

Программная инженерия



Пр **6**
ИН **2022**
Том 13

Рисунки к статье N. M. Borgest, S. A. Vlasov
 «USING VIRTUAL REALITY TECHNOLOGY IN THE DESIGN
 OF COMPLEX SYSTEMS»

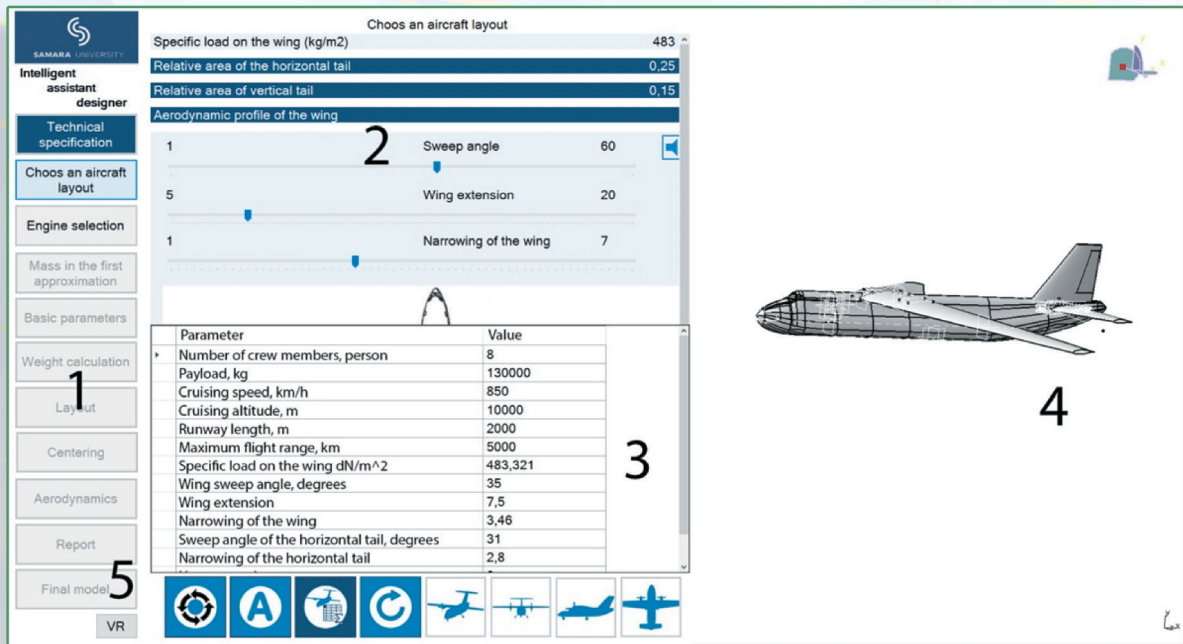


Fig. 2. Interface of the «Robot Designer»

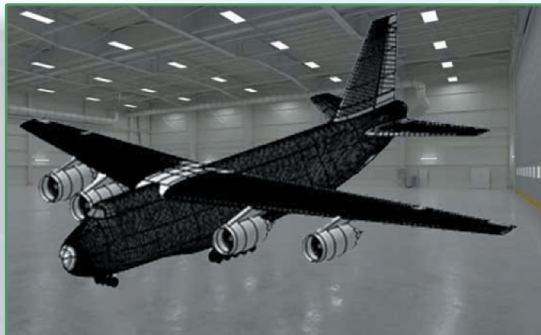


Fig. 3. Model AN-124, VR

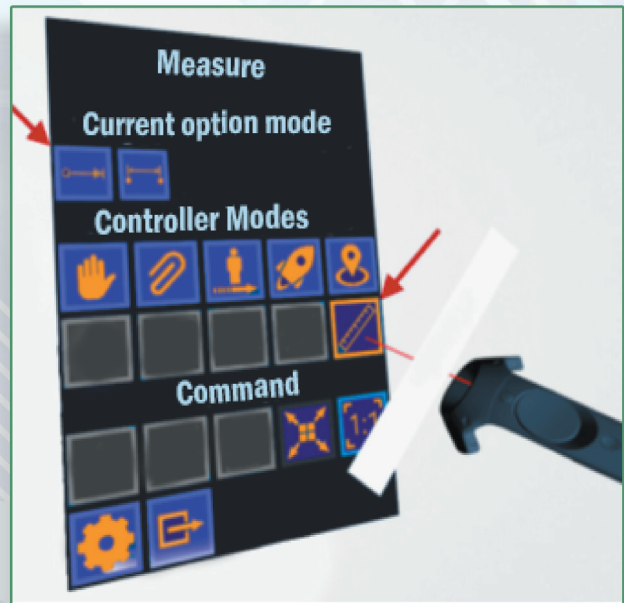


Fig. 4. Interface of the user menu

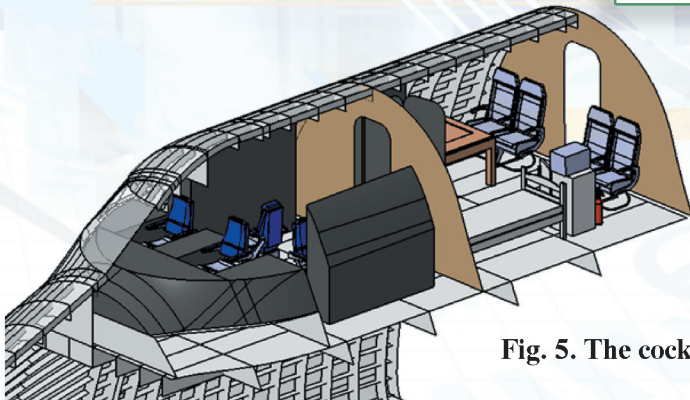


Fig. 5. The cockpit

Программная инженерия

Пр
ИН
Том 13
№ 6
2022

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Макаров В.Л., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назирова Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.А., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана

СОДЕРЖАНИЕ

- Романова Д. С., Непомнящий О. В., Легалов А. И., Рыженко И. Н., Сиротинина Н. Ю.** Методы редукции параллелизма в процессе высокоуровневого синтеза цифровых интегральных схем 259
- Третьяк А. В., Верещагина Е. А., Струк П. В.** Разработка системы символьных макрокоманд (симкоманд) процессора 272
- Avdoshin S. M., Pesotskaya E. Y., Kuruppuge D. M.** Software Engineering Education: Discovering Future Skills 277
- Borgest N. M., Vlasov S. A.** Using Virtual Reality Technology in the Design of Complex Systems 286
- Козицын А. С.** Алгоритмы тематического поиска данных в наукометрических системах 291
- Palchevsky E. V., Antonov V. V.** Decision Support System based on Application of the Second Generation Neural Network 301

Журнал зарегистрирован
в Федеральной службе
по надзору в сфере связи,
информационных технологий
и массовых коммуникаций.
Свидетельство о регистрации
ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в подписных агентствах (индекс по Объединенному каталогу "Пресса России" — 22765) или непосредственно в редакции (для юридических лиц).

Тел.: (499) 270-16-52.

Http://novtex.ru/prin/rus E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

© Издательство "Новые технологии", "Программная инженерия", 2022

SOFTWARE ENGINEERING

PROGRAMMAYA INGENERIA

Vol. 13

N 6

2022

Published since September 2010

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.),
Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad.
RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.),
Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.),
Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
AFONIN S.A., Cand. Sci. (Phys.-Math)
BURDONOV I.B., Dr. Sci. (Phys.-Math)
BORZOV JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A.V., Cand. Sci. (Phys.-Math)
GAVRILOV A.V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.),
Switzerland
KORNEEV V.V., Dr. Sci. (Tech)
KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
MANCIVODA A.V., Dr. Sci. (Phys.-Math)
NAZIROV R.R., Dr. Sci. (Tech)
NECHAEV V.V., Cand. Sci. (Tech)
NOVIKOV B.A., Dr. Sci. (Phys.-Math)
PAVLOV V.L., USA
PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
PETRENKO A.K., Dr. Sci. (Phys.-Math)
POZDNEEV B.M., Dr. Sci. (Tech)
POZIN B.A., Dr. Sci. (Tech)
SEREBR'YAKOV V.A., Dr. Sci. (Phys.-Math)
SOROKIN A.V., Cand. Sci. (Tech)
TEREKHOV A.N., Dr. Sci. (Phys.-Math)
FILIMONOV N.B., Dr. Sci. (Tech)
SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
SHCHUR L.N., Dr. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: CHUGUNOVA A.V.

CONTENTS

- Romanova D. S., Nepomnyashchiy O. V., Legalov A. I., Ryzhenko I. N., Sirotinina N. Y.** Parallelism Reduction Methods in the High-Level VLSI Synthesis Implementation 259
- Tretiak A. V., Vereschagina E. A., Struk P. V.** Developing a System of Symbolic Macro Commands (Symcommands) of the Processor . . . 272
- Avdoshin S. M., Pesotskaya E. Y., Kuruppuge D. M.** Software Engineering Education: Discovering Future Skills 277
- Borgest N. M., Vlasov S. A.** Using Virtual Reality Technology in the Design of Complex Systems 286
- Kozitsyn A. S.** Thematic Search Algorithms for Experts, Journals and Conferences 291
- Palchevsky E. V., Antonov V. V.** Decision Support System based on Application of the Second Generation Neural Network 301

Д. С. Романова^{1,2}, аспирант, ассистент, daryaooo@mail.ru,
 О. В. Непомнящий¹, канд. техн. наук, проф., зав. кафедрой, 2955005@gmail.com,
 А. И. Легалов³, д-р техн. наук, проф., legalov@mail.ru,
 И. Н. Рыженко¹, ассистент, rodgi.krs@gmail.com,
 Н. Ю. Сиротинина¹, канд. техн. наук, доц., nsirotinina@sfu-kras.ru

¹ Сибирский федеральный университет, Красноярск

² Красноярский государственный аграрный университет

³ Национальный исследовательский университет "Высшая школа экономики", Москва

Методы редукции параллелизма в процессе высокоуровневого синтеза цифровых интегральных схем¹

Рассмотрены проблемы и решения в области обеспечения архитектурной независимости и организации процесса сквозного проектирования цифровых интегральных схем. Представлены метод и язык параллельного программирования для функционально потокового синтеза проектных решений. При реализации метода функционально-потокового синтеза выделены задачи редукции параллелизма и оценки занимаемых ресурсов. Предложен способ свертки (сокращения) параллелизма, базирующийся на введении дополнительного слоя в процесс синтеза. Разработаны принцип и алгоритмы редукции параллелизма. Представлены результаты разработки программного инструментария поддержки проектирования и реализованные на практике проекты СБИС.

Ключевые слова: интегральная схема, модель параллельных вычислений, алгоритм, программа, высокоуровневый синтез, функционально-потоковый язык

Введение

Современный уровень развития цифровых интегральных схем (ЦИС) характеризуется постоянно возрастающими требованиями к системной организации всего процесса проектирования. В связи с этим одной из основных задач проектирования является обеспечение архитектурной независимости (первая задача), т. е. переносимости проектных решений между целевыми платформами ЦИС. Не менее важным является сокращение сроков получения конечного результата (вторая задача), что обеспечивается максимально возможным исключением из процесса проектирования итерационных операций возврата к предыдущим этапам, т. е. обеспечение процесса "сквозного проектирования" в процессе всех стадий маршрута проектирования ЦИС (структурного, поведенческого и физического уровней). На рис. 1 приведен весь маршрут проектирования ЦИС, где выделены первые пять этапов, иллюстрирующие место

VLSI-дизайна (*Very Large Scale Integration*). В случае обеспечения архитектурной независимости про-

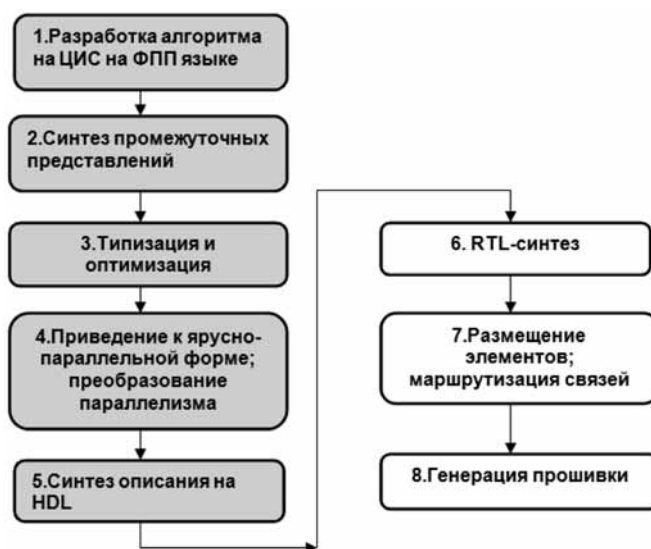


Рис. 1. Маршрут проектирования ЦИС:

ФПП язык — функционально-потоковый параллельный язык; HDL — *Hardware Description Language* (язык описания аппаратуры); RTL — *Register Transfer Level* (уровень регистровых передач)

¹ Статья подготовлена по материалам доклада на Седьмой Международной конференции "Актуальные проблемы системной и программной инженерии" АПСПИ 2021.

ектных решений переносимость позволит отказаться от привязки к конечной форме реализации на верхних уровнях абстракции проекта, когда происходит отработка алгоритмов функционирования. Это обеспечит наиболее эффективный выбор целевой платформы для конечной реализации при переходе на нижние уровни проектирования. Тем самым обеспечиваются наиболее оптимальные технические характеристики, такие как быстродействие, занимаемая площадь кристалла, энергопотребление и т. п. В случае сокращения сроков получения итогового результата сквозное проектирование ведет к уменьшению финансовых затрат и обеспечению конкурентоспособной продукции.

Интегральная схема, по существу, является системой параллельной обработки информационных потоков, а алгоритмы ее функционирования и архитектура на заключительных этапах синтеза представляются на языках описания аппаратуры. Поэтому эффективные решения по обеспечению архитектурной независимости могут быть найдены в области переносимых параллельных программ. Это позволит не только обеспечить требуемые технические характеристики СБИС, но и автоматизировать некоторые стадии работы над проектом. В свою очередь сквозное проектирование можно обеспечить за счет использования модели функционально-поточковых параллельных вычислений и описания исходных алгоритмов в виде ациклических конструкций [1–4]. Описание исходных алгоритмов на ФПП языке происходит на первоначальном этапе в маршруте проектирования СБИС.

Ключевой особенностью известных исследований в данном направлении является использование соответствующей модели вычислений и языка для описания исходных алгоритмов функционирования ЦИС.

1. Известные решения

В настоящее время наблюдается устойчивая тенденция повышения уровня абстракции исходных алгоритмов от конечной реализации проекта. При этом развитие способов описания архитектуры ЦИС идет несколькими путями. Наиболее распространенным является введение в существующие языки описания аппаратуры (HDL) конструкций для высокоуровневого описания [5]. Такие решения привели к появлению на базе классического Verilog языка SystemVerilog [5]. Но, как и VHDL, SystemVerilog позволяет описывать системы на алгоритмическом уровне и применяется только на низких уровнях абстракции,

где имеется привязка программы к определенной архитектуре, что обуславливает наличие семантического разрыва между представлениями схем на высоких уровнях и воплощении в кристалл.

Существует ряд решений, поддерживающих использование адаптированных, императивных языков программирования высокого уровня (в первую очередь С и С++) в качестве языков описания аппаратуры, например, SystemC [6], Handel-C [7] и Impulse-C [8]. Но стоит отметить, что эти языки создавались для решения узких задач, например, для реализации потоковых приложений или для поддержки альтернативных моделей программирования, и при моделировании и отладке системы происходит жесткая привязка к конкретному архитектурному решению.

Особый интерес вызывает язык программирования COLAMO [1], который является языком высокого уровня с неявным описанием параллелизма, а распараллеливание достигается с помощью объявления типов доступа к переменным и индексации элементов массивов, что характерно для языков потока данных. В настоящее время этот язык применяется для программирования реконфигурируемых вычислительных систем и позволяет создавать параллельные прикладные программы с высокой удельной производительностью [1]. Однако этот язык ориентирован на решение прикладных задач в области высокопроизводительных вычислений для многокристалльных систем.

Для описания ЦИС есть возможность использования и функциональных языков, которые имеют ряд преимуществ по сравнению с императивными языками. Во-первых, более мощный механизм абстракции позволяет таким языкам справляться с решением сложных задач. Во-вторых, система типов современных функциональных языков обеспечивает повышенную надежность вычислений. Исходные алгоритмы ЦИС, описанные на подобных языках, более лаконичны, проще поддаются преобразованиям и верификации [9]. Например, в Hydra [10] используются потоки для описания сигналов и рекурсивных выражений для обеспечения преобразований в схеме. Для описания схем на логическом уровне удобно использовать язык Wired [11]. Последняя разработка программистов для этого языка — библиотека Haskell — за счет использования отношений вместо функций обеспечивает более простое создание логических типов данных, которым могут быть поставлены в соответствие некоторые паттерны из библиотеки. Получающиеся описания схем сильно параметризованы, что поднимает вопрос о том, как предоставить общие методы проверки [11]. Язык

Chisel [12], являясь встроенным в язык Scala, позволяет применить всю мощь объектно-ориентированного и функционального программирования к разработке логических схем. Программы, описанные на Chisel, могут быть преобразованы на Verilog. Однако вследствие нечитаемости сгенерированного на Verilog кода зачастую затруднена его отладка.

Также как и Hydra, язык Lava [13] является встроенным подмножеством языка Haskell и имеет мощные средства для проектирования схем, полученные от своего предшественника. При этом благодаря расширенной системе типов для описания аппаратуры Lava является более простым и удобным в использовании. Однако свойственный Haskell механизм "ленивых" вычислений и последовательная структура списков накладывают ограничения на преобразования параллелизма и не позволяют эффективно осуществить автоматическое распараллеливание программ, т. е. добиться полной "свободы" при преобразовании параллелизма с учетом требуемых ограничений.

Таким образом, задача обеспечения эффективной переносимости исходного описания ЦИС может быть решена при использовании функциональных языков и описания алгоритмов в виде графа потока данных, что показано в работах Доннагари [2, 14]. На примере параллельного программного обеспечения (PaRSEC) для высокопроизводительных вычислений на практике показана высокая эффективность данного подхода [2].

В этом случае на этапах высокоуровневого синтеза решение лежит в применении определенной парадигмы программирования, которая должна соответствовать следующим условиям [2]:

- 1) отсутствие явного управления вычислениями;
- 2) поддержка модели потока данных;
- 3) параллелизм на уровне операций.

Такая модель с параллелизмом на уровне операций для обработки потока данных, без использования явного управления вычислениями лежит в основе ФПП языков [3]. Это дает основание считать методы ФПП программирования и соответствующую модель вычислений наиболее соответствующей решаемым задачам высокоуровневого, архитектурно-независимого синтеза СБИС.

Среди функциональных языков, поддерживающих ФПП программирование, выделен язык ПИФАГОР [3], который обеспечивает исходное описание с бесконечными ресурсами, поддерживает модель потока данных и параллелизм на уровне операций, что является наиболее значимым для организации процесса архитектурно-незави-

симого, сквозного проектирования ЦИС. В этом языке архитектурная независимость достигается за счет описания в программе только информационных связей. В отличие от базовых возможностей Haskell, ПИФАГОР поддерживает только явное описание задержанных вычислений. Это позволяет не допускать выполнения альтернативных фрагментов программы до того момента, пока они не потребуются. Кроме того, в ПИФАГОРе отсутствуют операторы цикла, что позволяет избежать конфликтов при использовании различных данных одних и тех же фрагментов параллельной программы. Теоретически это позволяет начинать выполнение любых функций сразу же по готовности их данных. Практически же на максимальный параллелизм, задаваемый языком, накладываются только ресурсные ограничения. Примеры кода, демонстрирующие реализацию мультиплексора 2 к 1 на языках ПИФАГОР и Verilog, представлены в листингах 1 и 2.

```
// Функция, реализующая логический элемент "Не"
Not << funcdef Param {
    return << Param: ~; }
// Функция, реализующая логический элемент "2-И"
And2 << funcdef Param {
    return << Param: *; }
// Функция, реализующая логический элемент "ИЛИ"
Or2 << funcdef Param {
    return << Param: +; }
MUX2_1 << funcdef Param { // Основная функция,
    реализующая мультиплексор 2 к 1
    X0 << Param: 1; // Первый информационный вход
    X1 << Param: 2; // Второй информационный вход
    A << Param: 3; // Адресный вход
    Y << ((X0, A:Not):And2, (X1,A):And2):Or2; // подсчет
    выходного сигнала
    return << Y; } // вывод результата работы программы
```

Листинг 1. Программа "Мультиплексор 2 к 1" на языке ПИФАГОР

```
module m2l(D0, D1, S, Y);
output Y;
input X0, X1, A;
assign Y = (A)?X1:X0; // подсчет выходного сигнала
endmodule
```

Листинг 2. Программа "Мультиплексор 2 к 1" на языке Verilog

На основании ФПП модели вычислений и поддерживающего ее языка ПИФАГОР авторами предложен метод архитектурно-независимого высокоуровневого синтеза ЦИС [3]. При реализации метода динамическая система типов была заменена на статическую, которая поддерживается в языках описания аппаратуры, исключены

задержанные вычисления и предложен метод их преобразования при переходе на целевую платформу ЦИС. Также в модифицированный ПИФАГОР введен механизм преобразования рекурсии в итеративную схему с последующим переходом к конвейерной схеме [4].

Для обеспечения архитектурной независимости при разработке метода высокоуровневого синтеза используется промежуточное представление программы на языке ПИФАГОР в виде информационного графа (ИГ), задающего информационные связи, и управляющего графа (УГ), использование которого в явном виде позволяет более детально описывать процесс управления вычислениями. Информационный граф формируется в ходе трансляции программы, а УГ может порождаться как динамически, как и статически. Последний способ используется для перехода от программы на ПИФАГОРе к программе описания ЦИС на HDL-языке.

При трансляции построение промежуточного представления ФПП программы осуществляется в два этапа. На первом этапе выполняется трансляция исходного текста в ИГ. Затем по полученному ИГ формируется УГ. Управляющий граф может формироваться по ИГ различными способами, например, в соответствии с принципами управления потоками данных модели вычислений, или сводиться к последовательному обходу ИГ, либо обеспечивать другую стратегию управления вычислениями.

Для перехода на платформу ЦИС выполняется преобразование ИГ к конвейерной схеме. Конвейерная схема вычислений представляет собой ярусно-параллельную форму (ЯПФ) информационного графа, в которой операторы, исполняющиеся независимо друг от друга (фактически параллельно), располагаются на одном уровне (ярусе). В данном случае длина конвейера показывает степень параллелизма решаемой задачи.

Для решения задачи преобразования параллелизма под конкретные ресурсные ограничения выполняются следующие шаги: определение границ изменения параллелизма, реализация алгоритма изменения параллелизма и оценка результата по ресурсным ограничениям.

При переходе от неограниченного параллелизма ФПП модели на целевую платформу решается основная задача — редукция (свертка) параллелизма. Под сверткой параллелизма следует понимать процесс сжатия или сокращения параллелизма. Это является ключевым моментом всех проводимых преобразований в архитектурно-независимом методе синтеза ЦИС. Для реализации механизма

редукции был введен новый метаслой, называемый "HDL-графом". Он представляет собой дополнительный слой для внесения изменений в ФПП модель. По мнению авторов, термин "HDL-граф" наиболее полно соответствует методу синтеза ЦИС. HDL-граф позволяет указывать связи между элементами списков связанных вершин, что в свою очередь позволяет выполнять оптимизационные преобразования с помощью вычисления операции выбора данных из списка. С помощью введенного дополнительного слоя при обработке типов в модели со статической типизацией вводится ограничение на динамическое изменение размерности списков во время вычислений.

Рекурсивные вычисления зачастую становятся препятствием при переносе таких программ на некоторые реальные вычислительные платформы, так как длительные повторяющиеся вычисления порождают переполнение памяти. А это в первую очередь критично для СБИС с ограничениями по объему встроенного ОЗУ, а также для СБИС, где не используется блочная память или применяется память, реализуемая на триггерах. Введение HDL-графа позволило решить эту проблему путем преобразования таких вычислений в итеративные с использованием хвостовой рекурсии и заданием глубины рекурсии на этапе трансляции.

2. Преобразование параллелизма

2.1. Алгоритм редукции параллелизма

Основной особенностью разрабатываемого метода синтеза является переход от распараллеливания задачи под конкретную архитектуру к методу свертки параллелизма под конкретные ресурсные ограничения платформы ЦИС. Такой переход выполняется из исходного максимально-параллельного описания алгоритма. Как показано в работах [2, 14], это обеспечивает переносимость параллельных алгоритмов на различные платформы. Основной особенностью методов редукции параллелизма по сравнению с методами распараллеливания является существенное сокращение числа шагов для получения конечного результата. При редукции (сжатии) максимально-параллельного ИГ число максимально допустимых вариантов преобразований задается на этапе синтеза. В процессе синтеза решаются следующие задачи:

- оценка ресурсов G_k получаемого архитектурного решения;
- оценка производительности получаемого решения;

- расчет коэффициента редукции по каждому классу ресурсов;
- редукция параллелизма схемы для достижения требуемых коэффициентов.

Для оценки ресурсов используется промежуточное представление программы — HDL-граф, в котором уже заданы архитектурно-зависимые данные. HDL-граф представляет собой ациклический граф в ЯПФ, в каждом узле которого заданы типы и разрядности данных [4]. В зависимости от целевой платформы определяются классы ресурсов N_k , по которым будет оцениваться схема.

Например, для платформы ПЛИС к основным классам ресурсов можно отнести:

- число регистров N_r ;
- число логических ячеек N_{lc} ;
- число блоков памяти N_m ;
- число арифметических и иных специализированных вычислительных блоков N_{dsp} .

В данном случае для свертки параллелизма ключевыми моментами являются оценка ресурсов памяти и оценка вычислительных ресурсов.

Ресурс памяти представляет собой множество $N_{mem} = \{N_r, N_m\}$, а вычислительный ресурс N_{comp} является множеством $\{N_{lc}, N_{dsp}\}$.

Внутри подмножества существуют ограничения для взаимозаменяемых ресурсов. Для ресурсов памяти любое хранение данных может быть реализовано на блочной памяти, а на триггерах оно реализуется с ограничениями по объему. В случае вычислительных ресурсов для операции любого типа можно реализовать схему на логических ячейках, при этом тип и набор операций для специализированных блоков ограничен.

Результаты оценки ресурсов используются для дальнейшей редукции параллелизма.

2.2. Оценка ресурсов памяти

Для оценки требуемой памяти суммарный объем ресурсов может быть приведен к общему объему, измеряемому в бит/Кбит.

