

УДК 004.415.2, 004.9

© А. Д. Обухов, М. Н. Краснянский

НЕЙРОСЕТЕВАЯ АРХИТЕКТУРА ИНФОРМАЦИОННЫХ СИСТЕМ

Рассматривается задача автоматизации процесса разработки информационных систем на основе применения оригинальной нейросетевой архитектуры. Проведен анализ существующих подходов к автоматизации проектирования информационных систем. Сформулированы рекомендации к архитектуре информационных систем, направленные на снижение негативного влияния человеческого фактора. Представлена общая концепция нейросетевой архитектуры в виде структурной модели, даны определения основных сущностей и компонентов. Ключевыми отличиями нейросетевой архитектуры являются: независимость ключевых сущностей информационных систем и возможность автоматизации их проектирования и взаимодействия на основе применения нейронных сетей; изолированность математического обеспечения архитектуры; разграничение моделей информационных процессов и функциональных элементов от управляющих систем и систем представления информации; учет влияния окружения на процессы движения информационных потоков, элементы управления и представления системы; возможность адаптации структурных блоков информационных систем под особенности предметной области, параметры оборудования пользователя без необходимости внесения существенных изменений в архитектуру. Рассмотрено понятие нейросетевого канала данных, его структура и обобщенное математическое обеспечение. Осуществлена декомпозиция структурной модели. Представлены структурные схемы каждой сущности нейросетевой архитектуры информационных систем, описание основных компонентов, используемые нейросетевые каналы данных для связи сущностей и их компонентов. Проанализирована область применения нейросетевой архитектуры.

Ключевые слова: нейросетевая архитектура, нейросетевой канал данных, автоматизация проектирования информационных систем, искусственный интеллект, адаптивность.

DOI: [10.20537/vm190312](https://doi.org/10.20537/vm190312)

Введение

Одним из направлений развития современных информационных систем является их персонализация, адаптация под индивидуальные особенности каждого пользователя. Другой актуальной задачей является переход к автоматизированному проектированию, что позволит снизить влияние человеческого фактора, сократить время и трудоемкость разработки.

Таким образом, при разработке информационных систем перед разработчиками встает ряд нетривиальных задач: формализация предметной области [1], реализация архитектуры информационной системы, выбор методов проектирования, персонализация навигации и интерфейса под особенности каждого пользователя, стабильная работа на различных платформах и оборудовании, интеграция в существующую информационную среду, модернизация системы в соответствии с внешними или внутренними воздействиями [2]. Однако основным отрицательным фактором, влияющим на развитие и распространение современных адаптивных информационных систем, является их высокая стоимость и большие сроки разработки, что снижает коммерческий интерес к подобным системам. Особенно это актуально для небольших коллективов разработчиков, занимающихся проектами малого и среднего масштабов.

Поэтому актуальной научно-практической задачей является совершенствование методологии проектирования информационных систем за счет получения новых архитектур, методов и подходов к автоматизированной разработке программного обеспечения. Это позволит снизить требования к коллективу разработчиков, увеличить объем рынка и конкуренцию за нем за счет привлечения новых участников, уменьшить стоимость разработки и сроки внедрения информационных систем. Достижение перечисленных целей также ускорит процессы цифровизации общества за счет использования и развития технологий искусственного интеллекта.

Проведен анализ существующих методологий и подходов к проектированию информационных систем: CASE-технологии [2–4]; SADT [2,3,5]; CALS-технологии [2,6]; гибкие методологии разработки (экстремальное программирование, DSDM, Scrum, FDD, BDD [3,7]); RUP [3]; RAD [3,8]. Несмотря на широкое распространение и эффективность данных методологий, задача автоматизации процесса разработки в них не решается. Их применение позволяет снизить затраты материальных или временных ресурсов, оптимизировать процессы проектирования систем, но универсальными и однозначно применимыми подходами они не являются, так как каждая задача и предметная область накладывает свои ограничения.

Исследуя подходы к формализации и математическому моделированию информационных систем, рассмотрены различные модели (графовые, теоретико-множественные, информационные, функциональные, автоматные и т.д. [2,9,10]), однако, все они направлены на описание процессов либо получение конечной структуры информационной системы. Несмотря на возможность формализации процессов, протекающих в информационных системах с использованием рассмотренных моделей, необходимо отметить их слабые стороны в области решения задачи автоматизации проектирования. Таким образом, до сих пор задача проектирования структуры информационных систем решается аналитически. С одной стороны, без влияния человеческого фактора получить оптимальную архитектуру системы затруднительно, с другой — субъективные оценки могут привести к ошибочному решению.

Одним из возможных методов решения задачи проектирования информационных систем является структурно-параметрический синтез. Рассмотрено несколько перспективных направлений автоматизации этого процесса: методология эволюционного структурно-параметрического синтеза имитационных моделей путем использования аппарата сетей Петри для моделирования процессов взаимодействия компонентов [11]; метод морфологического ящика [12]; структурно-параметрический моделлер на основе теоретико-графового аппарата и набора правил [13]; автоматический синтез математических моделей [14]. Проведенный анализ подходов к автоматизации проектирования информационных систем показал, что универсального и полного решения этой задачи не существует, в каждой предметной области предлагаются свои методы, позволяющие частично автоматизировать процесс синтеза сложных систем.

