFA: AP_DSCR_1 = 00000000000000000010010000000000
COM0,single,USR> stubAFA: AP_DSCR_2 = 00000000000000000010000000000101
COM0,single,USR> stubAFA: AT_DSCR_1 = 00000000000000000000000000000110
COM0,single,USR>
COM0,single,USR> stubAFA: idOutputPort1 new message send
COM0,single,USR> TSS: INPUT_THERE_PORT new message read and will transform and send in OUTPUT_THERE_PORT
COM0,single,USR> TSS: new message transformed
COM0,single,USR> TSS: OUTPUT_THERE_PORT new message send
COM0,single,USR> stubFMS: new msg was read:
COM0,single,USR> stubFMS: AP_HDG_TGT = 4086.000000
COM0,single,USR> stubFMS: AP_ALT_TGT = 2749.000000
COM0,single,USR> stubFMS: AP_VS_TGT = 12767.000000
COM0,single,USR> stubFMS: AP_FPA_TGT = 9084.000000
COM0,single,USR> stubFMS: AP_THR_TGT_01 = 0.000000
COM0,single,USR> stubFMS: AP_THR_DEF_01 = 0
COM0,single,USR> stubFMS: AP_THR_TGT_10 = 25089.000000
COM0,single,USR> stubFMS: AP_THR_DEF_10 = 1
COM0,single,USR> stubFMS: AP_CAS_TGT = 26966.000000
COM0,single,USR> stubFMS: AP_MACH_TGT = 4978.000000
COM0,single,USR> stubFMS: AP_TRK_TGT = 20495.000000
COM0,single,USR> stubFMS: W_302 = 00000000000001010000000000000000
COM0,single,USR> stubFMS: W_303 = 000000000000010011010000000000
COM0,single,USR>
COM0,single,USR> stubFMS: sending msg:
COM0,single,USR> stubFMS: CAS_CMD = 14535.500000
COM0,single,USR> stubFMS: MACH_CMD = 10668.000000
COM0,single,USR> stubFMS: ROLL_CMD = 0.000000
COM0,single,USR> stubFMS: VgpDev = 0.000000
COM0,single,USR> stubFMS: ALT_CMD = 0.000000
COM0,single,USR> stubFMS: FPA_CMD = 1590.500000
COM0,single,USR> stubFMS: TRUE_RST = 1
COM0,single,USR> stubFMS: VERT_GUID_READY = 1
COM0,single,USR> stubFMS: LAT_GUID_READY = 1
COM0,single,USR> stubFMS: CAS_RST = 0
COM0,single,USR> stubFMS: FPA_RST = 1
COM0,single,USR> stubFMS: SPD_THR_RST = 0
COM0,single,USR> stubFMS: ALT_RST = 1
COM0,single,USR> stubFMS: LANDING_RST = 1
COM0,single,USR> stubFMS: HDG_RST = 0
COM0,single,USR> stubFMS: MACH_RST = 1
COM0,single,USR> stubFMS: ROLL_RST = 0
COM0,single,USR>
COM0,single,USR> stubFMS: idOutputPort1 new message send
COM0,single,USR> stubAFA: preparing to read msg
COM0,single,USR> TSS: INPUT_BACK_PORT new message read and will transform and send in OUTPUT_BACK_PORT
COM0,single,USR> TSS: new message transformed
COM0,single,USR> TSS: OUTPUT_BACK_PORT new message send
COM0,single,USR> stubAFA: preparing to read msg
COM0,single,USR> stubAFA: new msg was read:
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_103_SelectedAirspeed = 14535.500000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_106_1_TargetMach = 10668.000000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_121_1_HorizontalCommand = 0.000000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_174_1_ILS_GLIDE_DEV = 0.000000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_102_1_SelectedAltitude = 0.000000
COM0,single,USR> stubAFA: CP1B_FMS_FCSU_322_1_TargetFlightPathAngle = 1590.500000
COM0,single,USR> stubAFA: FMS_DSCR_270 = 00000000000001010000000000000000
COM0,single,USR> stubAFA: FMS_DSCR_371 = 00000000000000000000010000000000
COM0,single,USR> stubAFA: FMS_DSCR_V = 00001111000000000000000000001100
COM0,single,USR> -----_end_connection_-----
COM0,single,USR>

```

Рис. 2. Пример отладочной печати информационного обмена

Заключение

В работе рассмотрен подход к унификации формата сообщений, которыми обмениваются авиационные функциональные приложения.