В качестве примера для оценки ресурса схемы рассмотрим ее HDL-граф. Каждый k -й слой ЯПФ состоит из набора информационных входов B_k и набора операций O_k . Каждый информационный вход вершин HDL-графа после типизации имеет разрядность W_k . Исходя из числа входов и разрядности каждого входа для конкретного слоя графа, можно определить количество ресурсов памяти, требуемых для хранения результата в соответствующем состоянии графа:

$$NR_k = \sum (W_k * B_k).$$

Для расчета ресурса выполняется обход графа и суммирование разрядностей входов и выходов всех вершин.

После первоначальной оценки требуемого ресурса памяти для исходной максимально-параллельной реализации схемы возможны следующие ситуации:

- требуемый ресурс меньше доступного $NR < N_{m/lc}$ ($N_{m/lc}$ — доступный ресурс);
- требуемый ресурс больше доступного $NR \geq N_{m/lc}$.

Первый вариант не требует расчета коэффициента редукции по ресурсам памяти G_m , но при этом при изменении схемы в процессе ее редукции по другим ресурсам оценку и проверку ресурса памяти необходимо проводить вновь.

Во втором случае коэффициент редукции NM рассчитывается согласно формуле

$$G_m = \frac{NR}{N_{m/lc}}.$$

Этот коэффициент используется в алгоритме редукции параллелизма схемы.

Помимо ограничения на объем памяти, необходимо учитывать также и ограничение на производительность памяти (ширина интерфейса данных). Если память построена на регистрах/триггерах, то данные отсутствуют, так как ширина шины данных равна объему данных. Для блочной памяти объем считываемых за такт данных меньше, чем объем хранимых данных. В случае, если ресурс памяти NR , требуемый для хранения результата решения конкретной проблемы, превышает доступный объем регистров N_r , необходимо рассчитывать коэффициент редукции по интерфейсу памяти G_{md} , иными словами, сократить параллелизм на G_{md} .

2.3. Алгоритм определения коэффициента редукции

Для определения минимально-необходимого коэффициента редукции по интерфейсу памяти из всех стадий конвейера выбирается набор стадий с максимальным суммарным реализуемым на регистрах интерфейсом. Для этого из множества стадий конвейера выбирается подмножество стадий, такое что:

$$\max \sum NR \parallel \sum NR_k < (N_{m/lc} - NR_r),$$

где \parallel — логическое сложение.

Исходя из изложенного выше, коэффициент G_{md} определяется как отношение суммарного интерфейса памяти оставшихся стадий к суммарному интерфейсу блочной памяти IM .

Для определения коэффициента редукции по интерфейсу памяти используется следующий алгоритм.

Шаг 1. Выбрать подмножество стадий конвейера, такое что сумма ресурсов NR_k выбранных стадий будет меньше NR_r , и сумма выбранных значений NR_k будет максимальной.

Шаг 2. Рассчитать ресурс памяти, реализуемый на блочной памяти/памяти с последовательным доступом согласно формуле

$$NR_m = NR - \sum NR_k,$$

где k принадлежит подмножеству стадий конвейера, выбранных на шаге 1.

Шаг 3. Рассчитать коэффициент NR_r согласно формуле

$$NR_r = 1 + \text{Int}(NR_m/IM),$$

где Int — округление до целого значения.

Рассмотрим в качестве примера вычисление 4-точечного БПФ (быстрого преобразования Фурье), где дискретное преобразование Фурье вычисляется как [15]

$$a_n = \frac{1}{n} \sum_j^{n-1} S_j * W_n^{-kj} + \frac{1}{n} \sum_j^{n-1} S_j * W_n^{kj},$$

где a_n — коэффициент ряда Фурье; n — число элементов ряда; S_j — измеренные значения сигнала; $W_n^{\pm kj} = \exp\left(-j \frac{2\pi}{n} nk\right)$ — фазовые множители (действительные (Re) и мнимые (Im) соответственно).

При этом тип входных данных — 16-битный знаковый.

Информационный граф после преобразования в HDL-граф и приведения в ЯПФ показан на рис. 2. На нем показано, что вычисление БПФ проводится в 4 стадии (Stages 1, 2, 3, 4). Все вычисления в рамках одной стадии выполняются параллельно. Количество ресурсов, необходимых для выполнения конкретных вычислений, приведено на каждой стадии для решения определенной подзадачи и обозначено как int16, int32 и т. п.

Значение ресурсов NR_k для каждой стадии конвейера рассчитывается следующим образом:

$$NR_1 = 10 * 2 * 16 = 320;$$

$$NR_2 = 16 * 4 + 8 * 32 = 320;$$

$$NR_3 = 16 * 4 + 4 * 32 = 196;$$

$$NR_4 = 33 * 4 + 4 * 34 = 268.$$

Общее значение ресурса $NR = 1104$ бита.

Рассмотрим две архитектуры — A_1 и A_2 . Значение $N_{m/lc}$ для обеих архитектур равно 1536 бит. В архитектуре A_1 весь ресурс памяти находится в регистрах. Для такой архитектуры схема БПФ 4 в максимально-параллельной форме реализуется без изменений, как показано на рис. 2.

В архитектуре A_2 ресурс регистров составляет 512 бит, а 1024 бит представлены в виде блочной памяти с интерфейсом данных 36 бит. Значение суммарного интерфейса блочной памяти IR составит 36 бит.

В соответствии с описанным выше алгоритмом выберем подмножество стадий, реализуемых на регистрах и имеющих максимальный интерфейс. В данном примере это может быть либо стадия 1 (Stage 1), либо стадия 2 (Stage 2).

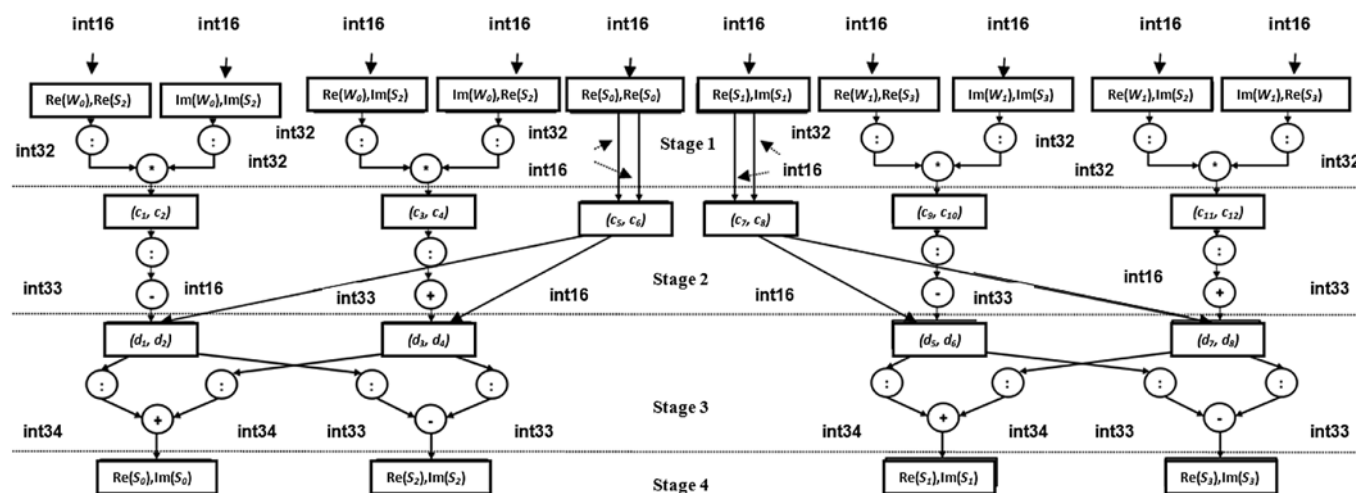


Рис. 2. HDL-граф максимально-параллельной формы для БПФ

Значение NR_m в таком случае будет рассчитываться следующим образом:

$$NR_m = NT = NR_1104 - 320 = 784 \text{ бита.}$$

Значение коэффициента редукции по интерфейсу памяти составит

$$G_{md} = \frac{784}{36} = 22.$$

В этом случае период подачи данных становится равным G_{md} и стадии конвейера 2, 3, 4 или 1, 3, 4 реализуются последовательно, так как результат записывается в один блок памяти. HDL-граф схемы после редукции на G_{md} приведен на рис. 3.

Запись значений $c_1 - c_{12}$, вычисление значений $d_1 - d_8$, $S_0 - S_3$ проводится в данной схеме последовательно. Так как стадии 2, 3 исходной схемы после редукции требуют 22 такта на выполнение, стадия 1 также может быть увеличена до 22 тактов (clock cycles) без влияния на общую производительность системы, что приведет к пропорциональному уменьшению вычислительного ресурса стадии 1.

2.4. Оценка вычислительных ресурсов

Вычислительный ресурс обозначим как N_{comp} — множество $\{N_{lc}, N_{dsp}\}$. В данном случае для операции любого типа можно реализовать схему на логических ячейках, при этом тип и набор операций для специализированных блоков ограничен.

Общее число слоев (стадий конвейера) равно M . На каждом j -м слое графа реализуется определенное множество операций O^j , где $j = 1, \dots, k$.

Для всего HDL-графа число каждой операции

$$F_k = \sum_{j=0}^M O_k^j.$$

Обозначим общее число различных типов операций через L , $i = 0, \dots, L$. Под типом операции понимается тип арифметической/логической и т. д. операции вместе с указанием разрядностей данных, например, сложение 16-битных данных, сравнение 20-битных данных и т. д.

После расчета общего числа операций каждого типа, исходя из доступного ресурса конкретной архитектуры, необходимо провести оценку степени параллелизма, с которой возможно реализовать схему.

Пусть количество ресурса для каждого конкретного типа операции известно. Обозначим Y — тип ресурса (логические ячейки, специализированные арифметические блоки DSP и т. д.), $V(Y)_k$ — количество ресурса типа Y , необходимое для реализации операции типа k . Тогда суммарный ресурс типа T для всех операций в HDL графе:

$$V(Y) = \sum_{k=0}^L V(Y)_k * F_k.$$

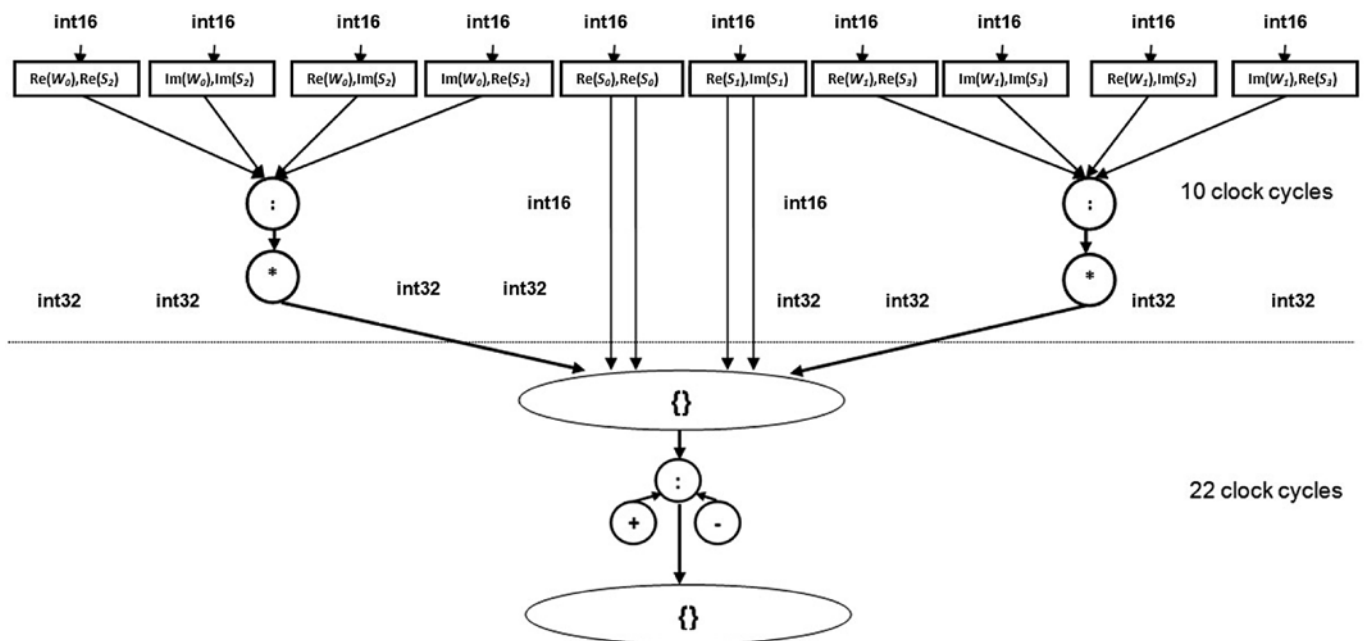


Рис. 3. HDL-граф после редукции

По каждому классу вычислительных ресурсов можно рассчитать коэффициент редукции G_y как отношение суммарного требуемого ресурса к ресурсу целевой архитектуры, округленное до ближайшего большего целого:

$$G_y = \text{Int}(V(Y)/N_y).$$

Итоговый коэффициент редукции по вычислительным ресурсам определяется как максимальный среди всех G_y :

$$G_{calc} = \max\{G_y\},$$

Рассмотрим данный алгоритм на примере графа, представленном на рис. 2. Общее число стадий конвейера данного графа 4. Число типов операций L составит 5: 16-битное умножение ($i = 0$), сложение и вычитание 33 и 16 бит ($i = 1, 2$), сложение 34-битных данных и вычитание 33-битных данных ($i = 3, 4$). Число операций по слоям графа составит

$$\begin{aligned} O_1 &= \{8, 0, 0, 0, 0\}; \\ O_2 &= \{0, 2, 2, 0, 0\}; \\ O_3 &= \{0, 0, 0, 4, 4\}; \\ O_4 &= \{0, 0, 0, 0, 0\}. \end{aligned}$$

Количество каждой операции k -го типа:

$$F_0 = 8, F_1 = 2, F_2 = 2, F_3 = 4, F_4 = 4.$$

Пусть имеется архитектура с двумя типами ресурсов для реализации вычислений: логические ячейки ($Y = 0$) и секции DSP ($Y = 1$). Значения ресурса для каждого типа операции возьмем следующие:

для логических ячеек получаем:

$$V(0)_0 = 0, V(0)_1 = 5,$$

$$V(0)_2 = 5, V(0)_3 = 10, V(0)_4 = 10;$$

для секций DSP получаем:

$$V(1)_0 = 1, V(1)_1 = 0,$$

$$V(1)_2 = 0, V(1)_3 = 0, V(1)_4 = 0.$$

Суммарный ресурс по каждому типу по всем операциям в графе составит

$$\begin{aligned} V(0) &= \sum_{k=0}^L V(Y)_k * F_k = 0 * 8 + 5 * 2 + 5 * 2 + \\ &+ 4 * 10 + 4 * 10 = 100, \end{aligned}$$

$$\begin{aligned} V(1) &= \sum_{k=0}^L V(Y)_k * F_k = 1 * 8 + 0 * 2 + \\ &+ 0 * 2 + 0 * 10 + 0 * 10 = 8. \end{aligned}$$

Рассмотрим архитектуру, где число DSP $N_1 = 2$, число логических ячеек $N_0 = 400$. Тогда коэффициент редукции по каждому типу ресурсов составит

$$G_0 = \text{Int}\left(\frac{V(0)}{G_0}\right) = \frac{100}{400} = 0,25,$$

$$G_1 = \text{Int}\left(\frac{V(1)}{G_1}\right) = \frac{8}{2} = 4.$$

Значение коэффициента редукции для вычислительных ресурсов G_{calc} равно максимальному из всех G_y , следовательно коэффициент G_{calc} равен

$$G_{calc} = \max\{0,25, 4\} = 4.$$

После редукции каждой стадии с коэффициентом 4 задержка каждой стадии вырастет до 4 тактов, ресурс при этом снизится также в 4 раза. Результирующий граф показан на рис. 4.

Отметим, что при изменении вычислительного ресурса может измениться ресурс памяти.

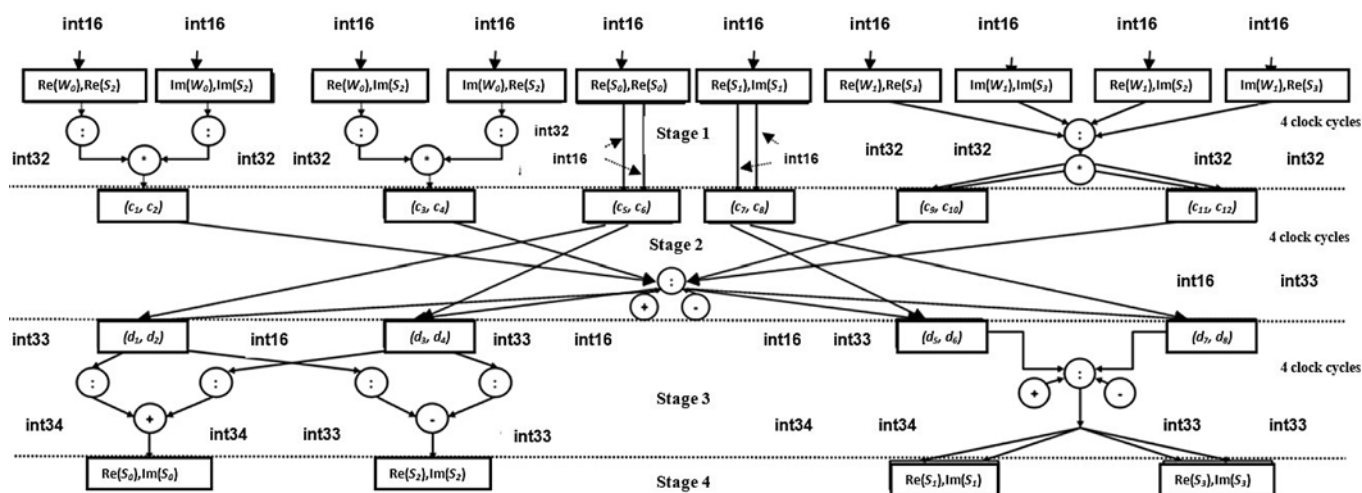


Рис. 4. HDL-граф после редукции на G_{calc}

2.5. Обобщенный алгоритм преобразования параллелизма

Пусть отношение ресурса, требуемого для реализации схемы в исходном максимально-параллельном виде к доступному ресурсу, обозначается как R . Предполагается, что доступный ресурс — это наименьший из ресурсов. При решении задачи трансформации параллелизма в общем виде в зависимости от доступного ресурса целевой платформы можно выделить три варианта преобразования исходной максимально параллельной схемы.

Первый вариант: если $R > 1$, то необходима редукция параллелизма.

Здесь возможно увеличение производительности путем размещения нескольких схем параллельно:

$$S = \text{Int}(1/R).$$

Рассмотрим алгоритм редукции с использованием коэффициентов редукции, описанных в подразд. 2.2/2.4.

Алгоритм состоит из следующих шагов.

Шаг 1. Рассчитываем коэффициенты редукции N_{calc} и G_m .

Шаг 2. Выбираем максимальный коэффициент $N_{max} = \max(G_m, N_{calc})$.

Шаг 3. Редуцируем параллелизм схемы на N_{max} .

Шаг 4. Для измененной схемы пересчитываем коэффициенты G_m и G_{calc} , если они меньше 1 — завершение алгоритма.

Шаг 5. Если какие-либо операции возможно реализовать с использованием другого типа ресурсов — изменяем данные операции на другой тип ресурсов без изменения коэффициента R и пересчитываем коэффициенты G_m и G_{calc} .

Шаг 6. Если какой-либо из коэффициентов больше 1: $G_{max} = 1 + G_{max}$ и возвращаемся на шаг 3.

Последовательное увеличение коэффициента редукции позволяет подобрать минимально возможный коэффициент для удовлетворения требований по ресурсам и достижения при этом максимальной производительности (минимально возможной редукции по производительности относительно исходного максимально параллельного варианта).

Второй вариант преобразования исходной максимально параллельной схемы:

при $R < 1/2$ возможно увеличение числа схем.

Третий вариант преобразования исходной максимально параллельной схемы:

$1/2 < R < 1$, ресурса достаточно для размещения одного максимально параллельного варианта схемы.

Во втором и третьем случаях никаких преобразований максимально параллельной схемы не требуется.

3. Практические результаты

В рамках проводимых исследований авторами реализован набор программных инструментальных средств, позволяющих:

- выполнять преобразование исходного кода на ФПП языке в промежуточное представление в виде информационного и управляющего графов;
- осуществлять оптимизационные преобразования, что позволяет повысить эффективность ФПП программ;
- проводить отладку и анализ ФПП кода во время выполнения, обеспечивая поиск ошибок и трассировку;
- осуществлять трансляцию промежуточного представления ФПП программ в описание СБИС на HDL-языках [15].

На рис. 5 представлена архитектура разработанных инструментальных средств поддержки

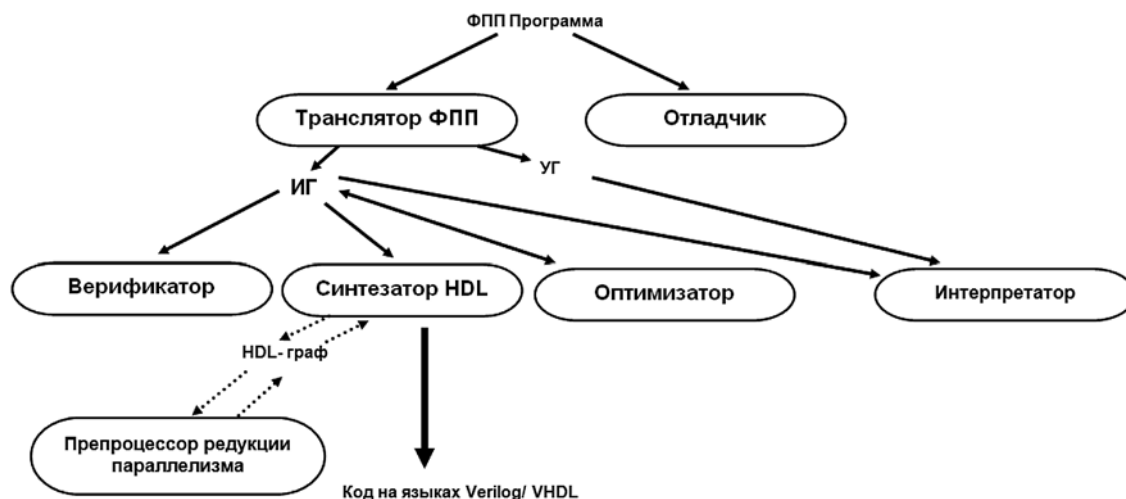


Рис. 5. Архитектура инструментальных средств поддержки проектирования

проектирования на основе предложенного метода высокоуровневого синтеза.

Интерпретатор позволяет исполнить текст программы на ФПП языке. Входными данными для интерпретатора являются информационный и управляющий графы, а также аргумент функции верхнего уровня. Аргумент представляется в формате описания ИГ, для этого он обрабатывается транслятором.

Оптимизатор также использует входное промежуточное представление, осуществляет оптимизирующие преобразования над ИГ, результат сохраняется в виде промежуточного представления ИГ.

К оптимизирующим преобразованиям, осуществляемым на данном этапе, относятся:

- удаление неиспользуемого кода;
- оптимизация повторяющихся вычислений;
- непосредственная подстановка функций;
- удаление повторяющегося кода;
- оптимизация на основе эквивалентных преобразований алгебры ФПП модели.

Транслятор позволяет проводить проверку синтаксиса программ, разработанных на языке ФПП программирования ПИФАГОР, и преобразование программы в ее промежуточное представление в виде информационного и управляющего графов [16]. Он реализует механизм трансформации с функционально-поточкового описания в описания на уровне комбинационных логических и автоматных схем. В состав транслятора входит отладчик, генератор ИГ и генератор УГ. Результатом функционирования транслятора является набор отлаженных функций, реализованных на языках Verilog/VHDL.

Пакет разработанных программных средств также включает препроцессор редукции параллелизма и препроцессор оценки ресурсов. Препроцессор редукции параллелизма позволяет проводить автоматические преобразования параллелизма программ, предназначенных для трансляции в HDL-язык с учетом ресурсных ограничений. Препроцессор использует в качестве входных данных промежуточное представление алгоритма СБИС на ФПП языке в виде типизированного ИГ (HDL-графа) и ресурсные ограничения целевой платформы, полученные при помощи препроцессора ресурсных ограничений. Результатом функционирования препроцессора редукции параллелизма является ИГ алгоритма, преобразованный с учетом ресурсных ограничений. Полученное представление используется синтезатором схем для получения описания схемы СБИС на HDL-языках.

Разработаны трансляторы для языков Verilog и VHDL [17]. Программа реализует проверку ис-

ходного описания на пригодность к синтезу, сборку исходного описания из множества функций, назначение типов данных в исходном описании и синтез выходного описания схемы на языках Verilog/VHDL.

Набор разработанных программных инструментов функционирует в составе интегрированной среды разработки и позволяет сформировать набор отлаженных функций для их реализации в виде СБИС. Оболочка предоставляет пользователю информационные ресурсы для организации всего процесса высокоуровневого синтеза СБИС на основе ФПП подхода.

Посредством разработанных инструментальных средств на практике получен ряд научно-технических решений, перечисленных далее.

- Комплект сложно-функциональных блоков однокристалльного устройства управления бортовой сети космического аппарата [18]. Программа, входящая в состав блоков, реализует бортовой коммуникационный модуль стандарта Space Wire, который представляет собой законченный приемопередающий узел бортовой сети космического аппарата и выполняет функции приемника, передатчика и контроллера канала между другими узлами сети.

- Сложно-функциональные блоки из состава СБИС спутникового модема ЯР-1040 [19, 20].

- СБИС блока ЦОС на базе БМК K5540ТН014А из состава навигационного прибора [21]. Программа, входящая в состав, реализуется на ПЛИС Xilinx Spartan-4 и предназначена для использования в составе приемника навигационных сигналов ГЛОНАСС/GPS и осуществляет первичную обработку сигналов на ПЛИС либо заказной микросхеме.

Заключение

Проведенный обзор современных языков и способов проектирования логических схем позволил обосновать выбор модели функционально-поточковых параллельных вычислений и языка ФПП программирования ПИФАГОР для разработки архитектурно-независимого метода синтеза интегральных схем.

В процессе разработки архитектурно-независимого высокоуровневого метода синтеза ЦИС, основанного на модифицированной модели ФПП, был предложен метод преобразования параллелизма, заключающийся в сокращении максимального параллелизма решаемой задачи при переходе к конкретной целевой архитектуре. Этот подход обеспечивает переносимость параллельных архитектур на различные платформы.

Алгоритм преобразования параллелизма включает оценку ресурсов, вычисление коэффициента редукции параллелизма по каждому классу ресурсов и редукцию параллелизма схемы для достижения требуемых характеристик конкретной целевой платформы.