Однако, анализ источников позволил сформулировать некоторые рекомендации для решения задачи автоматизации разработки информационных систем: необходимость разработки общей, универсальной архитектуры системы, позволяющей описать целый класс информационных систем; определение целевых функций; формирование блока структурно-параметрического синтеза с использованием современных технологий, в том числе экспертных систем, систем поддержки принятия решений, технологий искусственного интеллекта. Ни одна из ранее разработанных математических или информационных моделей не подходит для решения поставленной задачи, так как требуемая концепция должна объединить воедино программные составляющие информационной системы, особенности предметной области, структуру информационных потоков, набор необходимых функций, различные варианты представления данных, а также учитывать влияние внешней среды и пользователей,

характеристики тех устройств, с помощью которых они взаимодействуют с системой.

В рамках данной статьи будет рассмотрена концепция нейросетевой архитектуры информационных систем и ее декомпозиция. Применение предлагаемой архитектуры направлено на решение поставленных задач автоматизации проектирования информационных систем.

1. Концепция нейросетевой архитектуры информационных систем

Проведенный литературный обзор позволил сформулировать перечень требований и ограничений для разрабатываемой архитектуры. Их выполнение позволит решить поставленные задачи, связанные с автоматизацией процесса проектирования информационных систем:

- декомпозиция информационной системы в виде совокупности ключевых сущностей, связанных между собой унифицированными каналами передачи и обработки информации, функционирующими на основе технологий искусственного интеллекта;
- изолированность и независимость сущностей информационной системы, что позволяет осуществить декомпозицию задачи проектирования, параллельную автоматизированную разработку модулей системы, модернизацию одних компонентов без влияния на работоспособность других [15, 16];
- отделение математического и алгоритмического обеспечения функционирования информационной системы от математической модели данных и информационных потоков, протекающих в предметной области;
- независимость функционирования модулей информационной системы от структуры информационных потоков, воздействия внешней среды, программно-аппаратных характеристик оборудования [15, 16];
- независимость управляющих элементов и компонентов визуального представления информации от модели данных для кроссплатформенного проектирования информационной системы под различные виды интерфейса, программные и аппаратные платформы, операционные системы [12, 15];
- необходимость анализа и учета влияния пользователей и внешней среды на процессы движения информационных потоков, элементы управления и визуального представления информационной системы для последующей адаптации интерфейса под индивидуальные особенности пользователей и программно-аппаратные характеристики их оборудования, обеспечение саморегулирования и устойчивости системы [12].

Частично перечисленные выше требования учитываются в шаблоне проектирования MVC (Model – View – Controller) [17]. В данной концепции и ее производных отражены требования к изолированности и независимости компонентов информационной системы. При реализации нейросетевой архитектуры информационных систем использовались некоторые обозначения и подходы, применяемые в шаблоне MVC, чтобы обеспечить лучшее понимание предлагаемой концепции со стороны разработчиков, согласованность общепринятых понятий и выражений.

На первом этапе исследований выдвигалась гипотеза о возможности использования шаблона MVC в исходном его представлении для решения поставленных задач автоматизации проектирования информационных систем. При таком подходе структура информационной системы декомпозируется на три основные сущности: Модель (содержит необходимую

бизнес-логику и данные), Управление (функциональные элементы) и Представление (визуализация и отображение данных, интерфейс). Пользователи и внешние воздействия при таком подходе учитываются, как составная часть Модели.

Центральное положение в такой концепции занимает Управление, являясь достаточно сложной и громоздкой подсистемой со множеством разноплановых функций как по обработке и передаче данных, так и по организации взаимодействия между составными частями Модели. И, хотя это не противоречит поставленным условиям, оптимальным с точки зрения сложности проектирования такой подход назвать нельзя. Также открытым остается вопрос, каким образом можно автоматизировать проектирование настолько сложного по структуре Управления.

Необходимо учитывать, что пользователь оказывает влияние на саму систему (обусловлено требованием адаптивности системы) и структуру информационных потоков (пользователи выступают как инициаторы, наблюдатели, исполнители и т. д.). Кроме того, включенные в Модель факторы внешней среды также могут оказывать влияние на структуру и параметры системы. Таким образом, с одной стороны пользователи и факторы среды входят в состав Модели, а с другой — являются управляющей стороной, косвенно или напрямую влияющей на функционирование информационной системы в ходе взаимодействия с ней. Подобное противоречие можно реализовать на практике в программном обеспечении, но в теоретическом виде достаточно тяжело формализовать. При подобной реализации сущности архитектуры становятся тесно связанными, зависящими друг от друга, что не соответствует перечисленным выше условиям.

Таким образом, несмотря на применимость некоторых положений и подходов шаблона MVC в рамках решаемой задачи, решено реализовать оригинальную архитектуру информационных систем, не вступающую в противоречия со сформулированными требованиями.

На основе литературного обзора и имеющегося у коллектива практического опыта определено, что наибольшую сложность при решении задачи проектирования информационных систем представляет процесс автоматизации взаимодействия между компонентами системы. Недостаточно осуществить правильную декомпозицию информационной системы и реализовать каждый из модулей. Без правильно выстроенных каналов передачи и обработки данных между модулями система не будет работоспособна. Данная проблема часто встречается на практике, когда требуется организовать взаимодействие разных информационных систем, каждая из которых работает по своему принципу, значительно различается по структуре, интерфейсам получения и обработки информации.