В рамках данного проекта:

1. Обоснована необходимость разработки адаптера информационного обмена.
2. Определена среда реализации проекта.

3. Рассмотрены сообщения, передаваемые моделями модулей системы FMS и системы AFA.

4. Разработаны форматы конфигурационных и настроечных файлов.

5. Создан адаптер, обеспечивающий:
 – автоматическую транспортировку сообщений между приложениями;

– автоматическое преобразование содержимого сообщений;

– наличие функций, облегчающих реализацию преобразования.

6. Апробирован адаптер для обмена сообщениями между системами FMS и AFA.

7. Адаптер включен в состав демонстратора информационного обмена унифицированных программных компонентов на базе операционной системы JetOS.

Список литературы

1. Квалификационные требования. Часть 178С. Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники. Введено в действие с 22.11.2016. М.: Авиарегистр МАК, 2016. 105 с.
2. Batuwangala E., Kistan T., Gardi A., Sabatini R. Certification challenges for next-generation avionics and air traffic management systems. *IEEE Aerospace and Electronic Systems Magazine*. 2018. Vol. 33. No. 9. P. 44–53
3. Naushad F., Dehriya S. Unified Interconnect System for Next Generation Fighter Aircraft. *International Journal of electronics and communication technology*. 2015. Vol. 6. No. 3. P. 95–100
4. Кравченко О. Обмен сообщениями – стремление к унификации // Компьютер-пресс. 2011. № 11. URL: <https://compress.ru/article.aspx?id=12262&ysclid=19n6vm5gbf983054939> (дата обращения: 05.10.2022).
5. The Open Group FACE™ Consortium. FACE FAQs. [Электронный ресурс]. URL: <https://www.opengroup.org/content/future-airborne-capability-environment-face/faqs> (дата обращения: 05.10.2022).
6. Technical Standard for Future Airborne Capability Environment (FACE™). Edition 3.0. The Open Group. 2017. 569 p.
7. Буздалов Д.В., Зеленев С.В., Корныхин Е.В., Петренко А.К., Страх А.В., Угненко А.А., Хорошилов А.В. Инструментальные средства проектирования систем интегрированной модульной авионики // Труды ИСП РАН. 2014. [Электронный ресурс]. URL: https://www.ispras.ru/proceedings/docs/2014/26/1/isp_26_2014_1_201.pdf (дата обращения: 05.10.2022).
8. Книга В.Е., Жаринов И.О., Богданов А.В., Виноградов П.С. Принципы организации архитектуры перспективных бортовых цифровых вычислительных систем авионики // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 2 (84). [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/printsiyu-organizatsii-arhitektury-perspektivnyh-bortovyh-tsifrovyyh-vychislitelnyh-sistem-v-avionike/viewer> (дата обращения: 05.10.2022).
9. Маллачиев К.М., Пакулин Н.В., Хорошилов А.В. Устройство и архитектура операционной системы реального времени // Труды ИСП РАН. 2016. № 28 (2). С. 181–192.
10. ARINC Specification 653 Avionics Application Software Standard Interface Set. SAE-ITC. 2019. Vol. 1. No. 5. 118 p.
11. ГОСНИИАС.2200.100.1388-001СП. Спецификация требований к сервисному ПО, обеспечивающему информационный обмен унифицированных функциональных модулей. М.: ГосНИИАС, 2020. 18 с.
12. Молев А.А. Метод автоматического формирования телекоммуникационных модулей структурных элементов автоматизированных систем на основе XML-описания // Программные и аппаратные средства. 2017. № 1. [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/metod-avtomaticheskogo-formirovaniya-telekommunikatsionnyh-moduley-strukturnyh-elementov-avtomatizirovannyh-sistem-na-osnove-xml/viewer> (дата обращения: 05.10.2022).
13. Инструкция по эксплуатации (JetOS Manual Guide). Ревизия 0.53. М.: ГосНИИАС, ИСП РАН, 2022. 119 с.
14. IEEE 754 Standard for Floating-Point Arithmetic. IEEE Std. 2008. 70 p.