Представленные методы редукции позволяют реализовать изменение параллелизма исходного описания алгоритма и обеспечить реализацию механизма переноса на различные архитектуры с различными ресурсными ограничениями.

В отличие от методов распараллеливания, предлагаемый в работе метод снижает сложность процесса переноса за счет уменьшения перебора количества вариантов реализации, получаемых в процессе синтеза под новые ресурсные ограничения. Вместе с тем методы оценки ресурса на высокоуровневом этапе требуют учета накладных расходов при изменении параллелизма к более последовательным схемам. Для данного случая необходимо увеличение точности оценки. Для этого могут использоваться нейронные сети и методы машинного обучения, которые на основе параметров оценки схемы на высокоуровневом этапе могут с необходимой точностью предсказать значения параметров схемы после реализации на низком уровне.

Методы прямого подсчета ресурса результата реализации схемы осложнены по причине множества преобразований, которые происходят при реализации схемы на этапах синтеза и имплементации на низкоуровневом этапе предлагаемого метода. Реализация точных методов оценки ресурса позволит дополнительно уменьшить количество вариантов схемы, рассматриваемых в процессе синтеза под ресурсные ограничения.

На основе предложенных методов редукции параллелизма и оценки ресурсов реализован ряд инструментальных средств, таких как транслятор архитектурно независимого описания автоматных и комбинационных схем, препроцессоры оценки ресурсов и редукции параллелизма. Эти препроцессоры позволяют рассчитывать и количественно оценивать требуемый ресурс схемы по выбранным типам элементарных блоков (ячеек, памяти, триггеров, LUT-таблиц и т. д.) на поддерживаемых целевых платформах.

Список литературы

1. Левин И. И., Гудков В. А. Расширение языка высокого уровня COLAMO для программирования реконфигурируемых вычислительных систем на уровне логических ячеек ПЛИС // Вестник компьютерных и информационных технологий. 2010. № 12. С. 10–17.
2. Dongarra J., Bosilca G., Bouteiller A. et al. PaRSEC: A programming paradigm exploiting heterogeneity for enhancing

scalability// IEEE Computing in Science and Engineering. 2013. Vol. 15, No. 6. P. 36–45.

3. Легалов А. И. Функциональный язык для создания архитектурно-независимых параллельных программ // Вычислительные технологии. 2005. Том 10, № 1. С. 71–89.

4. Nepomnyashchy O. V., Legalov A. I., Tyapkin V. et al. Methods and algorithms for a high-level synthesis of the very-large-scale integration// WSEAS Transactions on Computers. 2016. No. 15. P. 239–247.

5. IEEE Std 1800-2012: IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language, 2013.

6. Алексин В. А. SystemC. Моделирование электронных систем. М.: Горячая линия — Телеком, 2018. 320 с.

7. Vivado Design Suite User Guide. High-Level Synthesis. UG902—Xilinx—2015. URL: <https://www.yumpu.com/en/document/view/7034022/xilinx-vivado-design-suite-user-guide-high-level-synthesis-ug902>

8. Handel-C Language Reference Manual. Celoxica Limited, 2005. 350 p.

9. Sérot J., Michaelson G. Compiling Hume down to gates// Draft Proceedings of 11th International Symposium on Trends in Functional Programming. Madrid, 2011. P. 191–226.

10. O'Donnell J., Rüniger G. Derivation of a logarithmic time carry lookahead addition circuit// Journal of Functional Programming. 2004. Vol. 14, No. 6. P. 697–713.

11. Axelsson E., Ciaessen K., Sheeran M. Wired: wire-aware circuit design// Proceedings of the Conference on Correct Hardware Design and Verification Methods (CHARME '05). 2005. Vol. 3725. P. 5–19.

12. Bachrach J., Vo H., Richards B. et al. Chisel: Constructing hardware in a Scala embedded language// DAC Design Automation Conference 2012. 2012. P. 1212–1221.

13. Gill A., Bull T., Kimmell G. et al. Introducing Kansas Lava// Implementation and Application of Functional Languages, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Sep. 2009. P. 18–35.

14. Dongarra J., Danalis A., Bosilca G. et al. PTG: An Abstraction for Unhindered Parallelism// Proceedings of the Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing, 2014. P. 21–30.

15. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления сверток. М.: Радио и связь, 1985. 248 с.

16. Nepomnyashchy O. V., Ryzhenko I. N., Shaydurov V. V. et al. The VLSI High-Level Synthesis for Building Onboard Spacecraft Control Systems// Proceedings of the Scientific-Practical Conference "Research and Development — 2016". Springer, Cham. 2017. P. 229–238.

17. Непомнящий О. В., Романова Д. С., Рыженко И. Н., Легалов А. И. Транслятор архитектурно-независимого описания автоматных и комбинационных схем. Свидетельство о государственной регистрации ПО для ЭВМ № 2021610682, 01.02.2021.

18. Комаров А. А., Рыженко И. Н., Непомнящий О. В. Программа синтеза описания схем на языках описания аппаратуры HDL с языка функционально-параллельного программирования "ПИФАГОР". Свидетельство о государственной регистрации ПО для ЭВМ № 2015619175, 26.08.2015.

19. Непомнящий О. В., Комаров А. А., Рыженко И. Н. и др. Программа драйвера бортовой сети космического аппарата. Свидетельство о государственной регистрации ПО для ЭВМ № 2015616896, 25.06.2015.

20. Рыженко И. Н., Комаров А. А., Андреев А. С. Программное обеспечение спутникового модема "ЯР-1040". Свидетельство о государственной регистрации ПО для ЭВМ № 2015614726, 27.04.2015.

21. Непомнящий О. В., Комаров А. А., Рыженко И. Н. Сложно-функциональный блок понижающего сумматора-ограничителя. Свидетельство о государственной регистрации ПО для ЭВМ № 2016619714, 26.08.2016.

Parallelism Reduction Methods in the High-Level VLSI Synthesis Implementation

D. S. Romanova, daryaooo@mail.ru, Siberian Federal University, Krasnoyarsk, 660041, Russian Federation, Krasnoyarsk State Agrarian University, Krasnoyarsk, 660049, Russian Federation,
O. V. Nepomnyashchiy, Siberian Federal University, Krasnoyarsk, 660041, Russian Federation,
A. I. Legalov, legalov@mail.ru, National Research University — Higher School of Economics, Moscow, 101000, Russian Federation,
I. N. Ryzhenko, rodgi.krs@gmail.com, **N. Y. Sirotinina**, nsirotinina@sfu-kras.ru, Siberian Federal University, Krasnoyarsk, 660041, Russian Federation

Corresponding author:

Romanova Darya S., Postgraduate Student, Krasnoyarsk State Agrarian University, Krasnoyarsk, 660049, Russian Federation
E-mail: daryaooo@mail.ru

*Received on July 14, 2021
Accepted on April 15, 2022*

The problems and solutions in the field of ensuring architectural independence and implementation of digital integrated circuits end-to-end design processes are considered. The paper focuses on the need to find a solution to the problem of program portability during the development of integrated circuits. A review of the main software and tools used to design digital circuits (Verilog, System-C, Handel-C, Lava, Hydra, Wired, COLAMO, Chisel and etc.) is presented. The method and language of parallel programming for functional flow synthesis of design solutions PIFAGOR is presented. The example of the source and generated code in the PIFAGOR and Verilog languages is given. During the method implementation, the tasks of reducing parallelism and estimating the occupied resources were highlighted. The main feature of the developed method is the introduction of the additional layer (HDL graph) into the synthesis process. Algorithms for the parallelism reduction have been developed. This method is demonstrated on the example of parallelism reduction while going to the FPGA platform solving the problem of calculating a 4-point FFT (Fast Fourier Transform). As part of the solution of this task, an assessment of memory resources and an assessment of computing resources were carried out. The results of software tools development for design support including the parallelism reduction preprocessor and resource estimation preprocessor and practical VLSI projects are presented.

Keywords: *integrated circuit, parallel computing model, algorithm, program, high-level synthesis, functional-stream language*

For citation:

Romanova D. S., Nepomnyashchiy O. V., Legalov A. I., Ryzhenko I. N., Sirotinina N. Y. Parallelism Reduction Methods in the High-Level VLSI Synthesis Implementation, *Programmnaya Ingeneria*, 2022, vol. 13, no. 6, pp. 259–271.

DOI: 10.17587/prin.259-271

References

1. **Levin I. I., Gudkov V. A.** Extension of the high-level language COLAMO for programming reconfigurable computing systems at the level of logical FPGA cells., *Vestnik komp'yuternykh i informacionnykh tehnologii*, 2010, no. 12, pp. 10–17 (in Russian).
2. **Dongarra J., Bosilca G., Bouteiller A., Danalis A., Favre M., Herault T.** PaRSEC: A programming paradigm exploiting heterogeneity for enhancing scalability. *IEEE Computing in Science and Engineering*, 2013, vol. 15, no. 6, pp. 36–45.
3. **Legalov A. I.** A functional language for creating architecture-independent parallel programs, *Vychislitel'nye tehnologii*, 2005, vol. 10, no. 1, pp. 71–89 (in Russian).
4. **Nepomnyashchy O. V., Legalov A. I., Tyapkin V., Ryzhenko I. N., Shaydurov V. V.** Methods and algorithms for a high-level synthesis of the very-large-scale integration, *WSEAS Transactions on Computers*, 2016, no. 15, pp. 239–247.
5. **IEEE Std 1800-2012:** IEEE Standard for SystemVerilog- Unified Hardware Design, Specification, and Verification Language, 2013.
6. **Alekhin V. A.** *SystemC. Simulation of electronic systems*. Moscow, Goryachaya liniya — Telecom, 2018, 320 p. (in Russian).
7. **Vivado Design Suite User Guide.** High-Level Synthesis. UG902—Xilinx—2015, available at: <https://www.yumpu.com/en/document/view/7034022/xilinx-vivado-design-suite-user-guide-high-level-synthesis-ug902>

-
-
8. **Handel-C Language Reference Manual** / Celoxica Limited, 2005.
 9. **Sérot J., Michaelson G.** Compiling Hume down to gates, *Draft Proceedings of 11th International Symposium on Trends in Functional Programming*, Madrid, 2011. pp. 191–226.
 10. **O'Donnell J., Rüniger G.** Derivation of a logarithmic time carry lookahead addition circuit. *Journal of Functional Programming*, 2004, vol. 14, no. 6, pp. 697–713.
 11. **Axelsson E., Ciaessen K., Sheeran M.** Wired: wire-aware circuit design, *Proceedings of the Conference on Correct Hardware Design and Verification Methods (CHARME '05)*, 2005, vol. 3725, pp. 5–19.
 12. **Bachrach J., Vo H., Richards B.** et al. Chisel: Constructing hardware in a Scala embedded language, *DAC Design Automation Conference 2012*, 2012, pp. 1212–1221.
 13. **Gill A., Bull T., Kimmell G.** et al. Introducing Kansas Lava, *Implementation and Application of Functional Languages, ser. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, Sep. 2009, pp. 18–35.
 14. **Dongarra J., Danalis A., Bosilca G.** et al. PTG: An Abstraction for Unhindered Parallelism, *Proceedings of the Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing*, 2014, pp. 21–30.
 15. **Nussbaumer G.** Fast Fourier Transform and Convolution Computing Algorithms. Moscow, Radio i svyaz', 1985, 248 p. (in Russian).
 16. **Nepomnyashchiy O. V., Ryzhenko I. N., Shaydurov V. V., Sirotinina N. Y., Postnikov A. I.** The VLSI High-Level Synthesis for Building Onboard Spacecraft Control Systems, *Proceedings of the Scientific-Practical Conference "Research and Development — 2016"*. Springer, Cham, 2017, pp. 229–238.
 17. **Nepomnyashchy O. V., Ryzhenko I. N., Romanova D. S., Legalov A. I.** Translator of architecture-independent description of automata and combinational circuits, Certificate of state registration of software for computers No. 2021610682, 02/01/2021 (in Russian).
 18. **Komarov A. A., Ryzhenko I. N., Nepomnyashchy O. V.** Program for synthesizing circuit descriptions in HDL hardware description languages from the Pifagor functional-parallel programming language, Certificate of state registration of software for computers No. 2015619175, 08/26/2015 (in Russian).
 19. **Nepomnyashchy O. V., Komarov A. A., Ryzhenko I. N.** Program for the driver of the onboard network of the spacecraft, Certificate of state registration of software for computers No. 2015616896, 06/26/2015 (in Russian).
 20. **Ryzhenko I. N., Komarov A. A., Andreev A. S.** Software of the satellite modem "YAR-1040", Certificate of state registration of software for computers No. 2015614726, 04/27/2015 (in Russian).
 21. **Nepomnyashchy O. V., Komarov A. A., Ryzhenko I. N.** Complex-functional block of the lowering adder-limiter, Certificate of state registration of software for computers No. 2016619714, 08/26/2016 (in Russian).

***Продолжается подписка на журнал
"Программная инженерия" на второе полугодие 2022 г.***

Оформить подписку можно через подписные агентства
или непосредственно в редакции журнала.
Подписной индекс по Объединенному каталогу
"Пресса России" — 22765

Сообщаем, что с 2020 г. возможна подписка
на электронную версию нашего журнала:

ООО "ИВИС": тел. (495) 777-65-57, 777-65-58; e-mail: sales@ivis.ru,
ООО "УП Урал-Пресс". Для оформления подписки (индекс 013312)
следует обратиться в филиал по месту жительства — <http://ural-press.ru>

Адрес редакции: 107076, Москва, Матросская Тишина, д. 23, стр. 2, оф. 45,
Издательство "Новые технологии",
редакция журнала "Программная инженерия"

Тел.: (499) 270-16-52. E-mail: prin@novtex.ru

А. В. Третьяк, аспирант, alextretyak2@gmail.com,
Е. А. Верещагина, канд. техн. наук, доц., everesh@mail.ru, Дальневосточный федеральный университет, г. Владивосток,
П. В. Струк, вед. специалист, pavelstruk@yandex.ru, "Приморский океанариум" — филиал ННЦМБ ДВО РАН, г. Владивосток

Разработка системы символьных макрокоманд (симкоманд) процессора

Традиционной формой записи машинного кода является язык ассемблера. Его команды прямо соответствуют отдельным командам машины или их последовательностям. Синтаксис программы на языке ассемблера определяется главным образом системой команд конкретного процессора и системой директив конкретного транслятора. Представленная в статье система символьных макрокоманд процессора является, по мнению авторов, более наглядной формой представления машинного кода, что способствует повышению продуктивности при работе с ней программистом, а также минимизации ошибок.

Ключевые слова: система команд процессора, язык ассемблера, макрокоманда

Введение

Традиционной формой записи машинного кода является язык ассемблера. Язык ассемблера — машинно-ориентированный язык программирования низкого уровня. Он представляет собой систему обозначений, используемую для представления в удобно читаемой форме программ, записанных в машинном коде.

Команды языка ассемблера прямо соответствуют отдельным командам машины или их последовательностям [1]. Синтаксис программы на языке ассемблера определяется главным образом системой команд конкретного процессора и системой директив конкретного транслятора. Система команд процессора — это набор допустимых для данного процессора управляющих кодов и способов адресации данных [2]. Система команд процессора включает в себя все команды, которые могут быть записаны с использованием различных видов синтаксиса языка ассемблера.

Для некоторых платформ может существовать несколько видов синтаксиса языка ассемблера, не совместимых между собой. Например, наиболее популярные синтаксисы языков ассемблера для Intel-совместимых процессоров — Intel-синтаксис и AT&T-синтаксис. Однако все существующие виды синтаксиса языка ассемблера ориентированы, в первую очередь, на "легкую читаемость" со стороны ЭВМ, а не программиста.

Объединение отдельных команд для выполнения определенной задачи в языке ассемблера называется макрокомандой процессора.

Макрокоманда — это символьное имя, заменяемое при обработке препроцессором на последовательность программных инструкций [3]. Другими словами, макрокоманда — это мнемоническая команда, которая разворачивается более чем в одну машинную инструкцию [4].

Команды процессора в языке ассемблера обозначаются латинскими буквами, например, команда `mov` присваивает значение заданному регистру.

Для записи команд процессора авторами предлагается ввести использование вместо латинских букв символьное обозначение, например, команде ассемблера `mov eax, 2` соответствует символьная команда `eax = 2`. Для обозначения данного способа записи команд предлагается ввести понятие "симкоманда".

Симкоманда — это символьная макрокоманда процессора с Си-подобным синтаксисом. Симкоманды предназначены для наиболее четкого выражения намерения программиста, скрывая незначительные особенности архитектуры процессора. Так, например, очевидно, что при использовании ассемблерной команды `xor eax, eax` намерением программиста было не выполнить операцию "исключающего или" над регистром `eax`, а просто установить его значение равным нулю. На языке симкоманд это действие выражается понятной

симкомандой `eax = 0`, в то время как на языке ассемблера x86 понятная команда `mov eax, 0` не используется в силу того, что есть более эффективная команда, соответствующая этому же действию: `xor eax, eax`. Аналогично и в случае с командой `test eax, eax`, которая равнозначна команде `cmp eax, 0`, однако является более эффективной.

Таким образом, целью введения нового языка симкоманд является повышение эффективности как с аппаратной точки зрения (более эффективное исполнение на процессоре, реализующем систему симкоманд), так и со стороны программиста (минимизация ошибок, более наглядная форма представления машинного кода).

Задача симкоманд — обеспечить понятное отображение на машинный код, причем по возможности самым оптимальным образом. Так, например, симкоманда `edx = eax + ebx*2` транслируется в `lea edx, [eax + ebx*2]` и является допустимой, а симкоманда `edx = eax + ebx*33` уже не является допустимой, так как не может быть транслирована в одну инструкцию. Хотя симкоманды вызова функций являются макрокомандами (т. е. составными, состоящими из нескольких инструкций, командами), они являются допустимыми на основании того, что порождаемые ими ассемблерные инструкции являются оптимальными и не нуждаются в раскрытии/отображении их программисту.

Цель исследования, результаты которого представлены в статье, — разработка системы символьных макрокоманд (симкоманд) процессора как более наглядной формы представления машинного кода для повышения эффективности при работе с ней программиста, включая минимизацию ошибок.

Симкоманды арифметических и логических операций

В табл. 1 представлены наиболее часто используемые арифметические [5] и логические [6]

Команда ассемблера x86	Симкоманда
<code>add eax, ebx</code>	<code>eax += ebx</code>
<code>sub eax, ebx</code>	<code>eax -= ebx</code>
<code>imul ebx</code>	<code>eax *= ebx</code>
<code>mul ebx</code>	<code>eax **= ebx</code>
<code>imul ecx, edx</code>	<code>ecx *= edx</code>
<code>imul ecx, edx, 10</code>	<code>ecx = edx * 10</code>
<code>idiv ebx</code>	<code>eax /= ebx</code>
<code>div ebx</code>	<code>eax //= ebx</code>
<code>neg eax</code>	<code>eax = -eax</code>
<code>inc eax</code>	<code>eax++</code>
<code>dec eax</code>	<code>eax--</code>
<code>and eax, ebx</code>	<code>eax &= ebx</code>
<code>or eax, ebx</code>	<code>eax = ebx</code>
<code>xor eax, ebx</code>	<code>eax (+) = ebx</code>
<code>xor eax, eax</code>	<code>eax = 0</code>
<code>not eax</code>	<code>eax = (-)eax</code>
<code>sal eax, cl</code>	<code>eax <<= cl</code>
<code>sar eax, cl</code>	<code>eax >>= cl</code>
<code>shr eax, cl</code>	<code>eax >>>= cl</code>
<code>rol eax, cl</code>	<code>eax (<<) = cl</code>
<code>ror eax, cl</code>	<code>eax (>>) = cl</code>

команды процессора (на примере ассемблера для архитектуры x86) и соответствующие им разработанные авторами симкоманды.

Симкоманды перехода

В качестве примера использования симкоманд перехода в табл. 2 представлена часть функции для определения того факта, является ли заданная строка палиндромом, т. е. читается ли она

Таблица 2

Ассемблер x86-64 [7]	Симкоманды x86-64
<pre> palindrome_start: cmp rcx, 0 jl palindrome_end mov rbx, rdx sub rbx, rcx sub rbx, 1 mov bl, byte [rdi + rbx] cmp byte [rdi + rcx], bl jne palindrome_failed dec rcx jmp palindrome_start palindrome_end: </pre>	<pre> palindrome_start: rcx < 0: palindrome_end rbx = rdx rbx -= rcx rbx -= 1 bl = [rdi + rbx] [rdi + rcx] != bl: palindrome_failed rcx-- :palindrome_start palindrome_end: </pre>

одинаково в обоих направлениях — слева направо и справа налево.

Можно заметить, что такая запись делает инструкции безусловного перехода симметричными записи, используемой для обозначения меток, так как запись `метка:` обозначает объявление метки, а `а :метка` — переход на эту метку.

Такая запись [`<условие_или_команда_меняющая_флаги>`] `:метка` имеет перечисленные далее преимущества.

1. Четко обозначает связь команды, меняющей флаги, с командой перехода. Так, новичкам в ассемблере может быть не очевидно, что команда `jnz label` следующая за командой `dec ecx` опирается на тот факт, что команда `dec ecx` устанавливает или сбрасывает флаг ZF (флаг нуля (zero flag) устанавливается, если при выполнении операции получается число равное нулю). Запись `--ecx != 0 :label` при этом показывает, что операция `--ecx (dec ecx)` непосредственно связана с командой перехода на метку `label`. Если посмотреть с позиции машинной логики, то можно отметить, что такая запись показывает, что флаг ZF, изменяемый командой `--ecx`, реально используется. Это означает, что в этом отношении симкоманды выражают намерение более четко как для программиста, так и для машины. Этот момент может быть использован для разработки новых процессорных архитектур, развивая направление, по которому пошли в ARM: "Adding the `s` suffix to `sub` causes it to update the flags itself, based on the result of the operation." [8]. Значит вместо `subs` в `subs r4, r4, #1` в гипотетической новой архитектуре можно было бы написать `subz`, так как реально используется только флаг ZF. Другие флаги при этом (например, флаг переполнения) не используются.

Следует заметить, что предложенная запись для условного перехода лучше, чем просто добавление новых инструкций типа `subz`. Лучше она тем, что дает явное указание не только на тот факт, что используется только флаг ZF, но и на тот факт, что флаг ZF используется **только** для соответствующей инструкции перехода и в дальнейшем он использован не будет. Использование обоих отмеченных фактов представляет возможность для дополнительной оптимизации на уровне процессора, если рассматривать возможность разработки архитектуры процессора, реализующего систему симкоманд.

2. При использовании для сравнения обозначения `==` можно ошибиться и написать например, `eax = 0` вместо `eax == 0`. Эту возможную ошибку можно обойти путем использования другого

обозначения для сравнения (например, оставив запись `cmp a, b` вместо `a == b`) либо другого символического обозначения для присваивания (например, `<-` или `:=` вместо `=`). Другое символическое обозначение для операции присваивания нерационально, так как оператор `=` уже слишком твердо закрепился в языках программирования и вполне подходит для этой операции. В связи с этим предлагается пойти по другому пути — просто запретить запись `x == 0` в чистом виде, а разрешить ее только как составную часть какой-либо более сложной операции, в составе которой оператор `=` запрещен. В этом случае для ошибки нужно будет ошибиться как минимум дважды — во-первых, перепутать `=` и `==`, и, во-вторых, использовать неверную запись. Это обстоятельство сводит вероятность случайной ошибки к минимуму. Речь в данном случае идет о том, что запись `eax == 0` оставляет возможность случайной ошибки, а запись `eax == 0 : label` такой возможности практически не оставляет, так как запись `eax = 0 : label` является запрещенной.

Рассмотрим также пример для ARM-архитектуры:

```
subs r4, r5, #1
beq loop_label
```

Данной паре ARM-инструкций соответствует следующая симкоманда:

```
r4 = r5 - 1 == 0: loop_label
```

Несмотря на некоторую громоздкость этой записи, здесь также исключается вероятность случайных ошибок, связанных с перепутыванием `=` и `==`, так как следующие три записи не являются действительными симкомандами:

```
r4 = r4 - 1 = 0: loop_label
r4 == r4 - 1 = 0: loop_label
r4 == r4 - 1 == 0: loop_label
```

Псевдонимы для регистров

Псевдоним для регистра можно объявить как

```
eax'i = 0
```

или:

```
i'eax = 0
```

Первая форма удобна для такой записи:

```
ecx'i = 0
ebx' length = 0
edx' some_variable_with_long_name = 0
```

Так как это выглядит лучше, чем

```
i'ecx = 0
length'ebx = 0
some_variable_with_long_name'edx = 0
```

Вторая форма удобна для следующего примера [9]:

```
theta_10:
; Theta
lodsd          ; t = st[i ];
xor  eax, [esi+5*4-4] ; t ^= st[i+5];
xor  eax, [esi+10*4-4] ; t ^= st[i+10];
xor  eax, [esi+15*4-4] ; t ^= st[i+15];
xor  eax, [esi+20*4-4] ; t ^= st[i+20];
stosd          ; bc[i] = t;
loop theta_10
popad
xor  eax, eax

theta_11:
movzx ebp, byte[ebx+eax + 4] ; ebp = m[(i+4)];
mov  ebp, [edi+ebp*4] ; t = bc[m[(i+4)]];
movzx edx, byte[ebx+eax + 1] ; edx = m[(i+1)];
mov  edx, [edi+edx*4] ; edx = bc[m[(i+1)]];
rol  edx, 1 ; t ^= ROTL32(edx, 1);
xor  ebp, edx

theta_12:
```

На языке симкоманд данный ассемблерный код можно написать так:

```
theta_10:
t'eax = [st'esi ++ ]
t' (+) = st'[ 5*4-4]
t' (+) = st'[10*4-4]
t' (+) = st'[15*4-4]
t' (+) = st'[20*4-4]
[edi'bc ++ ] = t'.
--ecx == 0: theta_10
popad
eax'i = 0

theta_11:
ebp't = byte ebx'm[i+4]
't = 'bc[t*4]
edx's = byte m[i+1]
's = 'bc[s*4]
's (<<) = 1
't (+) = 's

theta_12:
```

Верификатор симкода проверяет, что объявленный псевдоним больше не используется по-

сле назначения другого псевдонима на этот же регистр.

Пример:

```
i'eax = 10
//...
|| Здесь можно обращаться к i', например, так:
i' += 2
\\...
c'eax = 0
//...
|| А здесь уже нельзя!
\\...
```

Если псевдоним был объявлен как регистр' псевдоним, то обращаться к нему можно только посредством записи 'псевдоним, а если объявлен как псевдоним'регистр, тогда посредством записи псевдоним'.