Для решения этой ключевой проблемы предлагается использование таких инструментов машинного обучения, как нейронные сети [18, 19]. Интенсивность развития и перспективность данных технологий, постоянно растущая эффективность работы алгоритмов и повышение вычислительной мощности оборудования позволяют сделать вывод о применимости нейронных сетей при решении задачи автоматизации обработки данных, реализации взаимодействия модулей и компонентов информационной системы, выбора оптимальных параметров системы в зависимости от множества переменных. Возможность обучить нейронную сеть корректно реагировать на внешние воздействия и управлять структурой и параметрами модулей открывает совершенно новые перспективы по автоматизированной адаптации и проектированию информационных систем. Перечисленные достоинства позволяют говорить о том, что именно нейронные сети станут ключевой особенностью предлагаемой архитектуры.

Однако, необходимо понимать, что большинство информационных систем либо взаимодействуют с пользователем, либо в автоматическом режиме адаптируются к условиям внешней среды. Поэтому, с одной стороны, необходимо при автоматизации процессов обработки, передачи и хранения информации обеспечить комфортное взаимодействие чело-

века с системой, что зачастую не реализовано из-за огромного разнообразия платформ, особенностей программного и аппаратного обеспечения терминала пользователя. С другой стороны, в автоматических модулях и системах, функционирующих без участия человека, необходимо обеспечить их стабильную и надежную работу при различных внешних воздействиях. Поэтому вторым ключевым компонентом архитектуры является внешняя среда, как отдельная и определяющая многие параметры системы сущность. Таким образом, на основе концепции шаблона проектирования MVC предлагается оригинальная концепция нейросетевой архитектуры. Будем использовать следующие понятия.

Определение 1. *Нейросетевая архитектура* (Neural Network Architecture, NNA) — структура информационных систем, включающая следующие ключевые сущности: Окружение, Нейронные сети, Модель, Представление и Управление, обеспечивающая возможность их изолированного и независимого автоматизированного проектирования.

Определение 2. Под термином *Окружение* (Environment, E) в NNA будем понимать комплексную структуру, включающую внешние факторы воздействия на информационную систему, сведения о пользователях системы, программно-аппаратные характеристики их терминалов, права доступа к информации и принимающую участие в процессах управления и воздействия на другие сущности.

Определение 3. Под термином *Нейронные сети* (Neural Networks, NN) в NNA будем понимать некоторое множество алгоритмов машинного обучения на основе искусственных нейронных сетей, их математическое, алгоритмическое и программное обеспечение, используемое для решения задач генерации, обработки и передачи информации, автоматизированного синтеза структуры и параметров компонентов информационной системы и реализации их межмодульного взаимодействия.

Определение 4. Под термином *Модель* (Model, M) в NNA будем понимать математическое, даталогическое и алгоритмическое обеспечение информационной системы, объединенное в соответствии с направлением решаемых задач и изолированное от других сущностей архитектуры.

Определение 5. Под термином *Представление* (View, V) в NNA будем понимать текстовое, графическое или иное отображение информации, формируемое в соответствии с запросами пользователей либо с состоянием системы.

Определение 6. Под термином *Управление* (Control, C) в NNA будем понимать совокупность управляющих блоков, включающих алгоритмическое, математическое и программное обеспечение, по заданным соотношениям осуществляющих управление информационной системой и взаимодействие между остальными компонентами архитектуры под влиянием Окружения.

Определение 7. *Модулем* в NNA является элемент сущности или некоторое их множество, направленный на решение конкретной задачи хранения, обработки или передачи данных.

На основе перечисленных выше основных определений формализуем нейросетевую архитектуру информационных систем в виде структурной схемы первого уровня декомпозиции (т. е. не рассматривая внутреннюю структуру каждой сущности). Данная схема представлена на рис. 1.