Переназначение псевдонима без явного освобождения (см. далее) на другой регистр не допускается, так как в ассемблере нет понятия областей видимости. Как следствие, можно по ошибке в большой функции добавить в середину новый код, который переназначая псевдоним изменяет таким образом поведение последующего кода.

Псевдоним можно освободить посредством такой записи:

```
i'. // освобождает псевдоним i'
i', r'. // освобождает псевдонимы i' и r'
```

После освобождения этот псевдоним можно назначать на любой регистр.

Заключение

В результате апробации языка симкоманд на примере, представленном в разд. "Симкоманды перехода", можно заметить повышение эффективности исполнения на процессоре, реализующем систему симкоманд. Эффективность при этом повышается и для программиста, это касается вопросов минимизации ошибок и более наглядной формы представления машинного кода.

Следует отметить, что симкоманды условных переходов спроектированы таким образом, чтобы исключить ошибку спутывания оператора сравнения == и оператора присваивания =, встречающуюся в языках высокого уровня (C++, PHP, JavaScript). Для таких языков, например, конструкция if (a = 5) c == 7; является допустимой, хотя содержит сразу две ошибки, так как правильно писать if (a == 5) c = 7.

При использовании симкоманд отмеченная ошибка исключена, так как

- запись `eax == 0` в чистом виде не является симкомандой;
- симкоманда `eax = 0 : label` является недействительной.

Для появления ошибки нужно будет ошибиться как минимум дважды — во-первых, перепутать `=` и `==`, и, во-вторых, использовать неверную запись, что сводит вероятность случайной ошибки к минимуму. Учитывая изложенные выше аргументы, можно полагать, что представленная система символьных макрокоманд (симкоманд) процессора является более наглядной формой представления машинного кода для программиста, а также она способствует минимизации ошибок в коде, что повышает продуктивность при создании программного кода.

1. **Язык** ассемблера — Википедия. URL: https://ru.wikipedia.org/wiki/Язык_ассемблера
2. **Характеристика** системы команд процессора. URL: <https://studfile.net/preview/5055379/page:16/>
3. **Макрокоманда** — Википедия. URL: <https://ru.wikipedia.org/wiki/Макрокоманда>
4. **Васенин В. А., Кривчиков М. А.** Формальные модели программ и языков программирования. Часть I. Библиографический обзор 1930—1989 гг. // Программная инженерия. 2015. № 5. С. 10—19.
5. **x86 Assembly/Arithmetic** — Wikibooks. URL: https://en.wikibooks.org/wiki/X86_Assembly/Arithmetic
6. **x86 Assembly/Logic** — Wikibooks. URL: https://en.wikibooks.org/wiki/X86_Assembly/Logic
7. **GitHub** — jlhonora / asm-examples. URL: <https://github.com/jlhonora/asm-examples/blob/master/palindrome/palindrome.asm>
8. **Condition Codes 1: Condition flags and codes—Processors blog—Processors—Arm community.** URL: <https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/condition-codes-1-condition-flags-and-codes>
9. **GitHub**—odzhan / tinycrypt /.../ k800x.asm. URL: <https://github.com/odzhan/tinycrypt/blob/master/permutation/keccak/old/k800x.asm>

Developing a System of Symbolic Macro Commands (Symcommands) of the Processor

A. V. Tretiak, alextretyak2@gmail.com, **E. A. Vereschagina**, everesh@mail.ru, Far Eastern Federal University, Vladivostok, 690091, Russian Federation,

P. V. Struk, pavelstruk@yandex.ru, Primorsky Aquarim, Vladivostok, 690922, Russian Federation

Corresponding author:

Tretiak Alexander V., Assistant, Far Eastern Federal University, Vladivostok, 690091, Russian Federation
E-mail: alextretyak2@gmail.com

Received on March 23, 2022

Accepted on April 15, 2022

The traditional form of recording machine code is assembly language. Its commands correspond directly to individual machine commands or sequences of commands. The syntax of an assembly language program is determined mainly by the instruction system of the specific processor and the directive system of the specific translator. For some platforms, there may be several kinds of assembly language syntax that are not compatible with each other. For example, the most popular assembly language syntaxes for Intel-compatible processors are Intel syntax and AT&T syntax. However, all existing types of assembly language syntax are primarily focused on easy readability by the computer, not the programmer. The system of symbolic macro commands of the processor presented in the article is a more clear form of representation of machine code, which promotes productivity when the programmer works with it, and also minimization of errors.

Keywords: processor instruction set, assembly language, macro command

For citation:

Tretiak A. V., Vereschagina E. A., Struk P. V. Developing a System of Symbolic Macro Commands (Symcommands) of the Processor, *Programmnaya Ingeneria*, 2022, vol. 13, no. 6, pp. 272—276.

DOI: 10.17587/prin.13.272-276

References

1. **Assembly language** — Wikipedia, available at: https://ru.wikipedia.org/wiki/Язык_ассемблера (in Russian).
2. **Characteristics** of the processor instruction set, available at: <https://studfile.net/preview/5055379/page:16/> (in Russian).
3. **Macro command** — Wikipedia, available at: <https://ru.wikipedia.org/wiki/Макрокоманда> (in Russian).
4. **Krivchikov M. A., Vasenin V. A.** Formal Models of Programming Languages and Programs. Part 1. Literature Review: 1930—1989, *Programmnaya Ingeneria*, 2015, no. 5, pp. 10—19 (in Russian).
5. **x86 Assembly/Arithmetic** — Wikibooks, available at: https://en.wikibooks.org/wiki/X86_Assembly/Arithmetic
6. **x86 Assembly/Logic** — Wikibooks, available at: https://en.wikibooks.org/wiki/X86_Assembly/Logic
7. **GitHub** — jlhonora / asm-examples, available at: <https://github.com/jlhonora/asm-examples/blob/master/palindrome/palindrome.asm>
8. **Condition Codes 1: Condition flags and codes—Processors blog—Processors—Arm Community**, available at: <https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/condition-codes-1-condition-flags-and-codes>
9. **GitHub** — odzhan / tinycrypt /.../ k800x.asm, available at: <https://github.com/odzhan/tinycrypt/blob/master/permutation/keccak/old/k800x.asm>

Software Engineering Education: Discovering Future Skills¹

S. M. Avdoshin, Professor, savdoshin@hse.ru, **E. Y. Pesotskaya**, Associate Professor, epesotskaya@hse.ru, **D. M. Kuruppuge**, Graduate Student, dkuruppuge_1@edu.hse.ru, HSE University, Moscow, 101000, Russian Federation

Corresponding author:

Pesotskaya Elena Y., Associate Professor, Faculty of Computer Science / School of Software Engineering, HSE University, Moscow, 101000, Russian Federation
E-mail: epesotskaya@hse.ru

Received on August 30, 2021

Accepted on March 17, 2022

In the contemporary digital age the majority of industries become very much software dependent, require automation, digitization, and deployment of new technologies. The need for specialists, skilled in software engineering, is in high demand. In this study, we set out to investigate what challenges may arise in IT students' skill sets and new competences in the software engineering context and answer the question "how is skills demand evolving and where the gaps will be?". The paper analyzes what specific knowledge needs to be developed during university education and identifies their evolving skills demand. The paper recommends how students and young specialists in software engineering need to change themselves to adapt to the changing world.

Keywords: software engineering, technologies, IT, education, skills

For citation:

Avdoshin S. M., Pesotskaya E. Y., Kuruppuge D. M. Software Engineering Education: Discovering Future Skills, *Programmная Ingeneria*, 2022, vol. 13, no. 6, pp. 277–285.

УДК 004

С. М. Авдошин, канд. техн. наук, проф., savdoshin@hse.ru, **Е. Ю. Песоцкая**, канд. экон. наук, доц., epesotskaya@hse.ru, **Д. М. Куруппуге**, студент, dkuruppuge_1@edu.hse.ru, Национальный исследовательский университет "Высшая школа экономики", Москва

Программные инженеры будущего: требования к навыкам

В современную цифровую эпоху большинство отраслей становятся очень программно-зависимыми, требуют автоматизации, цифровизации и внедрения новых технологий. Потребность в специалистах, разбирающихся в программной инженерии, высока. В статье приведены результаты исследования того, какие могут возникнуть проблемы в наборе навыков при обучении программных инженеров и потребности в новых компетенциях при разработке программного обеспечения. Дан ответ на вопрос "как меняется спрос на навыки и где будут пробелы в знаниях?". Приведен анализ того, какие конкретные знания необходимо развивать во время обучения в вузе, и определена потребность в растущих навыках студентов. Даны рекомендации, как студентам и молодым специалистам в области программной инженерии необходимо изменить себя, чтобы адаптироваться к меняющемуся миру.

Ключевые слова: программная инженерия, технологии, ИТ, образование, навыки

Introduction

The future of work will be a race between education and technology.

Mauricio Macri, President of Argentina in 2018

With the transition of business to the Internet, many vacancies opened up, hence, employees with

IT skills and software engineers are in high demand. Employment requirements became more stringent as the workforce needs to understand and become familiar with new technologies and how they affect the industry landscape. Specific knowledge needs to be continuously updated, accordingly, workers need to adapt. In the future, employers will seek out candidates that can continuously acquire and perfect new skills.

¹ The article is based on the materials of the report at the Seventh International Conference "Actual problems of Systems and Software Engineering" APSSE 2021.

Technical expertise remains critical and in short supply; 65 % of global CIO (Chief Information Officer) survey participants say analytics and data science will be the hardest-to-find technical skills in the next five years, while 54 % named cyber and 49 % named emerging technologies [1].

In the emergent technology context where industry processes become more automated, jobs and roles change accordingly. Most tactical, algorithmic, and structured work will be automated by machines over time, reducing the need to perform several repetitive tasks manually. Many jobs require both routine technical and non-routine tasks that are not easily computerized, such as interpersonal interaction, flexibility, adaptability, and problem-solving skills. The cognitive capacity of emerging technology is increasing to the point where machines will soon outperform humans in many workplace tasks.

Automation allows employees to concentrate on more important tasks, leaving much creative and strategic work for human talent. Today, soft skills are having a breakout moment. These enduring, essentially human skills, are increasing in value in part because they cannot be replicated by machines [1]. It seems that new skills and capabilities should be needed, as tomorrow's technology workforce will be different from today's. This means that software engineering student education and skills development should be redesigned to sustain technological evolution. Emerging information and communication technologies, such as Big Data, Internet of Things, Machine learning, and Artificial Intelligence, demonstrate a dramatic impact on the labor market.

Our paper focuses on software engineers and specialists with digital skills, as they are the ones who create and boost emerging technologies which are so in demand nowadays. Today access to knowledge for IT professionals is virtually limitless thanks to private players in the education market and access to online courses. In the Software Engineering field potential employers look less at university education and more at the experience and practical knowledge, including the presence of soft skills.

The goal of this paper is to understand how the development of skills is approached based on the influence of new technologies and, based on the findings, to propose a high-level approach on how software engineers should develop their future skills. We believe that software engineering students, graduates, and professionals in software engineering, will be better prepared for the current and future IT job market and business requirements relying on the results of this research and analysis.

The study was carried out at HSE University, which is among the top 4 universities in Russia in terms of

IT education, according to the QS World University Rankings, 2020/2021 [2], and 322nd among the 500 best universities worldwide, according to the QS World University Rankings in 2020 [3]. The study was conducted by interviewing IT students, ranging from software engineers to specialists studying information communication technologies, to find out which skills are needed for young specialists in IT and software engineering and what is lacking in the current education system. In order to understand future IT skills phenomena, this study analyzes emerging technologies and related literature in Section 1. In Section 2, we describe the methodology with our data collection and analysis methods and the results of our analysis. Section 3 discusses the limitations and future research avenues. Last section concludes our study with the key findings summary.

1. Emerging technologies impact on the software engineering workforce

1.1. Emerging Technologies and skills demand

We have analyzed the descriptions of many vacancies and made conclusions on which hard skills software engineering specialists should have. As for soft skills, employers and companies rather accentuate the common skills necessary for all areas of emerging technologies, without highlighting any in particular for one technology or another.

1.1.1. Blockchain

Blockchain has moved from the world of crypto currency to business, becoming the solution to many problems. Distributed crypto currency technology is already broadly used in many areas, from trading any commodity to voting in elections.

Blockchain specialists help to integrate blockchain technologies into business operations to improve existing services and introduce new ones. Many companies are searching for specialists with blockchain skills to create internal crypto coins for a company, for which regular customers can receive bonuses.

Blockchain developer requirements include experience with JavaScript, C++, Python, cryptography and machine learning. To make this job more attainable, one needs to be proficient in the Java and Solidity programming languages, know the Oracle Identity software suite, have database management skills and understand blockchain platforms such as Hyperledger Fabric, Ripple and the Bitcoin payment network.

1.1.2. Artificial Intelligence

Artificial Intelligence (AI) is the ability of a computer or a robot to solve tasks that are usually performed by humans. AI learns from the interpretation of external

data and uses the knowledge gained to achieve goals, due to the specifics of learning; this area is closely related to machine learning. AI can be useful in many areas where automation, improvement, process simplification and forecasting are in demand. In this area developers are required to understand neural networks, their areas of application, the ability to program in Python, as well as to be fluent in statistical analysis and mathematical modeling.

1.1.3. Internet of Things (IoT)

This technology consists of physical devices that read external data, upload it to the cloud (the Internet), where the software processes it, hence the connection with big data. An example of such a device is a smart home that adjusts the temperature according to the environment. Use of IoT together with machine learning has spread to sectors like agriculture to monitor plant health [4] which shows that IoT is applicable in all kinds of spheres. With IoT, companies can automate processes and reduce labor costs, leading to better service delivery. IoT specialists must, first of all, have engineering skills: the ability to work with wireless sensor networks, cloud technologies, as well as know how to ensure the security of the products being developed, and be proficient in at least one programming language such as Java, .NET, Python, PHP, Go, C++.

1.1.4. Nanotechnology/Biotechnology

Nanotechnology is an interdisciplinary area at an intersection of science (fundamental and applied) and technology. Nanotechnology includes the development and creation of electronic circuits at the atomic and molecular levels, the development of robots (nanomachines, nanobots) the size of a molecule, as well as methods for studying such objects. Key skills that are required by a nanotechnologist are excellent analytical and problem solving skills, excellent IT skills and being good at mathematics.

1.1.5. Augmented Reality (AR)/Virtual Reality (VR)

Augmented or virtual reality can be useful not only in entertainment, but also in business processes, for example, staff training, project demonstration, as well as medicine, industry and tourism. The development of this direction is very promising. Such a modern use of AR is shown by a prototype system to reduce collision situations by drivers using AR to overlay guidance information and an AI based co-driver as a real-time alternative to the driver [5]. To succeed in this area, one needs to have rich imagination, possess the knowledge of the algorithms of computer vision, be proficient in programming languages, for instance Python, Java, C#, be able to create mobile and desktop applications, and have a good mathematical background.

1.1.6. Robotics

In robotics, the main task is to create automated systems, i.e. robots. The use of machines is multifaceted, including medicine, manufacturing, space industry, etc. To get a better understanding of modern robots, a recent survey on intelligent bots [6] shows their usability and consumer experience with them. In robotics, it is necessary to understand algebra, mathematical analysis, and geometry, have knowledge in the field of automatic control and the theory of designing automatic systems (software engineering).

The above mentioned emerging technologies are summarized in the table 1 with respect to the necessary skill.

1.2. Future Changes in the Software Engineering Skills Workforce: Literature review

The digital era and emergent technologies suggest a greater need to empower skill sets and career development for technology professionals, so they can be familiar with augmented and virtual reality, blockchain, unmanned vehicles, and other innovative technologies.

It is very difficult to predict the demand and success of professions, therefore, constant training and mastering of new competencies will allow specialists to remain in demand. It became more convenient for employers to reskill the current employees instead of hiring new ones. Employees are supporting this as they want to stay employed and can upskill their skills using the convenience of online courses. According to a recent McKinsey Global Survey, 87 % of executives said they are experiencing a skill shortage or expecting it for several years, which is why there is a trend towards online education and retraining [19].

A wide range of technical skills are needed for cloud computing, although security, networking, virtualization skills, and big data analytics are considered to be the key skills for cloud managing and analyzing very large data sets, such as images, text, or speech which are prominent future demanded skills. Competencies like data analysis, data selection, and data security became relevant for all employees in an organization [7].

Conferring to studies of the productivity of skilled workers, there are various factors that contribute to higher job performance. According to Erne [8], those factors are the quality of group interactions and skill development.

Also, a survey by Ahmed et al. [9] discovered, as a result of their survey, the main soft skills requirements for the software market in North America, Europe, Asia and Australia. They surveyed jobs advertised in online portals such as eurojobs.com (Europe), monsterindia.com (Asia), seek.com.au (Australia) and workopolis.ca (North America). According to the study, the main soft

Hard Skill Requirements for Software Engineering Specialists

Skill demanded	Emerging technologies with skill demand (skill highly demanded +/not relevant n/a)					
	Blockchain	AI	IoT	Nanotechnology/ Biotechnology	AR/VR	Robotics
Database	+	+	n/a	n/a	n/a	n/a
Programming	JavaScript, C++, Python	Python	Java, .NET, Python, PHP, Go, C++	n/a	Unity, SLAM, ARKit	Assembly, Java, MATLAB, Python, C/C++
Machine learning	n/a	+	+	n/a	n/a	+
Cybersecurity	n/a	n/a	+	n/a	n/a	n/a
Probability theory, statistics	n/a	+	n/a	n/a	n/a	+
Physics, chemistry, biology	n/a	n/a	n/a	+	n/a	+
Mobile application development/Web development	n/a	n/a	+	n/a	+	n/a
Algorithms and data structures	+	+	n/a	n/a	n/a	n/a
Software architecture	n/a	+	+	n/a	+	n/a

skills necessary for software engineering specialists are analytical and problem solving, organizational skills, fast learning, interpersonal skills, team playing, being open and adaptable to change, the ability to work independently, and being innovative.

Another survey, conducted by Matturro et al. [10] with software development team members and team leaders from the Uruguay software industry, arrived at the conclusion that soft skills, such as communication skills, leadership, interpersonal skills, customer orientation, and teamwork, are the most crucial skills needed for team leaders. Whereas commitment, responsibility, analytics, problem solving, motivation, teamwork and eagerness to learn, are the most important for team members.

During the first semester, students often struggle with their competencies and soft skills. Thurner et al. [11] was able to identify four reasons for this by conducting a learning project that concerns not only technical but also non-technical skills. One of the facts this study revealed is that the soft skills of students can be improved by self-reflection and self-awareness.

A survey by Reis J. et al. describes digital transformation as the use of new digital technologies that enables major business advances and influences all aspects of customers' life [12]. Further, they specify that skills such as communications skills and knowledge of virtual worlds are required in workplaces,

to either virtualize or to maintain control of people working remotely. Peter C. Verhoef et al. highlights in their research, the most essential digital properties and capabilities needed by businesses for digital transformation. They are digital assets, digital agility, digital networking capability, and big data analytics capability. According to them, human resource management views digital transformation as attracting employees with digital and analytical skills [13]. This shows that digital skills do not necessarily mean "tech" skills.

Our main assumption we have arrived at is that generic digital skills are not sufficient for emerging skills. Now employers working with these technologies want to see in their employees not only highly qualified specialists with all the necessary professional skills, but also individuals with such qualities as communication skills, flexibility, and adaptability to change. Currently, in the digital era it has become common to use digital educational and training content in order to overcome existing communication barriers and effectively complement existing and developing online education platforms [14]. The ability to learn fast with such educational platforms and tools becoming more important than acquiring specific knowledge.

We need to change software engineering specialists into digital talents so they can survive in the emerging technologies landscape. One of our crucial

Table 2

Soft Skills and References

Skill demanded	References
Creative thinking	[13, 15, 16]
Flexibility	[13, 17, 18]
Critical thinking	[12, 15, 19]
Adaptability and resilience	[13, 19–21]

recommendations is to improve meaningful skills, based on current labor market intelligence. This would also help IT specialists on furlough (due to the Covid pandemic), or who are worried about their jobs, to gain the skills for which there are current vacancies.

A summary of the most demanded skills associated with emerging technologies are presented in table 2.

A person who possesses all of the above skills will always be in demand in the labor market, and will also be able to negotiate on the conditions on which he will cooperate with companies.

It is quite difficult for an employer to find an employee with all these qualities. It is easier for employers to direct existing employees to improve their qualifications and create an "ideal" employee, rather than looking for one in the labor market. In the near future, companies that want to stay afloat will have to increase the budget for employee training. This, in turn, will help strengthen the company's position in the future and not crash when disruptions occur.

2. Case Study Methodology and Results

This research was conducted by means of a systematic literature review (SLR), following Kitchenham [22], with the purpose of understanding how the development of skills is approached in the existing pandemic and

remote work situation. Based on the findings we analyze the demands for digital skills for emergent technologies and identify the gaps in digital skill development. We propose a high-level approach on how software engineering specialists should develop and grow in their future skills and how to upskill students with a software engineering background.

The research was conducted to answer the following questions:

- RQ1: how do universities approach the development of skills in their curricula?
- RQ2: what hard/soft skills are most important for software engineering specialists to become more successful in their IT or digital career in the near future?

These questions helped in carrying out a systematic review process in order to identify applicable information in papers, journals and reports that can contribute to the goal of this research.

The data collection process was conducted through an online survey to get a better understanding on what students expect and what they need from education vs. what they actually get. We were expecting to discover what skills they learn outside the universities and what skills could help them to survive in employment in their future.

In the following sections we describe the methodology used to perform the research, how we collected data, and the criteria we used when selecting participants. Furthermore, it is described how we analyzed the selected data and the results obtained.

Data was collected from students who studied IT related subjects, such as software engineering, business informatics, and applied mathematics. The choice of students and specialties was based on the author's research on the correlation between specializations and emerging technologies (table 3).

As a result we developed a questionnaire to gather their general information such as the course they are

Table 3

Correlation between specializations and emerging technologies

Emerging Technology	University specialization
Blockchain	Information and Communication Technologies and Systems, Applied mathematics and Information Science, Business informatics, System and Software Engineering
AI	Information and Communication Technologies and Systems, Applied mathematics and Information Science, Software Engineering, System and Software Engineering
IoT	Information and Communication Technologies and Systems, System and Software Engineering
Nanotechnology/Biotechnology	Nanotechnology/Biotechnology
Augmented Reality/Virtual Reality	Software Engineering, System and Software Engineering, Other
Robotics	Applied mathematics, Information and Communication Technologies and Systems



Specialization

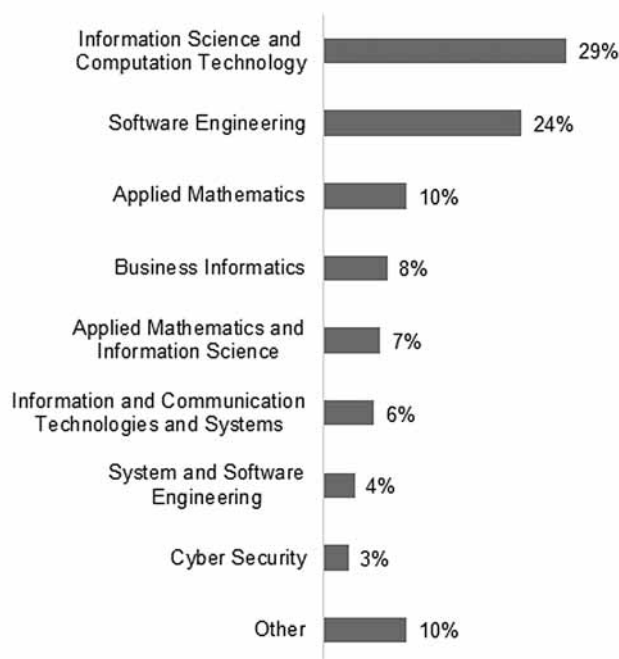


Fig. 1. Participants Specialization

studying, city/campus, and the name of the educational program in which they are studying. We then moved on

to questions related to skills, work experience, future professional expectations etc. The survey was voluntary and strictly anonymous.

The results show a summary of the survey in terms of the course of study; working experience; future IT professions they expect, in their opinion; the most useful IT skills they learned at university, and which skills are not adequate for a successful future IT profession, and more.

The participants (131) are studying under Information science (29 %) and Software engineering (24 %) programs. The rest are engaged in Applied mathematics (10 %), Business informatics (8 %), Applied mathematics and Computer science (7 %) and other programs related to computer science (fig. 1). The survey showed about 31 % of the participants have working experience in a software engineering specialty, while 44 % of them do not have any. 16 % of the participants are recorded as working/interned but not in the speciality they are studying. A smaller percentage of participants (10 %) are marked as worked/interned but are currently not working.

When asked about their future profession after graduation, the majority chose data analyst (data engineer, data science). In second place, backend developer was chosen, while mobile application developer came third on the list. Professions like game developer, BigData analyst and web developer, respectively, received fairly high responses as well. Less popular choices were PHP, technical writer, and scrum master (fig. 2).

Question: What job do you want to do after graduation, in what role? Choose all the answers that suit you

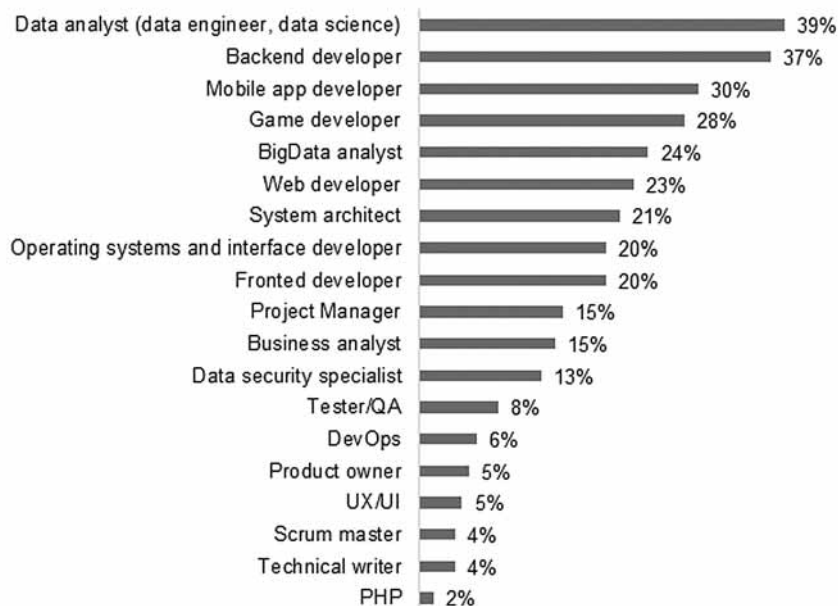


Fig. 2. Expected Future Profession

Question: What useful IT skills have you acquired for your future career while studying at the University? Choose all the answers that suit you

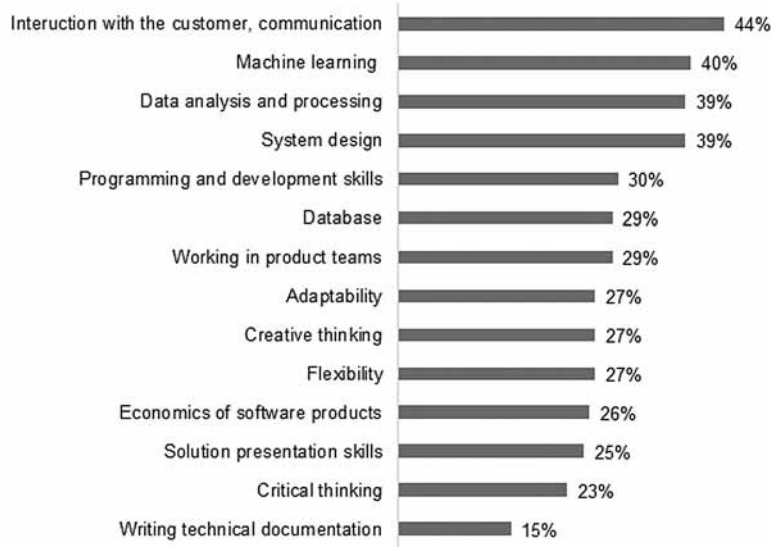


Fig. 3. Most useful IT skills that participants acquired during university studies for their future career

Participants were asked to list three hard skills that were useful during their studies and training period. Algorithms and data structures, object oriented programming, backend/frontend development, mobile application development, and programming languages were the significant choices that stood out among them.

When asked about the most useful IT skills they acquired at university which are beneficial for their future career, 44 % of the participants voted for interaction with the customer, and communication. A hard skill, such as machine learning were voted second, while data analysis and processing along with system design voted for equally. They recorded writing technical documentation as the least useful skill (fig. 3). Because the respondents chose multiple answers, the total amount is not equal 100 %

Most popular hard skills during university studies listed by the participants:

- Backend/Frontend development;
- Mobile application development;
- Machine learning;
- Object oriented programming;
- Programming languages; C#, C++, Java, Python, Js;
- Databases;
- Mathematical analysis;
- Algorithms and data structures;
- Operating systems;
- Web development;
- Quality Assurance and Testing;
- Software architecture;

- Algebra;
- Computing systems and computer networks.

According to the participants, the soft/hard skills that are missing from the university curriculum for a successful IT career in the future, are interaction with real companies and customers, and lack of necessary subjects (isolation of education from the labor market) respectively (fig. 4). The choices voted for the least are practical work (too much theory) and obsolete studying approaches.

Question: What are you missing and what can be strengthened in the current training format?

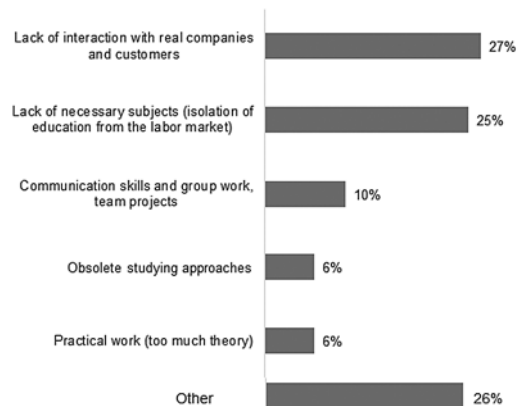


Fig. 4. Skills missing from the university curriculum for a successful IT career in the future

Table 4

Skills summation

Trending skills	IT Students survey (Valid, %)
Programming and development skills	62
Critical Thinking	51
Adaptability and resilience	44
Database	43
Writing technical documentation	32
Flexibility	30
Creative Thinking	25

When asked about the most useful IT skills which participants found most useful during university studies, 62 % of the participants voted for programming and development skills. A soft skill, like critical thinking was voted second (51 %), while adaptability and database was voted as 44 and 43 % accordingly. They recorded technical documentation, flexibility, creative thinking, data analysis and presentation skills, as not the most important skill but also the most highly demanded.

A summation of the main results of the survey and conclusions from the research are shown in table 4. The highlighted rows show that our findings from the survey validate our findings from the research. They are listed in descending order of participant votes from the survey. Some of the hard skills related to the IT area were not specified among general trends.

Adapting to the above skills is the foundation for securing a successful future profession. However, the skills that they cannot acquire during that time can be gained from other means, such as online courses or working as interns. Furthermore, we can propose that at least a portion of the university curricula should be based on what the individual chooses to learn so that they can pursue their interests.

3. Limitations and Future Work

The research was completely remote. Even though we followed a protocol to conduct the survey, the limited number of participants of this empirical study might affect its validity, scalability, and applicability of the results to broader contexts. The survey was circulated only among HSE University students. IT professionals were not included as survey participants as our goal was to discover skills needed by software engineering students and young specialists for their future, to work with emerging technologies. In our future work we are hoping to involve IT professionals to identify

their required skills and to update the research results considering their needs. This will help to leverage the supposed student's career development plans with the IT professionals' expectations.

Conclusion

To adapt to emerging trends, soft skills are a significant part of a student's development as well as hard skills. However, the majority of the higher education institutions generally neglect them in the computer science and software engineering curricula. Considering the importance of soft skills for the process of digital transformation and to keep up with emerging trends, the main question this study aims to answer is how we can improve computer science and software engineering students' skills, preparing them for the future market challenges by using industry emerging practices.

Our study showed that hard skills such as algorithms and data structures, object oriented programming, backend/frontend development, mobile application development, learning programming languages, and soft skills such as problem solving, dealing with uncertainty/adaptability, flexibility and creative thinking, are crucial skills that are absolutely vital for the software engineering specialist's successful career. We were able to confirm our findings using participants' responses. Furthermore, the study showed that soft skills, like interaction with businesses and customers, and a lack of necessary subjects resulting from the isolation of education from the labor market, are missing from the university curricula.

Relying on our literature overview and trends analysis, we concluded that we need to mold software engineering specialists into digital talents because new skills are required as new business opportunities arise. The automation process will change the workforce. We have identified it as being important to develop both soft skills as they are in high demand by employees, and hard skills which are required by employers, as they are the very basic requirements for employment and are crucial to keep up with emerging technical trends.

References

1. Autor D. Polyani's Paradox and the Shape of Employment Growth, *National Bureau of Economic Research*. 2014, Working paper 20485, pp. 39–40.
2. HSE University in Ratings. Moscow, Russia. Retrieved 11 01, 2021, available at: <https://strategy.hse.ru/rating>
3. QS World University Rankings. Retrieved 11 01, 2021, available at: <https://www.topuniversities.com/university-rankings/world-university-rankings/>
4. Espineli J. D., Lewis K. P. Internet of Things (IoT) Based Plant Monitoring Using Machine Learning, *Advances in Informa-*

tion and Communication. *FICC 2021. Advances in Intelligent Systems and Computing* Springer, Cham, 2021, vol 1363, pp. 278–289 DOI: 10.1007/978-3-030-73100-7_21.

5. **Bram-Larbi K. F., Charissis V., Khan S., Lagoo R., Harrison D. K., Drikakis D.** Intelligent Collision Avoidance and Manoeuvring System with the Use of Augmented Reality and Artificial Intelligence, *Advances in Information and Communication. FICC 2021. Advances in Intelligent Systems and Computing*, Springer, Cham, 2021, vol. 1363, pp. 457–46, DOI: 10.1007/978-3-030-73100-7_32.

6. **Tep S. P., Arcand M., Rajaobelina L., Ricard L.** From What Is Promised to What Is Experienced with Intelligent Bots. *Advances in Information and Communication. FICC 2021. Advances in Intelligent Systems and Computing*, Springer, Cham, 2021. Vol. 1363, pp. 560–565, DOI: 10.1007/978-3-030-73100-7_40.

7. **Zanoon N., Alhaj A., Khwaldeh S.** Cloud Computing and Big Data is there a Relation between the Two: A Study, *International Journal of Applied Engineering Research*, 2021, vol. 12, no. 17, pp. 6970–6982.

8. **Erne R.** What is productivity in knowledge work? *A cross individual view. Journal of universal computer science*, 2011, vol. 17, no. 10, pp. 1367–1389.

9. **Ahmed F., Capretz L. F., Campbell P.** Evaluating the demand for soft skills in software development, *IT Professional*, 2012, vol. 14 (1), pp. 44–49.

10. **Matturo G., Raschetti F., Fontán C.** Soft skills in software development teams: A survey of the points of view of team leaders and team members, *IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, 2015, pp. 101–104.

11. **Thurner V., Schlierkamp K., Bottcher A., Zehetmeier D.** Integrated development of technical and base competencies: Fostering reflection skills in software engineers to be, *IEEE Global Engineering Education Conference (EDUCON)*, 2016, pp. 340–348.

12. **Reis J., Amorim M., Melão N., Matos P.** Digital Transformation: A Literature Review, *Springer Nature*, Portugal: Springer International Publishing AG, 2018, pp. 411–421.

13. **Verhoef P. C., Bart Y., Bhattacharya A.** et al. Digital transformation: A multidisciplinary reflection and research agenda, *Journal of Business Research*, 2021, vol. 122, pp. 889–901, DOI:10.1016/j.jbusres.2019.09.022.

14. **Avdoshin S., Chernov A., Pesotskaya E.** Digital Content Service for the Distance Learning, *Information Technologies*, 2020, vol. 26, no. 9, pp. 529–537, DOI: 10.17587/it.26.529-537 (in Russian).

15. **Engler S.** Lack of Skills Threatens Digital Transformation. Retrieved 10 02, 2021, from Gartner, available at: <https://www.gartner.com/smarterwithgartner/lack-of-skills-threatens-digital-transformation/>

16. **Gopisetty S.** Global Pandemic: Business Model Impact on Enterprises reTHINK, reIMAGINE, reINVENT Businesses, *IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*, Atlanta, GA, USA, 2020, pp. 114-120, DOI: 10.1109/CogMI50398.2020.00024.

17. **Barr M., Nabir S. W., Somerville D.** Online Delivery of Intensive Software Engineering Education during the COVID-19 Pandemic, *IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*, Munich, Germany, 2020, pp. 1-6, DOI: 10.1109/CSEET49119.2020.9206196.

18. **Garousi V., Giray G., Tuzun E., Catal C., Felderer M.** Closing the Gap Between Software Engineering Education and Industrial Needs, *IEEE Software*, 2020, vol. 37, no. 2, pp. 68–77, March-April 2020, DOI: 10.1109/MS.2018.2880823.

19. **Agrawal S., Smet A. D., Poplawski P., Reich A.** Beyond hiring: How companies are reskilling to address talent gaps: Survey. Retrieved 12 03, 2021, from McKinsey & Company, available at: <https://www.mckinsey.com/business-functions/organization/our-insights/beyond-hiring-how-companies-are-reskilling-to-address-talent-gaps>, February 2020.

20. **Liu Y., Vijay A., Tommasini S. M.** et al. Hands-on engineering courses in the COVID-19 pandemic: adapting medical device design for remote learning, *Phys Eng Sci Med*, 2021, vol. 44, no. 1, pp. 195–200, DOI: 10.1007/s13246-020-00967-z.

21. **Kanij T., Grundy J.** Adapting Teaching of a Software Engineering Service Course Due to COVID-19, *IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*, Munich, Germany, 2020, pp. 1–6, DOI: 10.1109/CSEET49119.2020.9206204.

22. **Kitchenham B.** *Procedures for performing systematic reviews*, Keele, UK, Keele Univ., 2004, 33 p.

ИНФОРМАЦИЯ

X Международная научная молодежная школа-семинар «Математическое моделирование, численные методы и комплексы программ имени Е. В. Воскресенского»

14—18 июля 2022

Приглашаем Вас принять участие в X Международной научной молодежной школе-семинаре «Математическое моделирование, численные методы и комплексы программ» имени Е.В. Воскресенского, которая состоится в смешанном формате (очно и дистанционно) с 14 по 18 июля 2022 г. на базе Национального исследовательского Мордовского государственного университета им. Н.П. Огарёва.

СЕМИНАРЫ ШКОЛЫ

1. Принципы построения математических моделей.
2. Численная реализация алгоритмов математических моделей.
3. Динамические системы и качественные методы анализа математических моделей.
4. Математические модели физики, химии, биологии, экономики, социологии, экологии и других отраслей науки.

Подробности: <http://agora.guru.ru/SCHOOLSEMINAR2022/>

Using Virtual Reality Technology in the Design of Complex Systems¹

N. M. Borgest, PhD in Technological Sciences, borgest@yandex.ru,

S. A. Vlasov, Postgraduate Student, vlasov.ssau@mail.ru, Samara University, Samara, 443086, Russian Federation

Corresponding author:

Vlasov Sergey D., Postgraduate Student, Samara University, Samara, 443086, Russian Federation

E-mail: vlasov.ssau@mail.ru

Received on April 22, 2021

Accepted on April 26, 2022

This article provides a concrete example for using and creating a virtual environment for computer-aided design of complex systems. The subject area is design, and the object of research is an aircraft. CAD creates a parameterized 3D model, which is demonstrated and processed on a virtual stage using special equipment. Special equipment such as the HTC Vive Pro virtual reality helmet and the included controllers. The virtual assistant of the designer allows you to create detailed parameterized models of an aircraft, in particular, a transport aircraft based on the An-124 Ruslan, and virtual reality helps the user to learn more about the created project and make the necessary optimization edits directly in VR space. By changing the model in VR space, it will change in the main project. The work is a continuation of the development of a decision-making system, a virtual assistant to the designer: "Robot designer", whose application interface has been defined. The main control menu for the user in virtual space is shown, and the aircraft with the fuselage skin and cockpit removed is also shown. Plans for further use of virtual reality technology in the design of complex systems and training of specialists with the profile of aircraft engineers are presented.

Keywords: virtual reality, robot designer, 3D model, finite element model, thesaurus, aircraft

For citation:

Borgest N. M., Vlasov S. A. Using Virtual Reality Technology in the Design of Complex Systems, *Programmnyaya Ingeneria*, 2022, vol. 13, no. 6, pp. 285–290.

УДК 629.7.01

Н. М. Боргест, канд. техн. наук, проф., borgest@yandex.ru,

С. А. Власов, аспирант, ст. препод., vlasov.ssau@mail.ru, Самарский Университет им. С. П. Королева

Использование технологии виртуальной реальности при проектировании сложных систем

Приведен конкретный пример использования и создания виртуальной среды для автоматизированного проектирования сложных систем. Предметной областью является проектирование, а объектом исследования — летательный аппарат. САПР создает параметризованную 3D-модель, которая демонстрируется и обрабатывается на виртуальной сцене с помощью

¹ The article was based on the materials of the report at the IX All-Russian Scientific Conference "Information Technologies for Intelligent Decision Support" ITIDS'2021.

специального оборудования. Специальное оборудование — шлем виртуальной реальности HTC Vive Pro и прилагаемые контроллеры. Виртуальный помощник дизайнера позволяет создавать подробные параметризованные модели воздушного судна, в частности, транспортного самолета на базе Ан-124 "Руслан", а виртуальная реальность помогает пользователю узнать больше о созданном проекте и внести необходимые оптимизационные правки непосредственно в VR-пространстве. При изменении модели в VR-пространстве, она изменится и в основном проекте. Работа является продолжением разработки системы принятия решений, виртуального помощника проектанта "Робот-проектант", интерфейс приложения которого был определен. Показано главное меню управления для пользователя в виртуальном пространстве, а также показан самолет со снятой обшивкой фюзеляжа и кабиной пилота. Представлены планы дальнейшего использования технологии виртуальной реальности при проектировании сложных систем и подготовке специалистов по профилю авиационных инженеров.

Ключевые слова: виртуальная реальность, робот-проектант, 3D-модель, конечно-элементная модель, тезаурус, самолет

Introduction

The design processes have become more specialized and intensive, but still they are performed by human designers. The designers specialize in their own fields, and in order to solve challenging problems and implement larger projects, they have to cooperate with other designers. When comparing different design processes, one can notice a variety of forms that reflect the richness of product options and how said products are developed and further sold to many different people [1]. Now computer-aided design (CAD) systems are recognized as the basic systems in the design process. 3D models created in CAD systems provide a natural way to exchange project information between the designer and other participants of the production process. However, to share a CAD model between different people, each person should have access to a CAD workstation. This is reasonable if the involved participants work in the same company, preferably if they work in the same or related departments, and of course, they also need to know how to use the CAD system. In practice, the situation is often very different. Model data must be transferred from one system to another, often using the Internet as a data transfer medium. Due to the continuous development of CAD systems, they have become sophisticated tools for modelling 3D objects as well as highly parameterized and powerful tools in presentations, analysis and modelling. However, many problems still remain unresolved. For example, CAD does not provide support at the early design stage, although this stage is recognized as the most important for product development and it has a large impact on development and production costs. In addition, the old paradigm of using 2D design interfaces for modelling 3D objects is still applied, etc. [2].

Virtual reality (VR) is rapidly developing and offers interactive immersion into a different reality, in which

both science and industry, medicine [3], and tourism [4] are interested. From pilots to surgeons [5], VR has a strong impact on learning through embodied cognition, psychomotor abilities, and a high level of retention [6].

1. Related work

The main goal of this work is to develop and implement a computer system to assist in decision-making and studying parameterized 3D models in CAD as well as to manipulate them with the addition of parts and enable simultaneous data exchange. For effective aircraft design, it is necessary to move to the maximum possible modular structure that allows data exchange and automates the production process.

Due to the possibility of exchanging models through the Internet and taking other limitations into account, the system shown in fig. 1 has been developed. During the study the following tools have been used: CAD system CATIA [7] for the creation of parameterized models, a plugin written for UNITY [8] for representation of geometric data models in VR, XML as a carrier of configuration data, and C# as programming language [9]. The diagram of the work of the project manager is shown in fig. 1.

The VR project involves the introduction of this technology in the design of the aircraft, showing all the individual features of the aircraft design. It provides disassembling the aircraft by one click of the manipulator and the possibility of seeing all the individual parts [10].

In one of the latest versions of the "Robot designer" project [11], an application interface was developed that allows you to create a parameterized three-dimensional model of a transport aircraft. To build it, the user needs to enter basic data on the aircraft, the so-called technical task. The interface of the classic version is shown in fig. 2, see the 2nd side of cover.

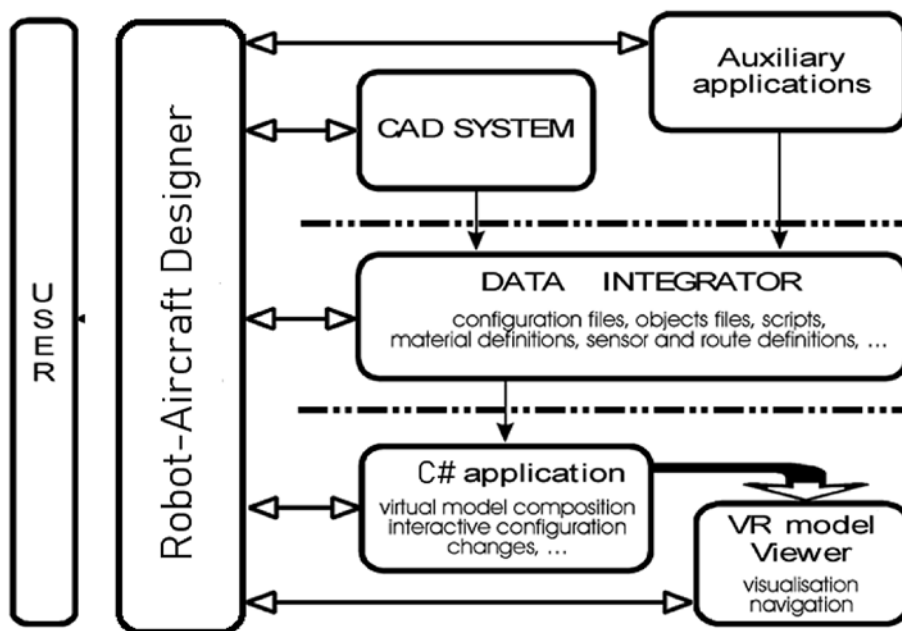


Fig. 1. Diagram of the work of the project manager

The interface of the "Robot designer" consists of several main parts: the project timeline (1), the interactive screen for changing the design parameters (2), the project matrix (3), the model preview window (4), the button for switching the application to virtual reality (5).

The interface is directly connected to the CATIA CAD system [12]. The user can move the model in space as it suits them, scale and rotate the model. Also, for convenience, the user can change the type of the model with one click [13]. There are four options for standard model positions in space: front view, side view, top view and isometry.

In addition to displaying the model in the classic interface [14], it is necessary to provide a more visual comparison image and an intuitive way to interact. This task is effectively handled by visualization using a virtual reality headset, which allows you to deepen the perception of the demonstrated model both from the point of view of development and from the point of scale relative to the user [15, 16].

This project was developed on the Unity platform [17] and tested on the HTC VIVE Pro eye equipment [18] during classes aimed at teaching students of the aviation industry. Currently, there is only one aircraft model built in CAD CATIA [19]. This is a model of the An-124 Ruslan transport aircraft. In a VR helmet, the user is transferred to a virtual hangar with an aircraft model. One of the joysticks allows you to move around the scene with the model shown in fig. 3, see the 2nd side of cover.

First of all, the design of the aircraft is the main point of interest. Taking into account said condition, the cockpit, the cargo compartment, and the locations of passengers and crew are omitted.

To provide comfort while working with the model, an interface with a special user menu is developed, see fig. 4, the 2nd side of cover. The interface has all the necessary functions. The user can move the entire aircraft around the virtual space, replace it around the model, measure the model using the joystick, disassemble the model in parts (remove the fuselage skin of the aircraft).

While the model is controlled in standard mode using a conventional computer mouse, it is controlled in virtual space using special joysticks. These joysticks are complemented by a virtual reality helmet. Using the joystick, user can move around the virtual space and perform manipulations with the model.

The cockpit is not empty and the user can see the location of the crew as a simple visual demonstration example, see fig. 5, the 2nd side of cover. Figure 5 also shows the location of the cabin floor, the room for accompanying cargo, the location and number of frames.

For a detailed analysis of the structure of the aircraft, the user can see the fuselage frame, the location of the frames, stringers and other power elements of the aircraft. By interacting with the menu bar, it is possible to select the appropriate "hide/show" function. Figure 6 (see the 3rd side of cover) provides the demonstration of said function.

The robot-designer is intended for visual demonstration of the aircraft design process [20]. A set of built-in methods and procedures allows to reduce the time for decision-making. In view of the above it is possible to move on to a detailed studying of the model quickly [21]. Therefore, the virtual reality technology allows one to see the model of the aircraft as close as possible and study the structure of the aircraft in detail. With a pair of controllers in hands, the user can use virtual hands to disassemble, measure, change, repaint, and rebuild the model of the aircraft. This technology is aimed primarily at young engineers, students and schoolchildren [22]. Figure 7 (see the 3rd side of cover) shows the process of working with the design robot in the VR mode. The parameterized model is built, the data is added by the user, namely: the range, the cruising speed, the number of seats for the crew, the commercial load.

The effectiveness of immersive design, conducted in VR, has been tested by the first-year students, specializing in aircraft engineering. The young engineers gave a positive feedback on the work with the "Virtual Assistant" system. The use of virtual aircraft design with the HTC VIVE helmet allows one to present the design of the aircraft in a more interactive way [23].

2. Working with other parts of the aircraft

In the design of an aircraft, there is a stage of selecting an engine for it. The choice of an engine for a projected aircraft consists of several stages.

- Preliminary selection of several real engines (or projects of promising engines). The choice is made by the amount of take-off thrust or power. When choosing these parameters, you should pay attention to the prototypes that are installed on similar aircraft.
- Take into account the capabilities of the engine, in the specified flight modes and operational requirements.
- The choice of the most suitable prototype engine.
- Preliminary selection of engines (or projects of promising engines) is carried out according to the scientific and technical characteristics, depending on their dimension (their thrust class or power). Each of the engines is considered as a possible prototype engine, the thrust or power of which will be specified based on the results of subsequent design calculations performed in the course work.

The analysis of the obtained engine variants should include a comparative quantitative assessment of their specific characteristics among themselves in the conditions of a determined technical specification. At the same time, the method of quantifying engines can be similar to the method described above for comparing the layout options of engines on an airplane.

The intelligent assistant "Robot designer" uses a selection engine according to the prototype of the projected transport aircraft. The user can select the appropriate type of engine and view its model, both in standard mode and in VR mode, as shown in fig. 8, see the 3rd side of cover.

3. Working with other parts of the aircraft

In the future, it is planned to create additional functions for working with the model within the present work [24]. Such as the development of a new model of the passenger aircraft based on the MC-21 aircraft with the option to access the cabin of the aircraft from within. This addition will require considerable time to recalculate basic parameters, create a new database since the calculation methods and the design of the passenger aircraft and the transport one are very different from each other.

It should be noted that there are also the plans to create multiplayer options in this application [25]. This will allow multiple users to be in the virtual space and work on the same model simultaneously. One person will be able to deal with the fuselage, another one — with the wing, the third — study the structure of the horizontal tail or model the landing gear and etc.

Another advantage of the present work is a remote usage. From anywhere in the world, the user will be able to connect to the server of the "Virtual Assistant" [26] and create 3D models or simply study the structure of complex systems, such as an aircraft, without leaving home just wearing a virtual reality helmet [27].

The project "Robot designer" has a great future, at least within the walls of Samara University. The project has acquired the appearance of an interface, parameterized 3D models of aircraft, a thesaurus [28], an integrated module for calculating aerodynamic parameters and a module for working with a model in virtual reality. Let the VR functionality now have only basic parameters (twist, rotate, measure, move). In future publications, it is planned to add a more detailed user interaction with the 3D model being created in the virtual environment itself. A detailed interaction is understood as a change in the size of the model and its geometry, these changes will take into account the limitations imposed when designing the aircraft. It is also planned to support this system with augmented reality technology [29].

It is also planned to develop a basic model of interaction with the system, which will take place in several stages. This connection will allow linking the "Virtual Assistant" with other disciplines of the aircraft engineering course and will help teachers integrate VR at any stage of training. This scheme can be seen in fig. 9 (see the 4th side of cover) [30].

Conclusion

In conclusion, the preliminary results of the study provide a positive incentive to continue exploring the use of virtual reality as a tool to support aircraft design concepts that may be difficult for design students to present, and to develop these applications using authentic design learning elements that can provide a solution to bridge the gap between theoretical knowledge and practical application. Further analysis is needed to better understand the expectations of students and their attitude to learning using these new technologies [31].