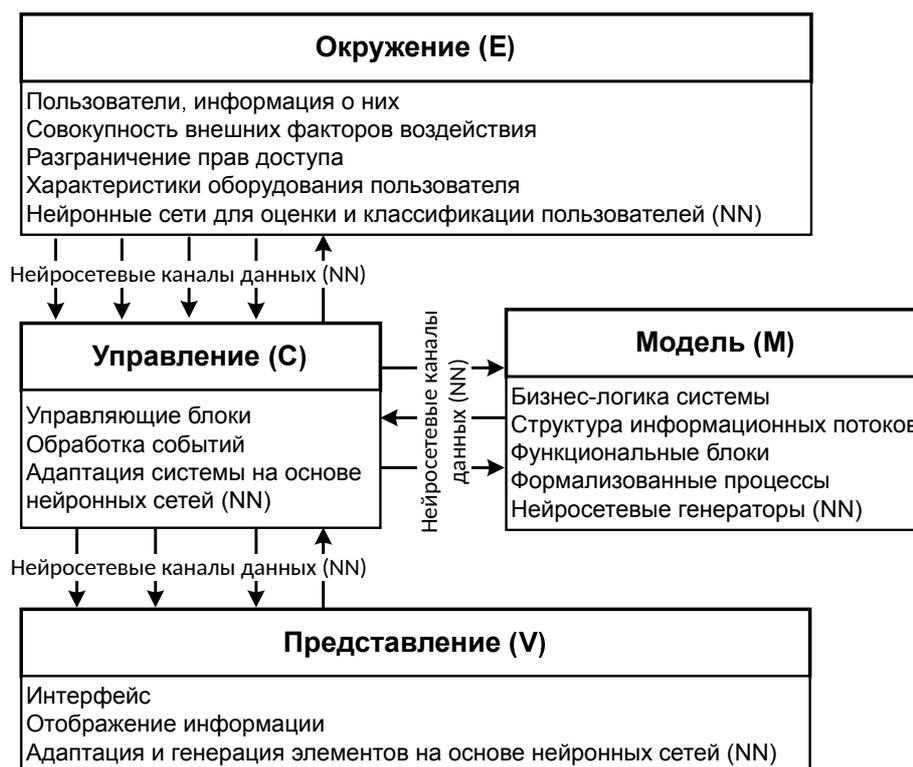


Рис. 1. Общая структурная схема нейросетевой архитектуры информационных систем

Окружение, как независимая сущность, объединяет сведения о множествах пользователей (их правах доступа, личных данных, спецификации программного и аппаратного обеспечения) и внешних факторов воздействия. С одной стороны, Окружение влияет на систему извне, адаптируя ее под особенности каждого пользователя и его личные предпочтения, с другой стороны — сами пользователи являются участниками процессов создания и обработки информации, исполнителями или наблюдателями, что выражается в Модели как ссылка на конкретного пользователя из Окружения. Напрямую при этом Модель и Окружение не взаимодействуют. Пользователи, входящие в Окружение, в процессе работы с информационной системой активно контактирует с сущностью Управление. При этом в зависимости от уровня доступа пользователя в Управлении ему доступны только необходимые управляющие блоки и функции.

Управление является связующим звеном между остальными сущностями. В нем происходит обработка команд от пользователей, отслеживание внешних воздействий, прием и передача данных в другие компоненты информационной системы, управление возникающими в системе событиями, регулирование параметров в соответствии с требованиями. Управление активно взаимодействует с Моделью: передает ей новые данные, получает необходимую информацию из хранилища, осуществляет контроль за обработкой и дальнейшей передачей информации из Модели в Представление. В Управлении также реализована некоторая часть интерфейса, ответственная за генерацию событий, получение и отправку данных.

Модель является достаточно сложной структурой в рассматриваемой архитектуре, так как включает формализованное представление бизнес-логики системы, структуры информационных потоков, хранение информации, математическое и алгоритмическое обеспечение требуемых операций по обработке и преобразованию данных. Отдельные компоненты Модели могут взаимодействовать между собой, но некоторая изолированность между ни-

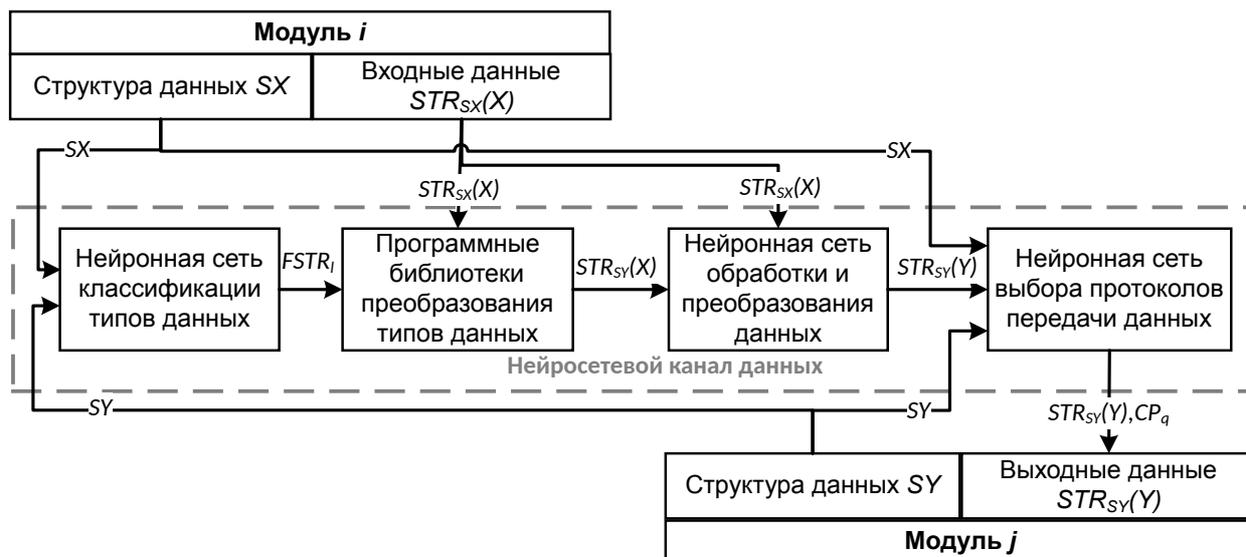


Рис. 2. Схематическая структура нейросетевого канала данных

ми остается. Модель не является стационарной, она постоянно динамически развивается за счет появления новых данных, расширения структуры информационных потоков и добавления дополнительной функциональности в информационную систему.

Сущность Представление отвечает за визуализацию интерфейса для Окружения. Параметры интерфейса могут гибко меняться в зависимости от воздействия Управления, подстраиваясь под требования конкретных пользователей, оборудование или иные внешние воздействия. Через Управление передается необходимый объем данных, отображаемый в Представлении, а также управляющие блоки, адаптированные под текущие настройки интерфейса.