References

1. **Pokojski J.** IPA-Concepts and Applications in Engineering, *Intelligent Personal Assistant — Design Process Modelling, IPA-Concepts and Applications in Engineering. Decision Engineering*. Springer, London, 2004, pp. 81–98.
2. **Jezernik A., Hren G.** A solution to integrate computer-aided design (CAD) and virtual reality (VR) databases in design and manufacturing processes, *The International Journal of Advanced Manufacturing Technology*, 2003, vol. 22, pp. 768–774.
3. **Saito Y., Sugimoto V., Imura S., Morine Y., Ikemoto T., Iwahashi S., Yamada S., Shimada M.** *Intraoperative 3D hologram. support with mixed reality techniques in liver surgery*. Ann. Surg, 2020, 271 p.
4. **Guttentag D. A.** Virtual reality: Applications and implications for tourism, *Tourism Management*, 2010, vol. 31, no. 5, pp. 637–651.
5. **Rao A. K., Pramod B. S., Sushil C., Varun D.** Influence of Indirect Vision and Virtual Reality Training Under Varying Manned/Unmanned Interfaces in a Complex Search-and-Shoot Simulation, *Springer International Publishing*, 2019, vol. 780, pp. 225–235.
6. **Zikas P., Papagiannakis G., Lydatakis N.** et al. Immersive visual scripting based on VR software design patterns for experiential training, *The Visual Computer*, 2020, vol. 36, pp. 1965–1977.
7. **Brunetti G., Golob G.** A feature-based approach towards an integrated product model including conceptual design information, *CAD*, 2000, vol. 32, no. 14, pp. 877–887.
8. **Tang X., Shang S.** Virtual Writing Interactive Display Based on Unity / Eds: F. Xhafa, S. Patnaik, M. Tavana, *Advances in Intelligent, Interactive Systems and Applications. IISA 2018*. Advances in Intelligent Systems and Computing, vol. 885. Springer, Cham, 2018, pp. 627–632.
9. **Kim T. J., Huh J. H., Kim J.M.** Bi-directional education contents using VR equipments and augmented reality, *Multimed Tools Appl*, 2018, vol. 77, pp. 30089–30104.
10. **Wohlgenannt I., Simons A., Stieglitz S.** Virtual Reality, *Bus Inf Syst Eng*, 2020, vol. 62, pp. 455–461.
11. **Borgest N. M., Vlasov S. A., Gromov Al. A., Gromov An. A., Korovin M. D., Shustova D. V.** Robot-designer: on the road to reality, *Ontology of designing*, 2015. vol. 5, no. 4, pp. 429–449 (in Russian).
12. **Davidowitz G., Kotick P. G.** The Use of CAD/CAM in Dentistry, *Dent. Clin. N. Am*, 2011, vol. 55, no. 3, pp. 559–570. DOI: 10.1016/j.cden.2011.02.011.
13. **Zhang C.** Top-down parameter assembly design based on CATIA software, *Qinghai Univ (Nat Sci)*, 2007, vol. 25 (1), pp. 83–85.
14. **Vlasov S., Borgest N. M.** Application of virtual reality technology based on ontology in education of aviation, *Open Semantic Technologies for Intelligent Systems*, 2020, vol. 4, pp. 263–266 (in Russian).
15. **Grajewski D., Diakun J., Wichniarek R.** et al. Improving the skills and knowledge of future designers in the field of ecodesign using virtual reality technologies, *International Conference Virtual and Augmented Reality in Education. Procedia Computer Science*, 2015, vol. 75, pp. 348–358.
16. **Kapela J., Frimodig A., Hellman T.** Collaborative VR — a new era of design, *Seinäjoki University of Applied Sciences 2020: Competence in the strengths of the strategy*, 2021, Studies 33, pp. 130–140.
17. **Ježek B., Šimeček O., Slabý A.** Virtual Scene Components for Data Visualization, *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, 2021, pp. 3–16.
18. **VIVE United States: Discover Virtual Reality Beyond Imagination**, available at: <https://www.vive.com/us/>
19. **Jezernik A., Hren G.** A solution to integrate computer-aided design (CAD) and virtual reality (VR) databases in design and manufacturing processes, *Int J Adv Manuf Technol*, 2003, vol. 22, pp. 768–774.
20. **Martin P., Masfrand S., Okuya Y., Bourdot P.** A VR-CAD Data Model for Immersive Design. Eds L. De Paolis, P. Bourdot, A. Mongelli, *Augmented Reality, Virtual Reality, and Computer Graphics, AVR 2017. Lecture Notes in Computer Science*, Springer, Cham, 2017, vol. 10324.
21. **Dalton J.** 5 Reasons to Use Virtual Reality for Data Visualisation, available at: <https://towardsdatascience.com/5-reasons-to-use-virtual-reality-for-data-visualisation-86cd37d5c1ee>.
22. **Sampaio A. Z., Ferreira M. M., Rosário D. P., Martins O. P.** CAD and VR Technologies Used in Civil Engineering Education / Y. Luo (eds), *Cooperative Design, Visualization, and Engineering. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2010, vol. 6240.
23. **Kilteni K., Groten R., Slater M.** The sense of embodiment in virtual reality, *Teleoperators and Virtual Environments*, 2012, vol. 21, no. 4, pp. 373–387.
24. **Feng Y., Duives D., Hoogendoorn S.** Wayfinding behaviour in a multi-level building: A comparative study of HMD VR and Desktop VR, January 2022, *Advanced Engineering Informatics*.
25. **Richards J.** Infrastructures for Immersive Media in the Classroom., *Virtual, Augmented, and Mixed Realities in Education. Smart Computing and Intelligence / Eds. D. Liu, C. Dede, R. Huang, J. Richards*, Springer, Singapore, 2017, pp. 89–104.
26. **Borgest N. M., Grigoriev V. A., Kuzmichev V. S.** Artificial intelligence in aircraft design and the role of professor V. G. Maslov's scientific school in the process of its development, *VESTNIK of Samara University Aerospace and Mechanical Engineering*, 2021, vol. 20, no. 3, pp. 171–190 (in Russian).
27. **Borgest N. M., Vlasov S. A.** Development of a modular interface of the computer support system for aircraft design, *Journal of Physics Conference Series*, 2020, vol. 1661 (in Russian).
28. **Borgest N., Orlova A.** Ontological Modeling of Satellite's Manufacturing Work Flow Instruction, *Procedia Engineering*, 2017, vol. 185, pp. 146–152 (in Russian).
29. **Borgest N. M., Vlasov S. A.** Application of VR and AR Technologies in Educational Process, *Proceedings of the International Conference On Modelling, Simulation and Applied Mathematics*, 2017.
30. **Doerner R., Horst R.** Overcoming challenges when teaching hands-on courses about Virtual Reality and Augmented Reality: Methods, techniques and best practice, *Graphics and Visual Computing*, 2022, vol. 6, article 200037.
31. **Heidi Tan Yeen-Ju.** Using augmented reality to enhance students' learning of 3D modelling: a preliminary study, *Proceedings of The 2nd Conference on Managing Digital Industry, Technology and Entrepreneurship*, 2021, Faculty of Management, Multimedia University, Malaysia, pp. 64–67.

А. С. Козицын, канд. физ.-мат. наук, вед. науч. сотр., alexanderkz@mail.ru,
МГУ им. М. В. Ломоносова

Алгоритмы тематического поиска данных в наукометрических системах

Рассмотрены разработанные автором методы тематического поиска журналов и конференций научно-технического характера с использованием программных средств, реализующих алгоритмы полнотекстового поиска и поиска экспертов. В начале статьи дан краткий обзор существующих систем тематического поиска данных наукометрического содержания с указанием их основных недостатков. Приведено описание разработанных автором алгоритмов, их программной реализации и апробации на больших данных наукометрической системы. Для выполнения запросов разработанные алгоритмы используют полнотекстовые описания объектов в наукометрических системах и графы взаимных связей между объектами, проводят фильтрацию данных по узконаправленным тематическим запросам и ранжируют результаты поиска по степени их авторитетности.

Ключевые слова: тематический анализ, наукометрия, информационные системы, библиография, журнал, конференция

Введение

При создании информационных наукометрических систем для анализа больших объемов данных необходимо использовать интеллектуальные методы поиска и анализа информации, которые заменяют или дополняют использование традиционного контекстного поиска и поиска по атрибутам. Одним из таких методов является тематический анализ информации. Тематический анализ применяется для уточнения наукометрических критериев, анализа основных тенденций научно-технического развития, для информационного поиска и решения других важных задач.

Наиболее распространенные критерии, которые традиционно используются для анализа результатов деятельности научных работников и определения эффективных методов их стимулирования [1], в значительной степени зависят от тематики научных исследований. Например, средние показатели индексов цитирования научных статей в различных тематических областях значительно отличаются [2, 3]. Использование методов тематического анализа позволяет вычислять более объективные наукометрические критерии, например, нормализованную цитируемость [4].

Применение методов тематического анализа также помогает более детально оценить основные тенденции развития мировой науки, наиболее значимые научные направления на уровне го-

сударства или отдельной организации. Для возможности проведения такой оценки большинство разрабатываемых наукометрических систем и систем цитирования имеют модули аналитической обработки и агрегирования данных [5, 6] и специализированные интерфейсы для визуализации тематического распределения данных [7, 8]. Например, наукометрическая информационно-аналитическая система "ИСТИНА" (ИАС "ИСТИНА" [1]) позволяет проводить анализ распределения показателей научных публикаций по тематическим рубрикам Scopus и ГРНТИ [9, 10]. Анализ основных тенденций развития различных тематических направлений науки и своевременное отслеживание их изменений необходимы, в том числе, для выбора эффективных способов стимулирования научной деятельности [11, 12].

В настоящей работе описаны разработанные автором методы тематического поиска журналов и конференций научно-технического содержания. Исходными данными для такого поиска являются описания объектов (статей, докладов, проектов и других результатов научно-технической деятельности), информация о которых загружена в наукометрическую систему. Используются два подхода для поиска: контекстный поиск по описаниям и тематический поиск экспертов, статьи и доклады которых представлены в журналах и на конференциях [13, 14].

Тематический поиск журналов

Актуальность задачи автоматизации тематической классификации и поиска журналов обусловлена в первую очередь большим числом публикуемых в мире журналов и сборников статей, в том числе, содержащих материалы проводимых конференций. Подобные материалы публикуют на русском, английском, немецком, французском и других языках, что также затрудняет задачу тематической классификации [15]. Например, в ИАС "ИСТИНА" [1, 16] зарегистрировано более 70 тысяч журналов и еще более 200 тысяч сборников научных публикаций. Увеличение числа журналов обусловлено в первую очередь увеличением общего числа исследователей, поскольку каждый новый журнал можно рассматривать как социологический результат деятельности вновь образовавшегося сообщества ученых-исследователей. Известный британский историк науки Дерек Джон де Солла Прайс в середине прошлого века предсказывал, что уже в начале XXI века число научных журналов превысит 1 млн [17]. Реальные темпы роста числа научных журналов значительно уступают этим смелым прогнозам. По современным оценкам их количество в мире продолжает увеличиваться ежегодно примерно на 4 % [18]. Однако даже такие объемы информации не позволяют использовать в поисковых системах исключительно ручную обработку данных экспертами и требует создания механизмов автоматизации процесса тематического анализа и поиска.

Существующие системы цитирования предоставляют интерфейс для проведения тематического анализа по заданным рубрикам (Scopus, ГРНТИ и др.). Однако уровень такой рубрикации слишком высок для проведения полноценного тематического анализа. Кроме того, представленные в работе [19] примеры показывают, что используемые в промышленных системах (РИНЦ, Антиплагиат) методы полнотекстового тематического анализа имеют определенные недостатки в области точности и полноты распознавания тематики текстов, а также имеют существенные сложности при сравнении тематики издающихся на разных языках журналов.

В работе [19] предложено использовать графы соавторства [20] для проведения тематического анализа и оценки тематической близости научных журналов в наукометрических системах. Предложенный метод позволяет искать тематически близкие журналы на разных языках и проводить анализ без необходимости получения полных текстов статей. Такой подход значительно расширяет

область применения этого метода. В системе РИНЦ, например, полные тексты статей имеют менее 11 % журналов [21].

Исходными данными для поиска журналов являются граф соавторства, который можно получить из наукометрических систем или систем цитирования, а также исходное опорное множество авторов. В работе [19] в качестве опорного множества используется множество авторов выбранного журнала j_1 . На его основе строится множество $M(j_1, j_2)$ всех пар статей, которые были опубликованы в журналах j_1 и j_2 и имеют хотя бы одного совпадающего автора:

$$M(j_1, j_2) = \{d_1, d_2 \in D \mid d_1 \in D_{j_1}, d_2 \in D_{j_2}, A_{d_1} \cap A_{d_2} \neq \emptyset\},$$

где D — множество статей; D_j — множество статей в журнале j ; A_d — множество авторов статьи d .

Вес автора в каждой статье определяется как $w(a, d) = \frac{1}{2} + \frac{1}{2k}$ для первого автора и $w(a, d) = \frac{1}{2k}$ для остальных соавторов, где a — автор; k — число авторов.

Функция определения степени тематической связи между журналами при этом имеет вид

$$F(j_1, j_2) \equiv \sum_a W(a, j_1, j_2), \quad (1)$$

где

$$W(a, j_1, j_2) \equiv \min\left(\max_{d_1 \in D_{j_1}} (w(a, d_1)), \max_{d_2 \in D_{j_2}} (w(a, d_2))\right).$$

Представленный в работе [19] метод достаточно эффективен и обеспечивает точность 78 %. Существенным недостатком изложенного в работе [19] подхода является невозможность поиска журналов по тематике, заданной ключевыми словами. Для применения метода и построения первичного опорного множества необходимо знать хотя бы один журнал, тематика которого соответствует информационной потребности пользователя. Указанное ограничение не позволяет использовать этот метод для тематического поиска по ключевым словам.

В настоящей работе для построения опорного множества авторов и обеспечения возможности поиска журналов по ключевым словам предлагается использовать методы поиска экспертов [13, 14]. Согласно такому подходу, на первом шаге по поисковому запросу выполняется построение множества экспертов с учетом их "весового" соответствия поисковому запросу. Для этого с использованием алгоритма поиска экспертов для поискового

запроса q строится функция релевантности авторов поисковому запросу $v(a, q)$ и определяется множество авторов релевантных запросу $A_q = \{a \in A \mid v(a, q) > 0\}$. Функция тематического соответствия журнала запросу определяется как

$$F^*(q, j) \equiv \sum_{a \in A_q^* \cap A_j} W^*(a, q, j),$$

где $W^*(a, q, j) \equiv v(a, q) \max_{d \in D_j} (w(a, d))$.

Схема работы алгоритма поиска представлена на рис. 1.

Программная реализация алгоритма выполнена на языке PL/SQL. Основная функция поиска имеет следующее описание на этом языке:

```
create or replace function find_journal4them
(keywords varchar2, --Список ключевых слов через запятую
type_id NUMBER default null, --Тип объектов для поиска опорного множества экспертов
is_graph NUMBER default 1, --Использование графа соавторства для поиска опорного множества экспертов
```

```
journalrum_list VARCHAR2 DEFAULT NULL, --Список рубрик для сужения области поиска
department_list VARCHAR2 DEFAULT NULL, --Список подразделения для сужения опорного множества авторов
if_oracle NUMBER DEFAULT 1, --флаг использования внутренних механизмов контекстного поиска
if_translate NUMBER default null --флаг использования переводов
)
RETURN mas_man_find PIPELINED
```

Для создания интерфейса пользователя использован механизм построения динамических отчетных форм на основе SQL-запросов ИАС "ИС-ТИНА" [22]. В интерфейсе поиска можно указать требуемые ключевые слова, область знаний по рубрике ГРНТИ, подразделение для поиска опорного множества авторов и дополнительные параметры поиска (рис. 2).

После выполнения поиска пользователю на экран выводится ранжированный список ре-



Рис. 1. Схема алгоритма тематического поиска журналов

Ключевые слова (через запятую)

Область знаний

Подразделение

Встроенный поиск

Учет переводов

Рис. 2. Интерфейс тематического поиска журналов по ключевым словам

зультатов с указанием названия журнала "NAME" и степени его соответствия поисковому запросу "V". Для удобства навигации после нажатия на название журнала осуществляется переход в карточку журнала

с описанием его характеристик цитирования, списком последних статей и другой информацией (рис. 3).

Следует отметить, что данный алгоритм может применяться как самостоятельно, так и в сочета-

Show by items

V	NAME
35.566	Доклады Академии наук
11.419	Russian Chemical Bulletin
10.832	Журнал физической химии
9.337	Вестник Московского университета. Серия 3: Физика, астрономия
9.165	Russian Journal of Physical Chemistry A
8.737	Journal of Physics: Conference Series
8.255	Вестник Московского университета. Серия 2: Химия
8.152	Известия Академии наук. Серия химическая
8.138	Mendeleev Communications
7.623	Doklady Akademii nauk SSSR
7.612	Вестник Московского университета. Серия 4: Геология
6.912	Известия Российской академии наук. Серия физическая

Рис. 3. Интерфейс тематического поиска журналов по ключевым словам

нии с другими алгоритмами полнотекстового тематического анализа. Результатом работы алгоритма является ранжированный список журналов, соответствующий поисковому запросу пользователя.

Тематический поиск научно-технических конференций

Использование данных наукометрических систем позволяет создать поисковые механизмы тематического поиска регулярных научно-технических конференций с возможностью оценки их авторитетности. В качестве основной гипотезы используется объективная гипотеза, что авторитетные ученые и эксперты посещают авторитетные конференции по своей тематике. Автором публикации разработано три алгоритма тематического поиска конференций: поиск конференций, похожих по тематике на заданный журнал; поиск конференций, похожих на заданную конференцию, поиск конференций с использованием поиска экспертов.

Поиск конференций, похожих на заданный журнал, использует в качестве опорного множества авторов публикаций журнала. После этого выполняется расчет функции определения степени тематической связи между конференциями и журналами аналогично расчету функции определения степени тематической связи между журналами (1). Алгоритм, показанный на рис. 4, применим, если пользователь знает один или несколько журналов по своей тематике.

Основой программной реализации алгоритма является вычисляемая хранимыми в базе данных запросами таблица результатов расчета близости журналов и конференций:

```
JOURNAL_CONF_THEM_LINK
(CNT      NUMBER --Степень соответствия
JOURNAL_ID NUMBER --Уникальный ключ журнала
CONF_ID   NUMBER --Уникальный ключ конференции
)
```

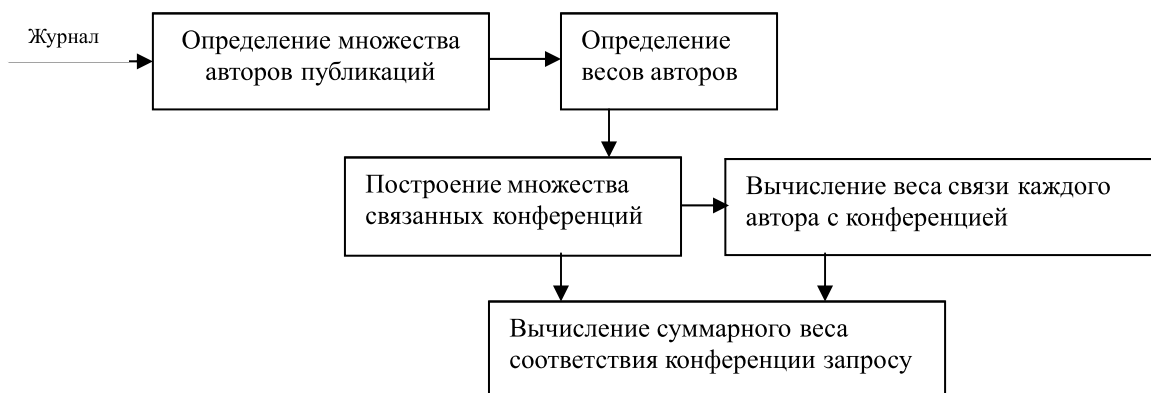


Рис. 4. Алгоритм поиска конференций по журналу, в котором публикуется автор

Использование такого подхода позволяет заранее рассчитать граф соответствия между журналами и конференциями для ускорения выполнения пользовательских запросов.

Поиск конференций, похожих на заданную конференцию, использует в качестве опорного множества докладчиков заданной конференции. После этого также проводится расчет функции F . Алгоритм, представленный на рис. 5, применим, если пользователь уже знает одну или несколько конференций по своей тематике и хочет расширить круг своего научного общения.

Основой программной реализации алгоритма является вычисляемая хранимыми в базе данных запросами таблица результатов расчета близости конференций:

```
CONF_THEM_LINK
(CNT      NUMBER --Степень соответствия
CONF_ID1 NUMBER --Уникальный ключ конференции
CONF_ID2 NUMBER --Уникальный ключ связанной
)
```

При поиске конференций с использованием методов и средств поиска экспертов на первом этапе обработки запроса, как и при тематическом поиске журналов, строится функция релевантности авторов поисковому запросу $v(a, q)$ и определяется множество авторов, релевантных запросу. После этого рассчитывается функция тематического соответствия F^* . Таким образом, алгоритм, приведенный на рис. 6, находит конференции, которые чаще всего посещают наиболее авторитетные специалисты по заданной в запросе тематике.

Для удобной визуализации результатов поиска разработан специальный интерфейс пользователя. На рис. 7 представлен пример отображения результата поиска конференций, тематически близких к журналу "Нейрокомпьютеры: разработка, применение".

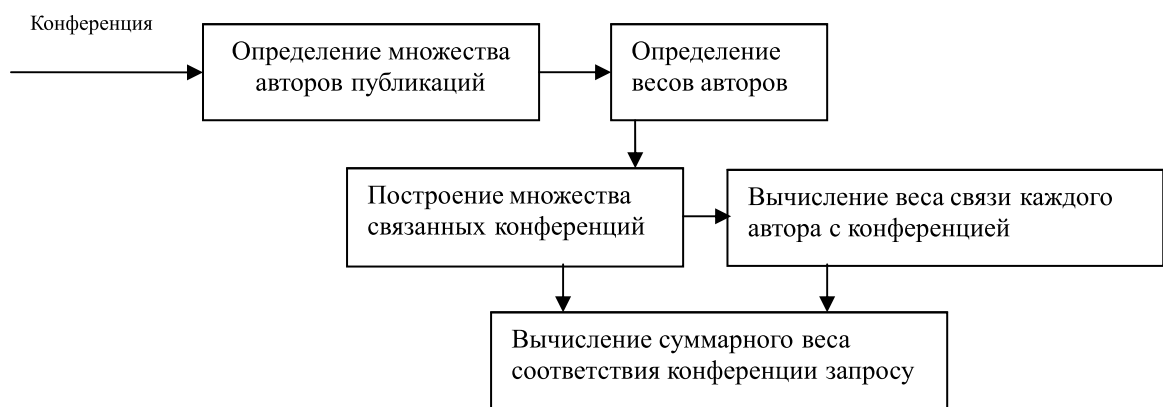


Рис. 5. Алгоритм поиска конференций по заранее выбранной пользователем конференции



Рис. 6. Алгоритм поиска конференций с использованием тематического поиска экспертов

В интерфейсе отображается название найденной конференции, степень соответствия, число загруженных в наукометрическую систему докладов с этой конференции и ссылки для дальнейшего поиска близких конференций и журналов, для осуществления которого можно нажать на

слова "конференции" и "журналы" в соответствующей строке. На рис. 8 представлены результаты перехода по одной из этих ссылок для поиска конференций, близких по тематике к конференции "XVIII Всероссийская научная конференция "Нейрокомпьютеры и их применение".

N	Конференция	Вес	Количество докладов	Похожие конференции	Похожие журналы
1	Ломоносовские чтения 2020. Секция философских наук(2020)	2,5	82	конференции	журналы
2	CAICS 2020: NATIONAL CONGRESS ON COGNITIVE RESEARCH, ARTIFICIAL INTELLIGENCE AND NEUROINFORMATICS(2020)	2,42	19	конференции	журналы
3	I Национальный Конгресс по когнитивным исследованиям, искусственному интеллекту и нейроинформатике(2020)	2,38	78	конференции	журналы
4	XVIII Всероссийская научная конференция "Нейрокомпьютеры и их применение". Москва. 17 марта 2020(2020)	2,08	4	конференции	журналы
5	61-е заседание международного междисциплинарного семинара "Нейрофилософия" Научного совета по методологии исследований искусственного интеллекта при Президиуме РАН(2021)	2	2	конференции	журналы
6	Нейронаука для медицины и психологии: XVII Международный междисциплинарный конгресс.(2021)	1,93	18	конференции	журналы
7	XII Международная научная конференция «Интеллектуальные системы и компьютерные науки»(2021)	1,92	77	конференции	журналы
8	Ломоносовские чтения - 2021, секция "Философские науки"(2021)	1,5	77	конференции	журналы
9	Международная междисциплинарная конференция «Искусственный интеллект в новой коммуникативной реальности» (2020)	1,25	17	конференции	журналы
10	Нейроинформатика 2021(2021)	1,17	3	конференции	журналы

Рис. 7. Результаты поиска для журнала "Нейрокомпьютеры: разработка, применение"

Список похожих конференций

Show by 10 items

Search:

N	Название конференции	Вес	Количество докладов	Похожие конференции	Похожие журналы
1	I Национальный Конгресс по когнитивным исследованиям, искусственному интеллекту и нейроинформатике(2020)	0,75	78	конференции	журналы
2	CAICS 2020: NATIONAL CONGRESS ON COGNITIVE RESEARCH, ARTIFICIAL INTELLIGENCE AND NEUROINFORMATICS(2020)	0,67	19	конференции	журналы
3	VI Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2020)(2020)	0,5	4	конференции	журналы
4	Нейронаука для медицины и психологии: XVII Международный междисциплинарный конгресс.(2021)	0,33	18	конференции	журналы

Items 1 to 4 shown. Total: 4

First Previous 1 Next Last

Рис. 8. Результаты поиска для конференции "XVIII Всероссийская научная конференция Нейрокомпьютеры и их применение"

Апробация алгоритмов проводилась на данных наукометрической системы ИАС "ИСТИНА". Для тестирования было сделано по пять поисковых запросов для каждой из трех программных реализаций представленных алгоритмов. Лучшие десять результатов для каждого запроса оценивались по следующей шкале: 2 — полное соответствие запросу; 1 — частичное соответствие запросу; 0 — несоответствие запросу. Интегральная оценка для каждой программной реализации рассчитывалась как сумма набранных баллов. Результаты тестирования представлены ниже.

Алгоритм	Точность, %
Поиск по журналу	81
Поиск по конференции	76
Использование поиска экспертов	89

Представленные результаты тестирования показывают, что наиболее точным способом поиска является использование алгоритма с тематическим поиском экспертов. Поиск по журналу имеет преимущество перед поиском похожих конференций вследствие более представительного опорного множества авторов.