В состав каждой сущности входят компоненты, основанные на технологиях машинного обучения и относящиеся к сущности Нейронные сети. Также все сущности взаимодействуют между собой через Управление, что отражается на структурной схеме в связях между блоками. Подобные связи обозначим понятием нейросетевые каналы данных и также отнесем к сущности Нейронные сети.

Для организации автоматизированного взаимодействия между модулями необходимо использовать принципиально новый способ передачи и обработки данных, основанный на технологиях искусственного интеллекта. Обозначим его, как нейросетевой канал данных.

Определение 8. Под термином *нейросетевой канал данных* (Neural Network Data Channel, NNDC) будем понимать общее представление программного интерфейса автоматизированной организации связи между компонентами NNA, включающее следующие элементы: входные и выходные данные, их структуры, искусственные нейронные сети для классификации типов данных и выбора протоколов передачи данных, программные библиотеки преобразования и передачи данных.

Графически структуру нейросетевого канала данных можно представить следующим образом (рис. 2).

Процесс преобразования данных можно формализовать следующим соотношением:

$$NNDC_{i,j}^{type}(STR_{SX}(X), SY) = STR_{SY}(Y),$$

где $NNDC_{i,j}^{type}$ — нейросетевой канал данных категории $type$, осуществляющий взаимодействие между i -м и j -м модулями информационной системы;

$X = \{x_n\} | n = \overline{1, xN}$ — входные данные размерностью xN со структурой данных SX ;

$Y = \{y_m\} | m = \overline{1, yM}$ — выходные данные размерностью yM со структурой данных SY ;

Примем обозначение $STR_{SX}(X)$ как приведение данных X к структуре вида SX . Аналогично $STR_{SY}(Y)$ — приведение данных Y к структуре вида SY .

Опишем общий принцип функционирования нейросетевого канала данных. Канал $NNDC_{i,j}^{type}$ устанавливает связь между двумя модулями: с модуля i поступают входные данные X заданной структуры SX , на выходе из канала модулем j ожидаются данные Y структуры SY . Для обеспечения этого соответствия нейронная сеть классификации типов данных $NN_{i,j,type}^{ct}$ анализирует структуры SX и SY входных и выходных данных соответственно и выбирает из множества программных библиотек преобразования типов данных функцию $FSTR_l$:

$$NN_{i,j,type}^{ct}(SX, SY) = FSTR_l,$$

$$FSTR_l: STR_{SX}(X) \rightarrow STR_{SY}(X),$$

Далее осуществляет обработка и преобразование данных нейронной сетью $NN_{i,j,type}^{proc}$, если это требуется в рамках установленной связи:

$$NN_{i,j,type}^{proc}: STR_{SY}(X) \rightarrow STR_{SY}(Y),$$

Преобразованные данные $STR_Y(SY)$ передаются модулю j по протоколу передачи данных CP_q , определяемому нейронной сетью $NN_{i,j,type}^{tr}$:

$$NN_{i,j,type}^{tr}: (SX, SY) \rightarrow CP_q.$$

На этом работа нейросетевого канала данных завершается. Изложенное определение отражает понятие термина в общем виде, однако, позволяет охарактеризовать его структуру. Далее будут рассмотрены частные случаи нейросетевых каналов данных, используемых для решения конкретных задач передачи и обработки информации.

Возможность реализации нейросетевых каналов данных доказывается успешным решением множества задач классификации, преобразования и прогнозирования данных в различных предметных областях. Точность их решения стремится к 1 при нахождении оптимальной структуры слоев и количества нейронов [18, 19]. Таким образом, нейросетевые каналы данных, функционируя на основе технологий искусственного интеллекта, будут использоваться для автоматизированной организации взаимодействия всех модулей нейросетевой архитектуры. Это позволит значительно снизить вероятность человеческой ошибки, ускорить процесс интеграции модулей, обеспечить масштабируемость и адаптируемость системы в процессе ее жизненного цикла.

2. Декомпозиция нейросетевой архитектуры информационных систем

Сформулировав общую структурную схему нейросетевой архитектуры, осуществим декомпозицию каждой сущности и, используя аппарат нейросетевых каналов данных, конкретизируем их взаимодействие. Декомпозиция сущности Окружение представлена на рис. 3.

Множество факторов воздействия внешней среды объединяют в одном модуле математическое описание влияния внешнего воздействия на структуру и параметры информационной системы. К этим факторам относятся: действия пользователей, текущее состояние оборудования, на котором функционирует информационная система, модернизация системы, приводящая к необходимости изменения структуры и параметров отдельных модулей.

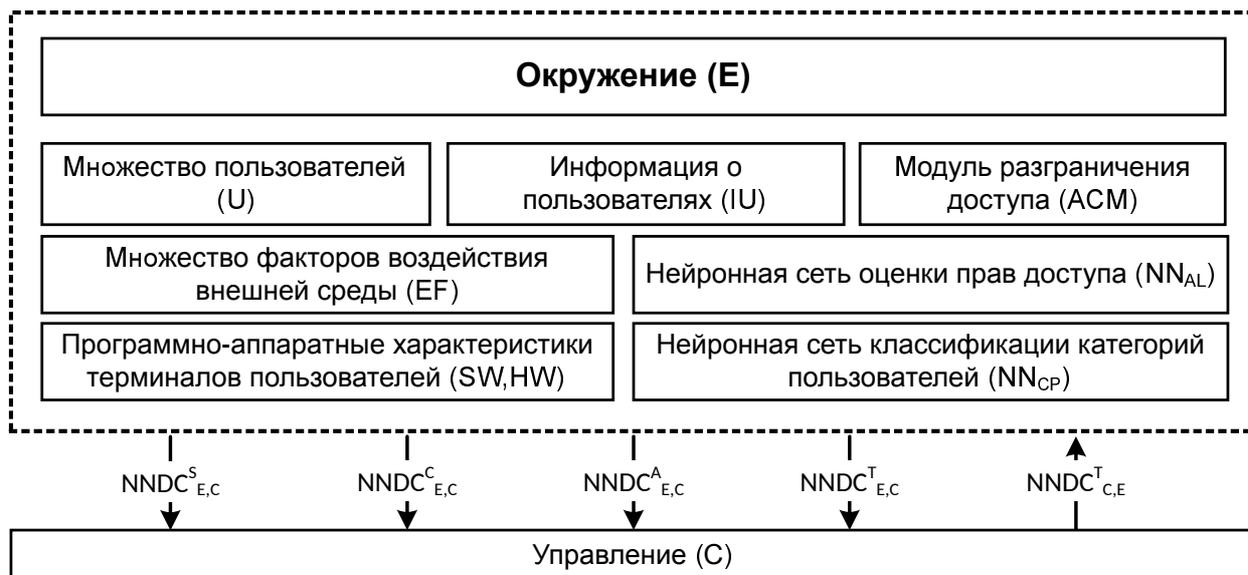


Рис. 3. Декомпозиция сущности Окружение

Модуль разграничения доступа (ACM) формализует в некотором (дискретном, ролевом или атрибутном) представлении множество уровней доступа AL к информации. В состав модуля входит нейронная сеть оценки прав доступа NN_{AL} , которая по формализованному представлению уровня доступа пользователя u_i позволяет получить некоторую общую оценку al_{ij} , используемую при решении задачи x_j разграничения доступа к данным, операциям, компонентам системы и т. д.

Нейронная сеть позволяет классифицировать пользователя u_i на основе данных из модуля «Информация о пользователях» (IU) по заданным категориям, кластеризовать по общим признакам, обнаруженным в процессе анализа.

Следующий модуль содержит сведения о программных SW и аппаратных HW характеристиках терминалов пользователей и включает все необходимые сведения об оборудовании пользователя, его производительности, размере и разрешении экрана устройства, операционной системе, браузере (если система реализована как Web-приложение), наборе дополнительного программного обеспечения.

Сущность Окружение связана с Управлением. Перечислим используемые нейросетевые каналы данных для связи этих сущностей.

- Нейросетевой канал безопасности $NNDC_{E,C}^S(AL) = (AF, AM, AD)$: сведения об уровне прав доступа AL пользователя поступают на вход нейронной сети и преобразуются во множество доступных для него в системе функций AF , модулей AM , данных AD .
- Нейросетевой канал управления $NNDC_{E,C}^C(Action) = Event$: выполняемые пользователем действия $Action$ в виде некоторого множества команд обрабатываются нейронной сетью для формирования множества генерируемых событий $Event$, необходимых для выполнения поставленных задач.
- Нейросетевой канал адаптации $NNDC_{E,C}^A(HW, SW, EF) = Interface$: собранная информация о программных SW и аппаратных HW характеристиках пользователей, текущем состоянии оборудования, на котором функционирует система, внешних факторах воздействия EF поступает на вход нейронной сети, которая преобразует полу-

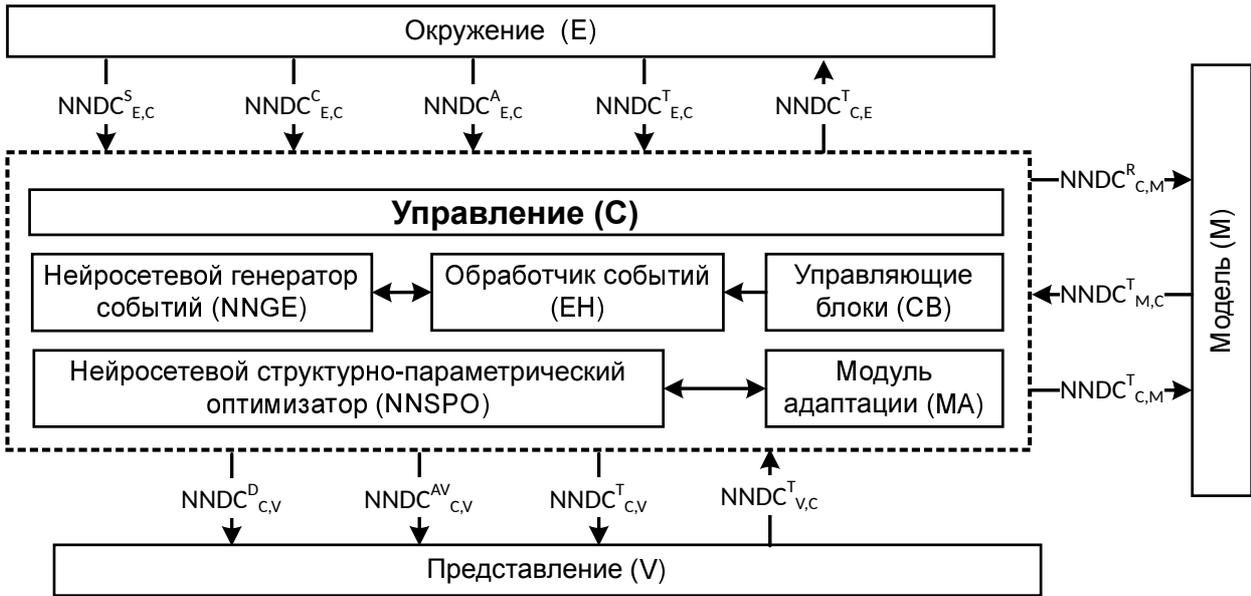


Рис. 4. Декомпозиция сущности Управление

ченные данные во множество доступных для пользователя элементов и параметров интерфейса *Interface*.