Реализация подобных механизмов значительно упрощает для пользователей процесс информационного поиска конференций, соответствующих их информационным потребностям.

Заключение

Представленные в работе алгоритмы позволяют проводить тематический поиск журналов и конференций с использованием данных наукометрических систем. Представленные алгоритмы могут успешно применяться для поиска объектов на других языках. Применение этих алгоритмов возможно, как автономно, так и в сочетании с классическими алгоритмами полнотекстового тематического анализа. В отличие от полнотекстового поиска алгоритмы учитывают авторитетность конференций.

Использование представленных алгоритмов тематического анализа информации позволяет проводить нормализацию наукометрических показателей в зависимости от тематики научных исследований; проводить выделение и анализ основных тематических направлений развития науки; осуществлять тематический поиск объектов (журналов и конференций) в целях расширения научных контактов и создания новых научных коллективов.

1. **Интеллектуальная** система тематического исследования научно-технической информации (ИСТИНА) / В. А. Садовничий, В. А. Васенин и др. М.: Изд-во Московского ун-та, 2014. 262 с.
2. **Орлов А. И.** Наукометрия и управление научной деятельностью // Управление большими системами. Специальный выпуск 44: Наукометрия и экспертиза в управлении наукой. Институт проблем управления им. В. А. Трапезникова РАН. 2013. С. 538—568.
3. **Полянин А. Д.** Недостатки индексов цитируемости и Хирша. Индексы максимальной цитируемости. URL: http://eqworld.ipmnet.ru/ru/info/sci-edu/Polyanin_IndexH_2014.html
4. **Бричковский В. В.** Наукометрический анализ в информационном обеспечении инновационной деятельности // В мире науки. 2017. № 8 (174). С. 64—67.
5. **Афонин С. А., Козицын А. С., Шачнев Д. А.** Программные механизмы агрегации данных, основанные на онтологическом представлении структуры реляционной базы наукометрических данных // Программная инженерия. 2016. Том 7, № 9. С. 408—413.
6. **Witten I. H., Frank E., Hall M. A., Pal C. J.** Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016. 558 p.
7. **Васенин В. А., Занчурин М. А., Козицын А. С., Кривчиков М. А., Шачнев Д. А.** Архитектурно-технологические аспекты разработки и сопровождения больших информационно-аналитических систем в сфере науки и образования // Программная инженерия. 2017. Том 8, № 10. С. 448—455.
8. **Vasenin V. A., Zanchurin M. A., Kozitsyn A. S., Krivchikov M. A., Shachnev D. A.** Architectural and technological aspects of the cloud data analysis system development, case of istina system // CEUR Workshop Proceedings. 5. Sep. "APSSE 2017 — Proceedings of the 5th International Conference on Actual Problems of System and Software Engineering" 2017. P. 90—96.
9. **Анализ** статистических данных в системе ИСТИНА. URL: <https://istina.msu.ru/statistics/organization/214524/dynamic/>
10. **Васенин В. А., Афонин С. А., Козицын А. С.** Автоматизированная система тематического анализа информации // Информационные технологии. 2009. № 7. Приложение. С. 2—32.
11. **Садовничий В. А., Васенин В. А., Афонин С. А., Козицын А. С., Голомазов Д. Д.** Информационная система "ИСТИНА" как big data - инструментарий в области управления на основе анализа наукометрических данных // Знания — Онтологии — Теории (ЗОНТ-2015) Материалы Всероссийской конференции с международным участием. Новосибирск, 2015. С. 115—123.
12. **Васенин В. А., Зензинов А. А., Лунев К. В.** Использование наукометрических информационно-аналитических систем для автоматизации проведения конкурсных процедур на примере информационно-аналитической системы ИСТИНА // Программная инженерия. 2016. Том 7, № 10. С. 472—480.
13. **Shachnev D. A.** Searching for activity results and experts in a given subject area, taking results significance into account // Программная инженерия. 2021. Том 12, № 5. С. 260—266.
14. **Kozitsin A., Afonin S., Shachnev D.** Algorithm for searching experts in scientometric systems // CEUR Workshop Proceedings. Proceedings of the 23rd Conference on Scientific Services & Internet (SSI 2021). Vol. 3066. CEUR, 2022. P. 59—68.
15. **Плагиат** в научных статьях: трудности обнаружения перевода. URL: http://ai-news.ru/2018/01/plagiat_v_nauchnyh_statyah_trudnosti_obnaruzheniya_perevoda.html
16. **Афонин С. А., Голомазов Д. Д., Козицын А. С.** Использование систем семантического анализа для организации поиска научно-технической информации // Программная инженерия. 2012. № 2. С. 29—34.

17. Домнина Т. Н., Хачко О. А. Научные журналы: количество, темпы роста // XIX Научно-практический семинар с международным участием "Информационное обеспечение науки: новые технологии". 2015. С. 83–96.

18. De Solla Price D. D. Little Science, Big Science. New York, 1963. 109 p.

19. Козицын А. С., Афонин С. А., Шачнев Д. А. Метод оценки тематической близости научных журналов // Программная инженерия. 2020. Том 11, № 6. С. 335–341.

20. Краснов Ф. В. Анализ методов построения графа соавторства: подход на основе двудольного графа // International Journal of Open Information Technologies. 2018. Vol. 6, № 2. P. 31–37.

21. Статистика по наличию журналов РИНЦ. URL: https://www.elibrary.ru/project_free_access.asp

22. Afonin S., Kozitsyn A., Astapov I. Sqlreports: Yet another relational database reporting system // Proceedings of the 9th International Conference on Software Engineering and Applications. 2014. P. 529–534.

Thematic Search Algorithms for Experts, Journals and Conferences

A. S. Kozitsyn, alexanderkz@mail.ru, Lomonosov Moscow State University, Moscow, 119192, Russian Federation

Corresponding author:

Kozitsyn Alexander S., Leading Researcher, Lomonosov Moscow State University, 119192, Moscow, Russian Federation

E-mail: alexanderkz@mail.ru

Received on April 02, 2022

Accepted on April 14, 2022

The paper discusses methods for thematic search of experts, journals and conferences, provides a brief overview of existing systems with an indication of their main shortcomings, and describes the algorithms for the thematic search of experts developed by the authors, as well as algorithms for thematic search of journals and conferences. A brief description of the software implementation of the developed algorithms is given. User interfaces for searching for conferences are presented. The results of testing software implementations of algorithms based on the data of the scientometric system "ISTINA" are presented. The algorithms developed by the authors use data from scientometric systems, including graphs of mutual relations between objects, are able to filter data by narrowly focused thematic queries and rank search results by authority. The approach used makes it possible to perform a thematic search for conferences in different languages with the determination of their authority and the determination of their thematic proximity. Such thematic analysis makes it possible to normalize scientometric indicators depending on the subject of scientific research, to identify and analyze the main thematic directions in the development of science, to carry out a thematic search for objects (journals and conferences) in order to expand scientific contacts and create new research teams.

Keywords: thematic analysis, scientometrics, information systems, bibliography, author, graph

For citation:

Kozitsyn A. S. Thematic Search Algorithms for Experts, Journals and Conferences, *Programmnaya Ingeneria*, 2022, vol. 13, no. 6, pp. 291–300.

DOI: 10.17587/prin.13.291-300

References

1. *Intelligent system of case study of scientific and technical information (ISTINA)* / V. A. Sadovnichy, V. A. Vasenin et al. Moscow, Moscow University Press, 2014. 262 p. (in Russian).

2. **Orlov A. I.** Scientometrics and management of scientific activities, *Management of large systems. Special Issue 44: Scientometrics and Expertise in Science Management*, Institute of Management Problems V. A. Trapeznikova RAS, 2013, pp. 538–568 (in Russian).

3. **Polljanin A.** Disadvantages of citation and hirsch indices. Maximum Citation Indices, available at: http://eqworld.ipmnet.ru/ru/info/sci-edu/polyanin_indexh_2014.html (in Russian).

4. **Brichkovsky V. V.** Scientometric analysis in the information support of innovative activity, *V mire nauki*, 2017, no. 8 (174), pp. 64–67 (in Russian).

5. **Afonin S. A., Kozitsyn A. S., Shachnev D. A.** Software data aggregation mechanisms based on the ontological representation of the structure of the relational database of scientometric data, *Programmnaya Ingeneria*, 2016, vol. 7, no. 9. pp. 408–413 (in Russian).

6. **Witten I. H., Frank E., Hall M. A., Pal C. J.** *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, 2016, 558 p.
7. **Vasenin V. A., Zanchurin M. A., Kozitsyn A. S., Krivchikov M. A., Shachnev D. A.** Architectural and Technological Aspects of the Development and Maintenance of Large Information Analysis Systems in the Area of Science and Education, *Programmnaya Ingeneria*, 2017, vol. 8. no. 10, pp. 448–455 (in Russian).
8. **Vasenin V. A., Zanchurin M. A., Kozitsyn A. S., Krivchikov M. A., Shachnev D. A.** Architectural and technological aspects of the cloud data analysis system development, case of istina system, *CEUR Workshop Proceedings. 5. Ser. "APSSE 2017 — Proceedings of the 5th International Conference on Actual Problems of System and Software Engineering"* 2017, pp. 90–96.
9. **Analysis** of statistical data in the system ISTINA, available at: <http://gis-lab.info/qa/srtm.html> (in Russian).
10. **Vasenin V. A., Afonin S. A., Kozitsyn A. S.** Avtomatizirovannaya sistema tematicheskogo analiza informatsii, *Informatsionnye tekhnologii*, 2009, no. 7, supl., pp. 2–32 (in Russian).
11. **Sadovnichy V. A., Vasenin V. A., Afonin S. A., Kozitsyn A. S., Golomazov D. D.** Information system "ISTINA" as big data — a tool in the field of control based on the analysis of scientometric data, *Knowledge — Ontologies — Theories (ZONT-2015) Materials of the All-Russian Conference with international participation*, Novosibirsk, 2015, pp. 115–123 (in Russian).
12. **Vasenin V. A., Zenzinov A. A., Lunev K. V.** Using scientometric information-analytical systems to automate competitive procedures using the example of the information-analytical system ISTINA, *Programmnaya Ingeneria*, 2016, vol. 7, no. 10, pp. 472–480 (in Russian).
13. **Shachnev D. A.** Searching for activity results and experts in a given subject area, taking results significance into account, *Programmnaya Ingeneria*, 2021, vol. 12, no. 5, pp. 260–266.
14. **Kozitsyn A., Afonin S., Shachnev D.** Algorithm for searching experts in scientometric systems, *CEUR Workshop Proceedings. Proceedings of the 23rd Conference on Scientific Services & Internet (SSI 2021)*, vol. 3066, CEUR, 2022, pp. 59–68.
15. **Plagiat** v nauchnykh statiyakh: trudnosti obnaruzheniya perevoda, available at: http://ai-news.ru/2018/01/plagiat_v_nauchnyh_statyah_trudnosti_obnaruzheniya_perevoda.html (in Russian).
16. **Afonin S. A., Golomazov D. D., Kozitsyn A. S.** The use of semantic analysis systems to organize the search for scientific and technical information, *Programmnaya Ingeneria*, 2012, no. 2, pp. 29–34 (in Russian).
17. **Domnina T. N., Khachko O. A.** Nauchnye zhurnaly: kolichestvo, tempy rosta, *XIX Nauchno-prakticheskii seminar s mezhdunarodnym uchastiem "Informatsionnoe obespechenie nauki: novye tekhnologii"*, 2015, pp. 83–96 (in Russian).
18. **De Solla Price D. D.** Little Science, Big Science. New York, 1963, 109 p.
19. **Kozitsyn A. S., Afonin S. A., Shachnev D. A.** Metod otsenki tematicheskoi blizosti nauchnykh zhurnalov, *Programmnaya Ingeneria*, 2020, vol. 11, no. 6, pp.335–341 (in Russian).
20. **Krasnov F. V.** Analysis of co-authorship graph construction methods: bipartite graph approach, *International Journal of Open Information Technologies*, 2018, vol. 6, no. 2, pp. 31–37 (in Russian).
21. **Statistika** po nalichiiu zhurnalov RINTS, available at: https://www.elibrary.ru/project_free_access.asp (in Russian).
22. **Afonin S., Kozitsyn A., Astapov I.** Sqlreports: Yet another relational database reporting system, *Proceedings of the 9th International Conference on Software Engineering and Applications*, 2014, pp. 529–534.

ИНФОРМАЦИЯ

В рамках Международного конгресса "**Суперкомпьютерные дни в России**" **Нижегородский государственный университет им. Н. И. Лобачевского** проводит международную конференцию и молодежную школу "Математическое моделирование и суперкомпьютерные технологии". Проведение конференции (14—15 ноября 2022 г.) и молодежной школы (16—18 ноября 2022 г.) будет содействовать дальнейшему повышению синергетического эффекта от взаимовыгодного взаимодействия двух приоритетных направлений развития науки и техники – математического моделирования и суперкомпьютерных технологий.

В программе конференции – лекции приглашенных экспертов, доклады известных ученых и специалистов, сообщения молодых исследователей. Школа будет включать мастер-классы и практикумы по применению актуальных инструментов и технологий.

Конференция и молодежная школа проводятся под эгидой **Суперкомпьютерного консорциума университетов России** при партнерстве с Национальным центром НТИ «**Технологии машинного обучения и когнитивные технологии**» (СПбГУ ИТМО) и Национальным центром НТИ «**Технологии хранения и анализа больших данных**» (МГУ).

Подробности: <http://agora.guru.ru/hpc2022/>

Decision Support System based on Application of the Second Generation Neural Network¹

E. V. Palchevsky, Lecturer, teelxp@inbox.ru, Financial University under the Government of the Russian Federation, Moscow, 125167, Russian Federation,

V. V. Antonov, Professor, antonov.v@bashkortostan.ru, Ufa State Aviation Technical University, Ufa, 450008, Russian Federation

Corresponding author:

Palchevsky Evgeniy V., Lecturer, Financial University under the Government of the Russian Federation, Moscow, 125167, Russian Federation

E-mail: teelxp@inbox.ru

Received on October 28, 2021

Accepted on April 25, 2022

The issue of the feasibility of using existing statistical and hydrological methods for short-term and early forecasting in the framework of forecasting the levels of water rise in water bodies is considered: a comparative review is given, which describes their advantages and disadvantages. In the course of analyzing the shortcomings of these methods, the problem of operational and early (advance) forecasting of water rise levels was identified. To solve this problem, a decision support system is proposed for predicting the water rise levels in advance, based on a neural network (intelligent) analysis of retrospective data (date, water level, air temperature, atmospheric pressure and wind speed) to calculate the water level values for 5 days in advance. The artificial neural network itself is based on the freely distributed library of machine learning programs "TensorFlow", and a modified backpropagation method is used as training, the main difference of which is an increase in the learning rate of an artificial neural network. The results of the analysis of the effectiveness showed that the proposed decision support system is more accurate (the error between the real and calculated values does not exceed 2.10 %), compared to existing common methods/systems (8.36 %). This will allow to give the necessary time to special services for the implementation of flood control measures to prepare for the protection of technical facilities of enterprises.

Keywords: forecasting, intellectual analysis, neural network analysis, retrospective data, data mining, forecasting water levels

Acknowledgements: *The study is conducted with financial support from the Ministry of Education and Science of the Russian Federation as part of the basic part of the state assignment to higher education educational institutions # FEUE-2020-0007.*

For citation:

Palchevsky E. V., Antonov V. V. Decision Support System based on Application of the Second Generation Neural Network, *Programmnaya Ingeneria*, 2022, vol. 13, no. 6, pp. 301–308.

УДК 004.032.26

Е. В. Пальчевский, препод., teelxp@inbox.ru, Финансовый университет при Правительстве Российской Федерации, Москва,

В. В. Антонов, д-р техн. наук, проф., antonov.v@bashkortostan.ru, Уфимский государственный авиационный технический университет

Система поддержки принятия решений на основе применения нейронной сети второго поколения

Рассмотрен вопрос о целесообразности использования существующих статистических и гидрологических методов краткосрочного и раннего прогнозирования в рамках прогнозирования уровней подъема воды в водных объектах — дан сравнительный обзор, в котором описаны

¹ The article was based on the materials of the report at the IX All-Russian Scientific Conference "Information Technologies for Intelligent Decision Support" (ITIDS'2021).

их преимущества и недостатки. В ходе анализа недостатков этих методов была выявлена проблема оперативного и раннего (заблаговременного) прогнозирования уровней подъема воды. Для решения этой задачи предлагается система поддержки принятия решений для заблаговременного прогнозирования уровней подъема воды, основанная на нейросетевом (интеллектуальном) анализе ретроспективных данных. Приведены результаты анализа эффективности предлагаемой системы поддержки принятия решений.

Ключевые слова: прогнозирование, интеллектуальный анализ, нейросетевой анализ, ретроспективные данные, интеллектуальный анализ данных, прогнозирование уровня воды

Исследования выполнены при поддержке Министерства науки и высшего образования РФ в рамках выполнения Государственного задания № FEUE-2020-0007.

Introduction

At present, various natural phenomena (in this article, we will take the level of water rise in water bodies as an example) are one of the most frequent and dangerous causes of external impact on technical objects of complex distributed systems (CDS). In this article, complex distributed systems mean an interconnected set of objects of varying complexity and physical nature (for example, water bodies of a region), and technical objects of complex distributed systems mean various potentially hazardous objects, oil and gas pipelines, while, according to [1], technical objects belong to the class of unique complex systems. For these technical objects, as a rule, it is necessary to carry out special (in our case, flood control) measures to minimize social and financial risks: for example, it is necessary to predict in advance the water levels at stationary hydrological and automatic measurement and observation posts.

Thus, based on these problems, there is a need for a sufficiently fast and early forecasting of water level values based on a decision support system based on neural network data analysis, which will allow special services to give the necessary time to implement flood control measures to prepare for the protection of technical facilities of enterprises.

1. Related Works

Statistical methods, for example, generalized regression models [2–4], the method of least squares [5], and numerical methods [6, 7], are widely used in predicting water level values based on retrospective data. But at the same time, the problem of forecasting in advance arises, since the main purpose of these methods is short-term forecasting of water level values, which is often insufficient for early implementation of flood control measures. If we consider less common methods (hydrological [8–10]) of short-term and early forecasting of water levels, then it is necessary

to note the problem of resource intensity: the duration of calculations of water level values is not permissible, especially in critical (peak) moments of flood situations. Many scientists (both domestic and foreign) are engaged in these problems, but due to the insufficient number of works describing methods for early (early) forecasting of water levels based on intelligent (neural network) analysis of retrospective data using an artificial neural network (ANN) for Better forecasting of water levels in the shortest possible time and using outdated architectures of artificial neural networks (for example, multilayer perceptron [11]), the implementation of the proposed method within the framework of early warning of a flood situation becomes relevant.

2. Formalized statement of the problem on the example of the levels of water rise in water bodies

One of the main parameters of the impact on various technical objects is the flood, which is expressed in the rise in the values of h — water levels on water bodies. As an example, we can take the stationary hydrological posts of the Republic of Bashkortostan. In this regard, we introduce the following notation: h_{ji}^k — is the value of the water level measured at the k -th gauging station on the i -th date of the j -th year. Here $k = 1, \dots, n$, where n is the number of gauging stations involved in the calculations, j is the number of the year, i is the specific date of measurement. The task of early forecasting is (it should be noted that the task of short-term forecasting for 1 day was solved in [15], and on the basis of the methodology proposed in [15], forecasting for 2–5 days is performed) in the fact that in a specific current i -th day of measurement, calculate the value of the water level for $i + 2$ days, that is h_{ji+2}^k , as well as for $i + 3$ (h_{ji+3}^k), $i + 4$ (h_{ji+4}^k) and $i + 5$ (h_{ji+5}^k) for any $k = 1, \dots, n$.

3. Application of an artificial neural network for early prediction of water levels at stationary hydropost

To solve this problem, it is proposed to use the results of previous measurements of water levels (an array of large retrospective data) at all stationary hydrological stations located on the territory under consideration for all previous years. The proposed method of early forecasting is based on the use of an artificial neural network (fig. 1), implemented using the free library of machine learning programs "TensorFlow" [12]. The method of training ANN backpropagation (backpropagation of the error), changed by the authors of the article, is integrated into this neural network via the API. As shown by the results of the experiment presented in Section 4, the accuracy and time of calculating the forecast prove the feasibility of using this method in the framework of the advance forecasting of water level values at stationary hydrological posts. Thus, the forecasting process is proposed to be carried out in three stages (fig. 2): data preprocessing, forecasting model formation and data post-processing.

At the first stage, which is performed, as a rule, before the development of a flood situation, the ANN parameters are selected: the total number of neurons (107); the number of neurons in the input (h_{ji}^k (6)) and output (h_{ji}^k (5)) layers; the number of intermediate layers (3) and neurons in them (H_{jk}^k (96)); the activation

function on hidden layers is sigmoidal, while a linear activation function is used on the output layer due to its weighted sum on each neuron of the output layer [13–15]; the amount of data for the samples: training (train, 60 %), validation (validation, 20 %) and checking (test, 20 %); number of epochs (200); learning rate α (0.90), selected experimentally with further training of the artificial neural network. But it is worth noting that when applying the backpropagation method, it is necessary to normalize the input values of the water levels H to prevent vanishing and exploding gradients:

$$\bar{h}_{ji}^k = \frac{\overline{h_{ji}^k} - \overline{h_{\min}^k}}{\overline{h_{\max}^k} - \overline{h_{\min}^k}} \quad (1)$$

where $\overline{h_{\max}^k}$ and $\overline{h_{\min}^k}$ are the minimum and maximum values of water levels for all data supplied to the input layer of the neural network for all h_{ji}^k , where $k = 1, \dots, n$.

To train the neural network (fig. 3), the backpropagation method, modified by the authors of the article, is used. The essence of the change lies in the application of an experimentally set parameter (coefficient) of the neural network learning rate.

A. Calculation of the squared error

As a rule, the inputs of the network are considered as an input vector ($\overline{h_{ji}^k}$), where $\overline{h_{ji}^k} = [\overline{h_{ji1}^k}, \overline{h_{ji2}^k}, \dots, \overline{h_{jiK}^k}]$, and the outputs of the network can be represented as

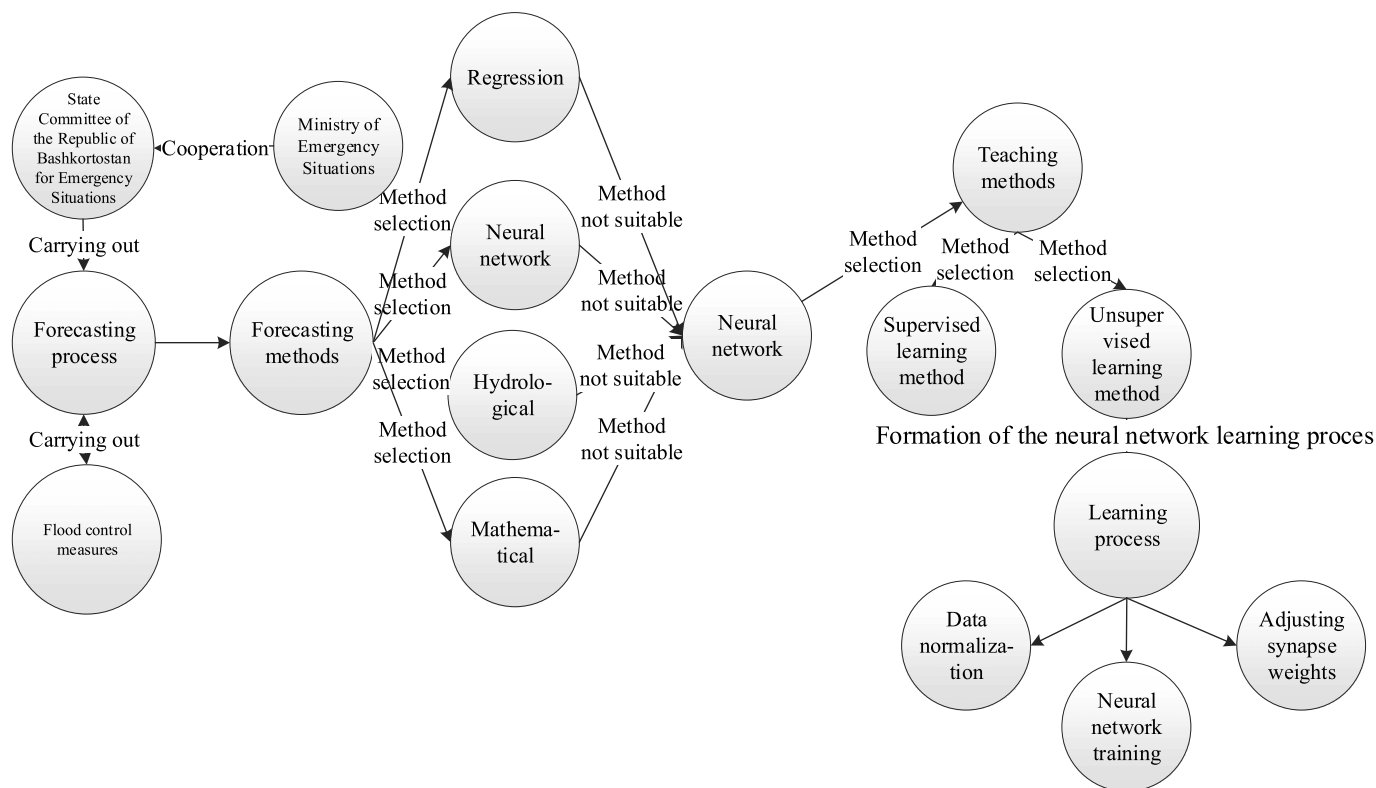


Fig. 1. Ontological model of the water level forecasting process

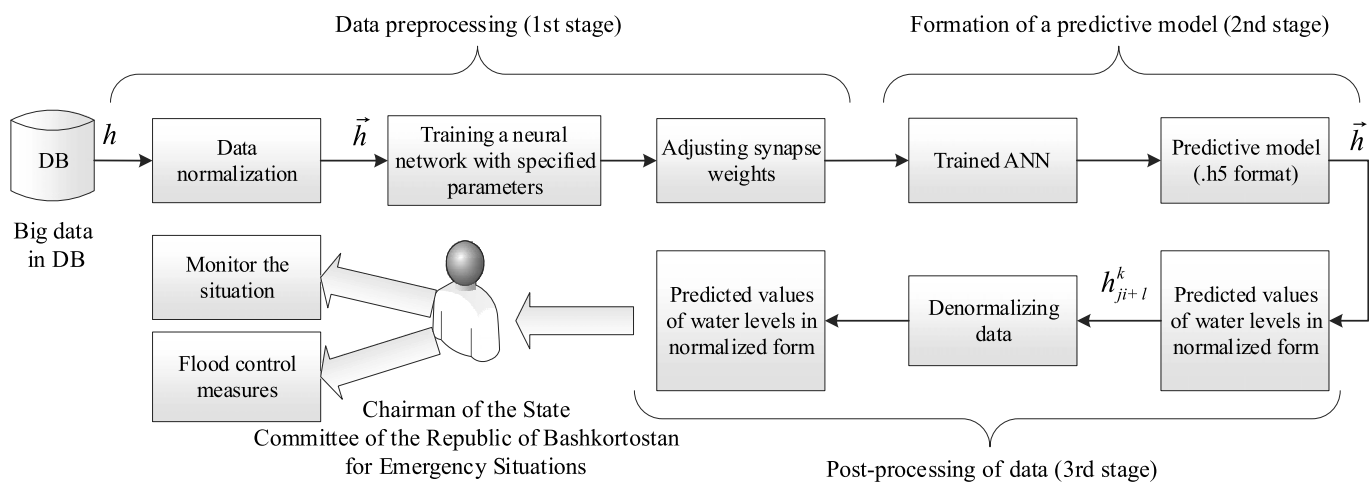


Fig. 2. Scheme of using a neural network to predict the values of water levels

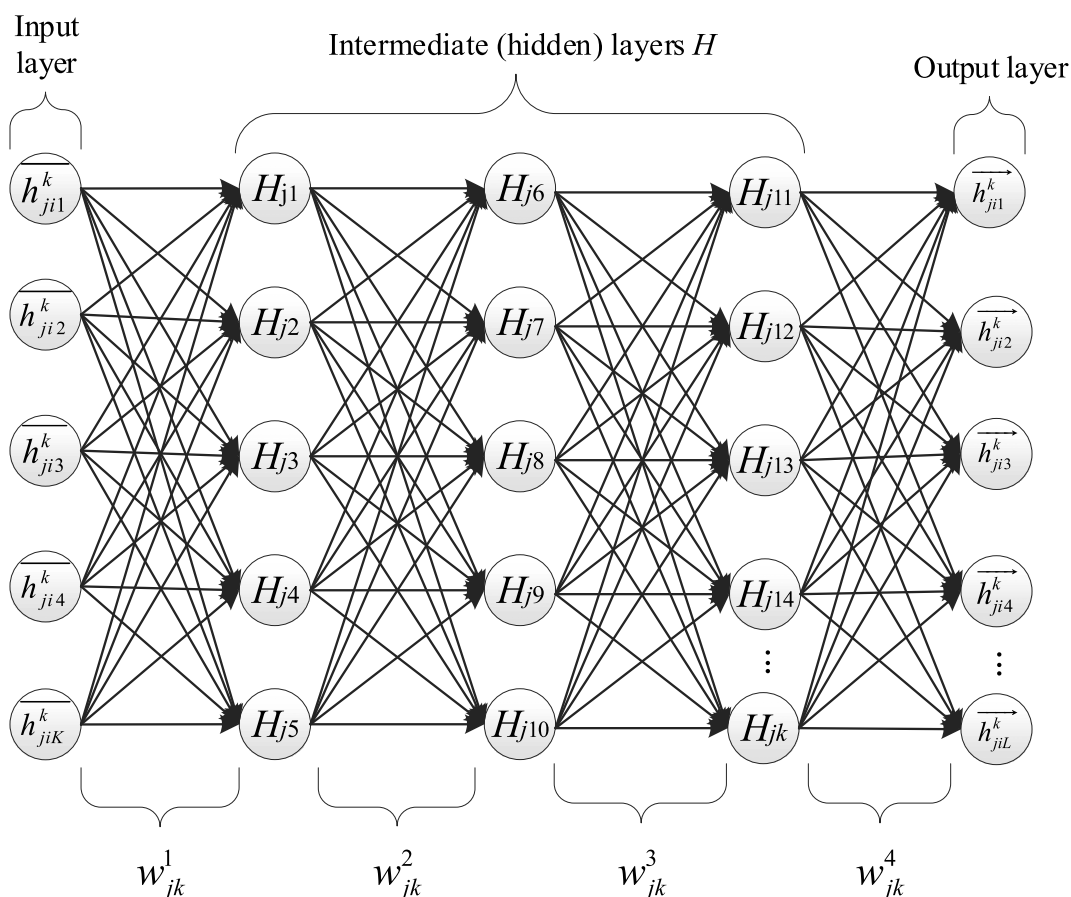


Fig. 3. Neural network structure

an output vector $\overline{h_{ji}^k}$, where $\overline{h_{ji}^k} = [\overline{h_{ji1}^k}, \overline{h_{ji2}^k}, \dots, \overline{h_{jiL}^k}]$, where, in this case $K = 1, \dots, M$ and $L = 1, \dots, N$, in which M and N represent the number of values input and output vectors. Accordingly, the training sample

is a set of pairs R of input vectors $\overline{h_{jiK}^k}$ and the desired output vectors: $\overline{h_{jiL}^k}$:

$$R = \left\{ \left(\overline{h_{ji1}^k}, \overline{h_{ji1}^k} \right), \left(\overline{h_{ji2}^k}, \overline{h_{ji2}^k} \right), \dots, \left(\overline{h_{jiK}^k}, \overline{h_{jiL}^k} \right) \right\}. \quad (2)$$

Each time $\overline{h_{jiK}^k}$ is applied from R to the neural network to the neural network, the actual output $\overline{h_{jiL}^k}$ of the output layer will be calculated:

$$\overline{h_{jiL}^k} = f(\sigma_L), \quad (3)$$

where σ is the activation function, is determined by the relation

$$\sigma(\overline{h_{jiK}^k}) = \frac{1}{1 + e^{-\overline{h_{jiK}^k}}}, \quad (4)$$

and is the weighted sum of the outputs of neuron j in each hidden layer H_j :

$$\sigma_L = \sum_{j=1}^l w_{jk}^j H_{jk}^j, \quad (5)$$

where w is the weight of the synapse, k is the k -th neuron in the intermediate layer. Based on this, we can determine the squared error for each pair of vectors of the set R by summing the squared errors in each output neuron:

$$E_l = \frac{1}{2} \sum_{k'=1}^L (\overline{h_{jiK_{k'}}^k} - \overline{h_{jiL_{k'}}^k})^2; \quad (6)$$

and, as a consequence, the total squared error E by summing all pairs of inputs and outputs in the training set:

$$E = \frac{1}{2} \sum_{l'=1}^K \sum_{k'=1}^L (\overline{h_{jiK_{k'l'}}^k} - \overline{h_{jiL_{k'l'}}^k})^2. \quad (7)$$

Thus, the goal of training is to minimize E by finding an appropriate set of weights w_{xk}^l and w_{jk}^l , where l varies from 1 to 4.

B. Determination of partial derivatives by synapse weights

Based on relation (6), we can note that

$$\frac{dE}{dw_{jk}^l} = \overline{h_{jiK}^k} - \overline{h_{jiL}^k}, \quad (8)$$

in this case, relying on (5), we obtain

$$\frac{d\sigma_k}{dw_{jk}^l} = H_j. \quad (9)$$

Based on (9), we determine the partial derivative E with respect to the weight w_{jk} to perform gradient descent in (10) for the weights of the synapses between the intermediate and output layers:

$$\frac{dE}{dw_{jk}^l} = ((\overline{h_{jiK}^k} - \overline{h_{jiL}^k}) * \overline{h_{jiL}^k}) \cdot ((\sigma(\overline{h_{jiK}^k}) - \overline{h_{jiL}^k}) \cdot H_{jk}). \quad (10)$$

Moreover, if we consider the case between the input and hidden (intermediate) layers, then it is worth highlighting that in this case all outputs depend on w_{xk} . Thus, the partial derivative in this case will be found as follows:

$$\frac{dE}{dw_{xk}^l} = \sum_{k'=1}^L ((\overline{h_{jiK_{k'}}^k} - \overline{h_{jiL_{k'}}^k}) * \overline{h_{jiL_{k'}}^k}) \times ((\sigma(\overline{h_{jiK}^k}) - \overline{h_{jiL_{k'}}^k}) \cdot (w_{jk}^l \cdot H_{jk}^l)) \cdot ((1 - H_{jk}^l) \cdot \overline{h_{jiK}^k}). \quad (11)$$

Thus, equations (9) and (10) give all the necessary values of $\frac{dE}{dw}$ for applying (10) and (11) in the framework of gradient descent for all weights of the neural network.

C. Change in synapse weights

Each weight will be changed to dw to decrease E :

$$w(t+1) = w(t) + \Delta w(t), \quad (12)$$

where $\Delta w(t) = -\frac{dE}{dw}|_t$, $w(t)$ — is the weight of the synapse at time t , $w(t+1)$ is the modified (updated) weight of the synapse. And to increase the learning rate of the neural network, the previously indicated parameter α is added:

$$\Delta w(t) = -\frac{dE}{dw}|_t + \alpha \cdot \Delta w(t-1). \quad (13)$$

At the second stage, the forecast model is formed directly, and at the third stage, the water level values are forecast. But since all the values were previously normalized ($\overline{h_{ji}^k} \in [0; 1]$), then for their further use within the framework of flood control measures, it is necessary to denormalize $\overline{h_{ji+1}^k}$ (14) according to the inverse relation (1):

$$\overline{h_{ji}^k} = \overline{h_{ji}^k} \cdot (\overline{h_{\max}} - \overline{h_{\min}}) + \overline{h_{\min}}. \quad (14)$$

4. Analysis of the efficiency of the proposed method in the framework of forecasting the levels of water rise

To analyze the effectiveness of the proposed method for predicting water levels, we used long-term data on water level measurements at hydrological posts provided by the Bashkir Department for Hydrometeorology and Environmental Monitoring (Bashhydromet) from 01.01.2000 to 02.05.2021 in the form: stationary hydrological code station/automatic station, date, water level at the stationary hydrological station/automatic station.

During the experiment, many iterations were carried out for the entire period of long-term observations (from 01.01.2000 to 06.05.2021) to calculate the predicted values of water levels. The total array of data used in the experiment is 1,025,815 values, of which 60 % are data of long-term observations in the period from 01.01.2000 to 31.12.2013, fed to the training sample of the neural network, to the validation sample — 20 % (01.01.2014 — 31.12.2017) and for the verification — 20 % (01.01.2018 — 06.05.2021), while the forecasting depth, on the basis of which the predicted value is calculated, is set experimentally depending on the stationary hydrological post.

The article, as an example, presents the results of an experiment, predicting water levels in the stationary

hydrological city of Ufa (76289, Belaya River) during the spring flood of 2021 in the Republic of Bashkortostan. The first stage consisted in comparing the real and predicted values of water levels at hydrological stations in different periods of the passage of the flood. The results are shown in Table, where the real water level is the actual measured value at the stationary hydrological station and automatic station, and the predicted water level is the water level value obtained using the neural network. It is important to note that comparison with other forecasting methods (see fig. 1) was made in [14], according to the results of which neural network forecasting showed a better result (the output values turned out to be more accurate than when calculating the predicted values of water levels by other methods).

Forecast values of water levels at stationary hydrological station (76289) Ufa obtained based on an artificial neural network

Forecast date	Forecast for 5 days, cm	Forecast for 4 days, cm	Forecast for 3 days, cm	Forecast for 2 days, cm	Forecast for 1 day, cm	Actual water level for the forecast date, cm
07.04.2021	71	109	294	344	364	Forecast for 12.04.21: 297
08.04.2021	114	311	329	311	365	Forecast for 13.04.21: 416
09.04.2021	338	396	328	376	477	Forecast for 14.04.21: 481
10.04.2021	353	419	403	531	514	Forecast for 15.04.21: 516
11.04.2021	432	372	552	523	567	Forecast for 16.04.21: 550
12.04.2021	455	582	603	537	556	Forecast for 17.04.21: 586
13.04.2021	581	565	589	631	608	Forecast for 18.04.21: 607
14.04.2021	613	606	596	615	655	Forecast for 19.04.21: 626
15.04.2021	635	636	634	619	636	Forecast for 20.04.21: 636
16.04.2021	635	654	660	655	637	Forecast for 21.04.21: 637
17.04.2021	626	649	673	660	669	Forecast for 22.04.21: 628
18.04.2021	607	635	663	688	689	Forecast for 23.04.21: 607

19.04.2021	574	610	643	673	695	Forecast for 24.04.21: 587
20.04.2021	551	568	612	649	677	Forecast for 25.04.21: 565
21.04.2021	532	540	562	613	649	Forecast for 26.04.21: 526
22.04.2021	496	516	528	557	609	Forecast for 27.04.21: 475
23.04.2021	466	470	501	518	548	Forecast for 28.04.21: 432
24.04.2021	507	489	443	404	401	Forecast for 29.04.21: 387
25.04.2021	476	423	375	368	366	Forecast for 30.04.21: 318
26.04.2021	405	351	339	332	294	Forecast for 01.05.21: 255
27.04.2021	332	315	306	260	238	Forecast for 02.05.21: 232
28.04.2021	297	282	236	208	240	Forecast for 03.05.21: 189
29.04.2021	265	212	187	214	200	Forecast for 04.05.21: 179
30.04.2021	198	166	195	180	155	Forecast for 05.05.21: 161
01.05.2021	154	177	150	165	152	Forecast for 06.05.21: 141
Average error, cm	12,84	0,76	14,68	23,88	37,08	—

Every day, since the start of monitoring the development of the flood situation (in 2021 it is April 7, but the forecast was made for 1–5 days, respectively, the date of the beginning of the water rise level during the spring flood can be considered 12.04.2021, that is, $i = 1$), using a neural network, a forecast was made for the next m (for example, 1–5) days at 1 gauging station (that is, $n = 1$; this value of n was taken to reduce the experiment time). The next day, the actual value of water levels was measured at the same posts $h_{j,i+m}^k$. Forecasting daily until May 01 (respectively, we introduce the additional designation l – the number of flood days, that is, the value of $l_{k,i+m} = 25$) for each post, we obtain 25 forecasts and actual values of water

levels for each of the n posts, which makes it possible to determine the average forecast accuracy for each k -th post:

$$E_{cp}^k = \frac{1}{l_{k,i+m}} \sum_{i=1}^{l_{k,i+m}} E |h_{ji}^k - h_{j,i+m}^k|.$$

In this case, the error E (in centimeters) on the i -th day of the j -th year of each k -th gauging station is calculated by the ratio of the absolute difference:

$$E_{ji}^k = |h_{ji}^k - h_{j,i+m}^k|.$$

An important difference between the proposed method for predicting water levels using an artificial

neural network in comparison with other known methods is the speed of obtaining a forecast and its correctness (more accurate) with advance forecasting (in this case, by 5 days), the results are shown in fig. 4, see the 4th side of cover. According to Table and fig. 4, it can be seen that the artificial neural network gives fairly accurate results: the average error does not exceed 37.08 cm for the entire flood period. Thus, the implemented artificial neural network showed a more accurate and stable result, which proves the feasibility of using the proposed method in the framework of early forecasting of a flood situation.

Conclusion

This method was carried out at a high level of abstraction without reference to physical objects (this allows us to reduce the construction of a decision support system to a formal algorithm), which opens up the possibility of using it in a wide range of areas and tasks, including forecasting various natural case — water levels on water bodies) phenomena, man-made disasters, economic phenomena, etc. The analysis of efficiency based on the proposed method showed that the use of an artificial neural network showed more stable results when predicting water level values using the example of a spring flood in the Republic of Bashkortostan, which will give the necessary time for special services to implement flood control measures to prepare for the protection of technical facilities.

Thus, performing a sequence of data normalization, training a neural network with specified parameters, and adjusting the synapse weights, we obtain a method for constructing a decision support system for a wide range of tasks.

References

1. Reimers N. F. Nature management, Dictionary-reference book, 1990, 637 p.
2. Supriya P., Krishnaveni M., Subbulaksmi M. Regression Analysis of Annual Maximum Daily Rainfall and Stream Flow for

Flood Forecasting in Vellar River Basin, *Aquatic Procedia*, 2015, vol. 4, pp. 957–963.

3. He Y., Yan Y., Wang X., Wang C. Uncertainty Forecasting for Streamflow based on Support Vector Regression Method with Fuzzy Information Granulation, *Energy Procedia*, 2019, vol. 158, pp. 6189–6194.

4. Luo X., Yuan X., Zhu S., Xu Z., Meng L., Pengm J. A hybrid support vector regression framework for streamflow forecast, *Journal of Hydrology*, 2019, vol. 568, pp. 184–193.

5. Strupczewski W. G., Kaczmarek Z. Non-stationary approach to at-site flood frequency modelling II Weighted least squares estimation, *Journal of Hydrology*, 2001, vol. 248, no. 1, pp. 143–151.

6. Silvestro F., Rossi L., Campo L., Parodi A., Fiori E., Rudari R., Ferraris L. Impact-based flash-flood forecasting system: Sensitivity to high resolution numerical weather prediction systems and soil moisture, *Journal of Hydrology*, 2019, vol. 572, pp. 388–402.

7. Hu R., Fang F., Salinas P., Pain C. C., Domingo N. D., Mark O. Numerical simulation of floods from multiple sources using an adaptive anisotropic unstructured mesh method, *Advances in Water Resources*, 2019, vol. 123, pp. 173–188.

8. Tian J., Liu J., Yan D., Ding L., Li C. Ensemble flood forecasting based on a coupled atmospheric-hydrological modeling system with data assimilation, *Atmospheric Research*, 2019, vol. 224, pp. 127–137.

9. Gomez M., Sharma S., Reed S., Mejia A. Skill of ensemble flood inundation forecasts at short- to medium-range timescales, *Journal of Hydrology*, 2019, vol. 568, pp. 207–220.

10. Lopes V. A., Fan F. M., Pontes P. R., Siqueira V. A., Collichonn W., Marques D. M. A first integrated modelling of a river-lagoon large-scale hydrological system for forecasting purposes, *Journal of Hydrology*, 2048, vol. 565, pp. 177–196.

11. Velasco L., Serqunia R., Zamad M., Juanico B., Lomoso J. Week-ahead Rainfall Forecasting Using Multilayer Perceptron Neural Network, *Procedia Computer Science*, 2019, vol. 161, pp. 386–397.

12. TensorFlow, available at: <https://www.tensorflow.org>

13. Kratzert F., Klotz D., Shalev G., Klambauer G., Hochreiter S., Nearing G. Benchmarking a Catchment-Aware Long Short-Term Memory Network (LSTM) for Large-Scale Hydrological Modeling, *Hydrology and Earth System Sciences Discussions*, 2019, vol. 22, pp. 1–36.

14. Palchevsky E. V., Khristodulo O. I., Pavlov S. V. Threat prediction in complex distributed systems using artificial neural network technology, *CEUR Workshop Proceedings*, 2020, vol. 2763, pp. 289–294 (in Russian).

15. Mokhov V. G., Tsimbol V. I. Electrical Energy Consumption Prediction of the Federal District of Russia on the Based of the Recurrent Neural Network, *Journal of Computational and Engineering Mathematics*, 2018, vol. 5, no. 2, pp. 3–15 (in Russian).

ООО "Издательство "Новые технологии". 107076, Москва, ул. Матросская Тишина, д. 23, стр. 2
Технический редактор Е. М. Патрушева. Корректор А. В. Чугунова.

Сдано в набор 27.04.2022 г. Подписано в печать 30.06.2022 г. Формат 60×88 1/8. Заказ PI622
Цена свободная.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru

Рисунки к статье N. M. Borgest, S. A. Vlasov
 «USING VIRTUAL REALITY TECHNOLOGY IN THE DESIGN
 OF COMPLEX SYSTEMS»

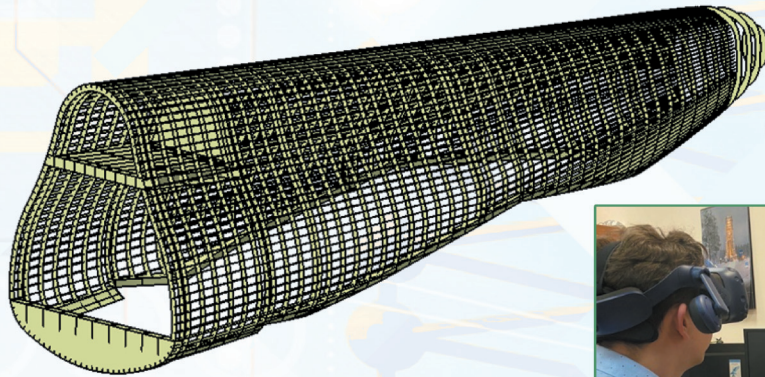


Fig. 6. Fuselage frame



Fig. 7. Working with the robot designer in VR

Engine selection

PST to ensure straggling from a dirt runway	0.098
PST to ensure straggling from a dirt runway	0.28
PST to ensure a given run-up length	0.58
PST in cruising mode	0.435

Engine types

Choos an aircraft layout

Engine selection

Mass in the first approximation

Basic parameters

Parameter	Value
Number of crew members, person	8
Payload, kg	130000
Cruising speed, km/h	850
Cruising altitude, m	10000
Runway length, m	2000
Maximum flight range, km	5000
Specific load on the wing dN/m ²	483.321
Wing sweep angle, degrees	35
Wing extension	7.5
Narrowing of the wing	3.46
Sweep angle of the horizontal tail, degrees	31
Narrowing of the horizontal tail	2.8

3

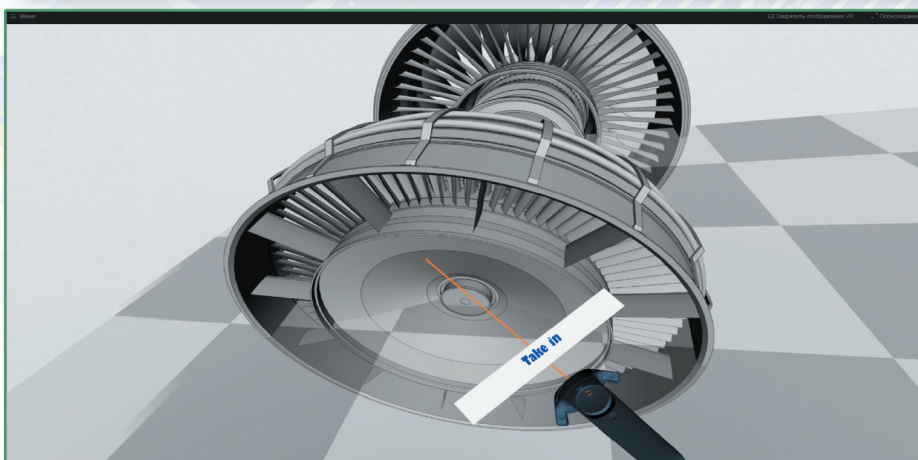
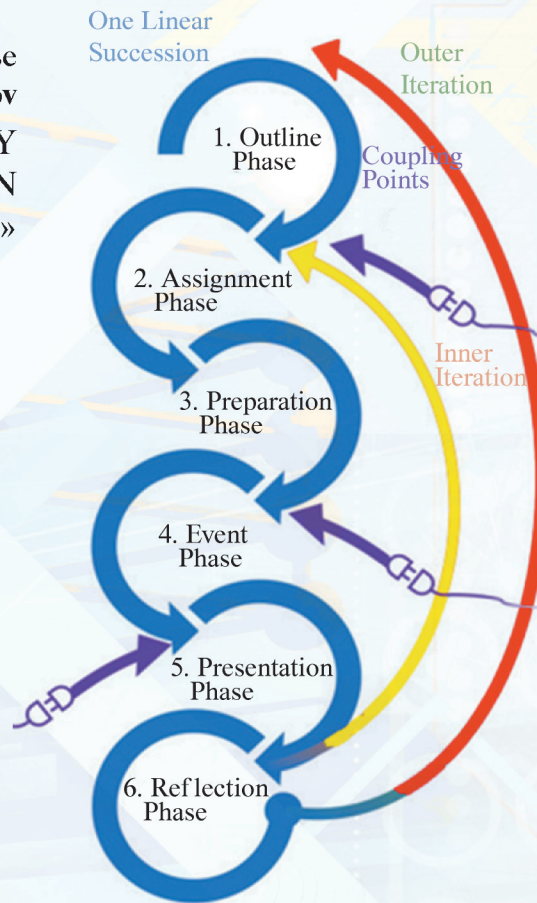


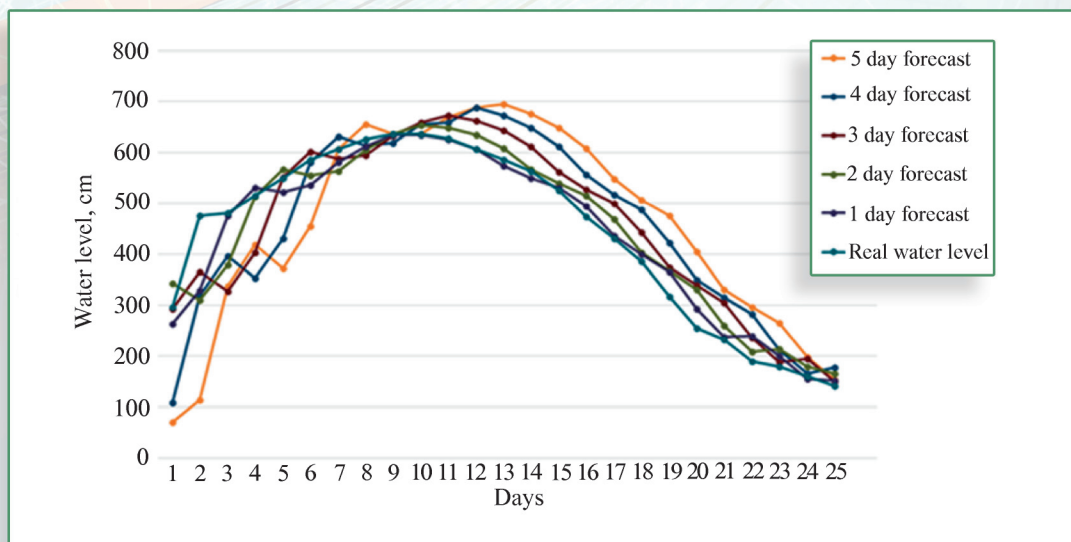
Fig. 8. Working with the robot designer in VR, engine model

Рисунок к статье
N. M. Borgest, S. A. Vlasov
**«USING VIRTUAL REALITY
 TECHNOLOGY IN THE DESIGN
 OF COMPLEX SYSTEMS»**



**Fig. 9. A phase model for
 VR and AR training**

Рисунок к статье **E. V. Palchevsky, V. V. Antonov**
**«STABILITY DECISION SUPPORT SYSTEM BASED ON APPLICATION
 OF THE SECOND GENERATION NEURAL NETWORK»**



**Fig. 4. The result of forecasting water levels for 5 days
 at the stationary hydrological station 76289 in Ufa**