- Нейросетевой канал передачи данных

$$NNDC_{E,C}^T(STR_{SE}(Data), SC) = STR_{SC}(Data^*):$$

используется для передачи информации от Окружения к Управлению, которая преобразуется к требуемой структуре STR_{SC} и содержанию $Data^*$. Примем следующую сокращенную запись:

$$NNDC_{E,C}^T(STR_{SE}(Data), SC) = STR_{SC}(Data^*) \equiv NNDC_{E,C}^T(Data) = Data^*.$$

- Нейросетевой канал передачи данных $NNDC_{C,E}^T(Data) = Data^*$: полученные от Управления данные $Data$ (текстовая, графическая информация, документы, архивы, файлы и т.д.) анализируются нейронной сетью для их преобразования к формату $Data^*$, поддерживаемому на оборудовании, операционной системе и программном обеспечении терминалов Окружения. Механизм работы аналогичен предыдущему, отличаясь лишь направлением движения информации.

Таким образом, Окружение через нейросетевые каналы данных может передавать всю необходимую информацию и команды пользователей Управлению, а также получать от него требуемые данные. Сущность Управление (рис. 4) состоит из трех основных компонентов.

Управляющие блоки (CB) – совокупность элементов интерфейса и программных модулей, реализующие управляющие функции в *NNA*. Пользователи по нейросетевому каналу управления осуществляют передачу команд, что порождает необходимые события, работа с которыми осуществляется в обработчике событий.

Обработчик событий (EH) отвечает за реагирование компонентов системы на поступающие от пользователей и внешней среды команды по нейросетевому каналу управления. Обработчик работает совместно с нейросетевым генератором событий $NNGE(event_i) = (pd_{ij}, pl_{ij})$ – программным модулем на основе нейронных сетей, формирующим необходимое множество программного кода pd и логику pl обработки множества событий.

Модуль адаптации (MA) используется для настройки параметров информационной системы (например, интерфейса) под программно-аппаратные характеристики оборудования пользователей, а также для оптимизации системы в процессе ее функционирования и модернизации по выбранному набору критериев. Основой модуля адаптации является нейросетевой структурно-параметрический оптимизатор $NNSPO(SM, PM, R) = (SM^*, PM^*)$, который на основе текущих структуры SM , параметров PM модулей системы и заданного множества критериев R подбирает оптимальные SM^* и PM^* . К функциям модуля адаптации относится также корректировка структуры информационных потоков вследствие структурных изменений в информационной системе.

Занимая центральное положение в нейросетевой архитектуре, Управление взаимодействует с остальными сущностями через множество нейросетевых каналов данных. Помимо уже рассмотренных выше, к основным нейросетевым каналам данных Управления относятся:

- Нейросетевой канал отображения данных $NNDC_{C,V}^D(\{ce_i\}) = \{de_i\}$: данный тип связи используется для передачи и преобразования управляющих элементов $\{ce_i\}$ из Управления в Представление для их последующего отображения в виде множества адаптированных под требования интерфейса элементов $\{de_i\}$.
- Нейросетевой канал адаптации представления

$$NNDC_{C,V}^{AV}(Interface) = (V_{type}, V_{param}):$$

на основе полученных от Окружения параметров интерфейса $Interface$ по данному каналу передаются управляющие команды, осуществляющие настройку Представления к заданному виду интерфейса V_{type} и значениям его параметров V_{param} . Нейронная сеть при этом осуществляет аппроксимацию тех параметров, что не удалось найти в управляющих командах, либо подбирает наиболее подходящее под них состояние Представления.

- Нейросетевые каналы передачи данных $NNDC_{C,V}^T$ и $NNDC_{V,C}^T$: используются для передачи данных из Управления в Представление и наоборот.
- Нейросетевой канал запроса данных $NNDC_{C,M}^R(r) \rightarrow NNDC_{C,E}^T(Data) = Data^*$: согласно условиям текущего события формируется запрос r на получение данных, обрабатываемый нейронной сетью с целью получения необходимого набора данных $Data^*$ из Модели по соответствующему нейросетевому каналу передачи данных $NNDC_{M,C}^T$. Для сокращения записи примем, что под символом « \rightarrow » будем понимать последовательность движения информации по нейросетевым каналам (слева направо).
- Нейросетевые каналы передачи данных $NNDC_{M,C}^T$, $NNDC_{C,M}^T$: осуществляют прием информации от Модели в рамках запросов и передачу данных от Управления в Модель.

Таким образом, Управление через нейросетевые каналы данных может взаимодействовать со всеми сущностями архитектуры, с некоторыми эта связь двусторонняя.

Декомпозиция сущности Модель представлена на рис. 5. Каждый внутренний компонент Модели изолируется от остальных и связывается только через нейросетевые каналы данных, что позволяет при изменениях одного компонента не нарушить структуру и данные другого. Данное правило обеспечивает целостность данных и возможность независимого формирования каждой Модели и замены их одну на другую. Анализ решаемых Моделью

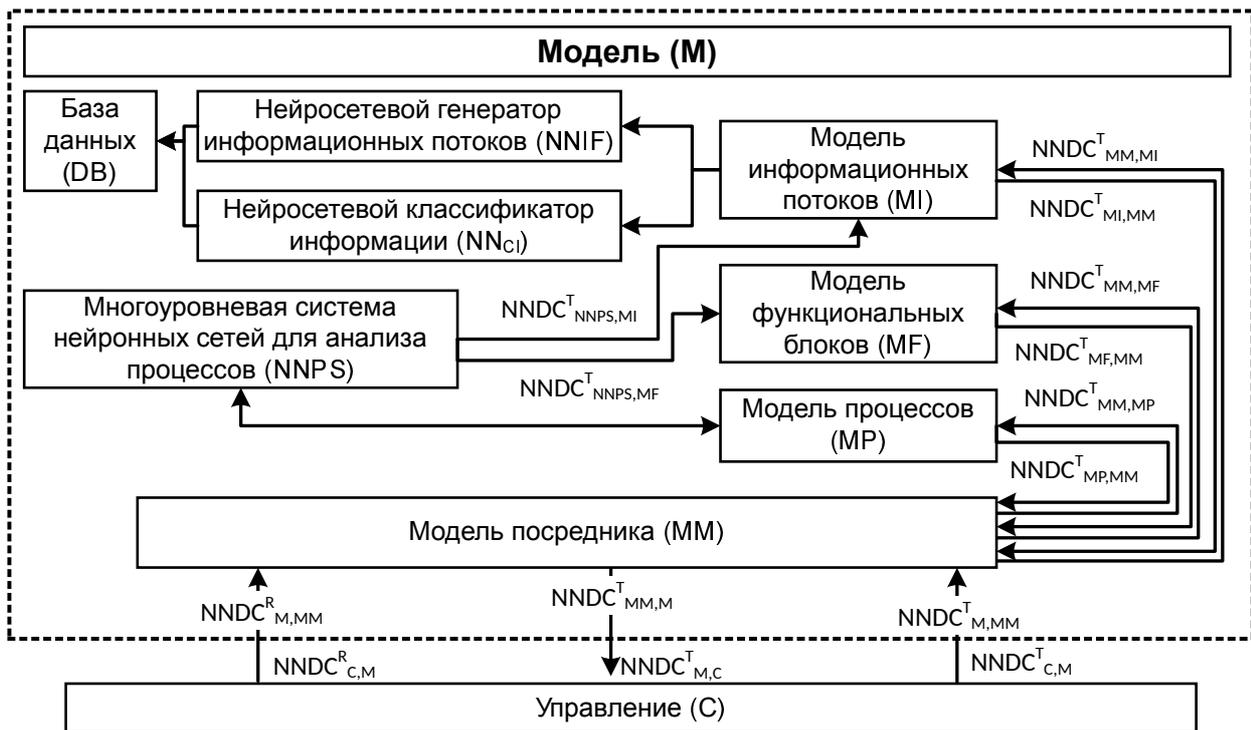


Рис. 5. Декомпозиция сущности Модель

задач показал необходимость ее декомпозиции на четыре ключевых компонента: Модель информационных потоков (содержит данные о структуре, процессах движения и взаимодействия информационных объектов, их жизненных циклах), Модель функциональных блоков (содержит математические соотношения, реализующие определенные операции по обработке и преобразованию данных, направленные на решение конкретных задач в рамках программных модулей), Модель процессов (используется для анализа предметной области и автоматизированной генерации данных и модулей для других компонентов Модели) и Модель посредника (используется для упрощения механизма взаимодействия между Моделью и Управлением и координации операций приема-передачи данных с другими сущностями в одном блоке).

Модель информационных потоков MI будет отвечать только за данные о процессах движения информации, формализуемых с помощью многоуровневой модели графов MG [1]. Данная модель MG обрабатывается в нейросетевом генераторе информационных потоков $NNIF(MG) = DB$, образуя необходимую структуру базы данных DB , которая далее в автоматизированном формате заполняется данными об информационных объектах, их жизненных циклах, пользователях системы, служебной и отчетной информацией и т. д.

Распределение информации в автоматическом режиме возможно за счет работы нейросетевого классификатора информации $NNCl(\{a_{ij}\}) = cd_k, cd_k \rightarrow \{dbt_m\}$, по множеству атрибутов $\{a_{ij}\}$ поступивших данных определяющего его категорию cd_k , которой сопоставлены таблицы базы данных dbt_m .

Модель функциональных блоков MF включает набор готовых программных модулей, реализуя таким образом функции (интерфейсы) по обработке данных. Важным условием при реализации каждого модуля является его независимость и стандартизация входных $Data_{in}$ и выходных $Data_{out}$ данных, что позволит применять его для решения конкретной задачи в рамках любой предметной области и обеспечивать работоспособность на любых данных из области допустимых значений. Работу каждого функционального блока MF_i

можно описать следующим соотношением: $MF_i(Data_{in}) = Data_{out}$.

Модель процессов MP формализует специфические процессы движения информации и на основе их анализа генерирует необходимые функциональные блоки или информационные потоки для остальных компонентов Модели, что позволяет системе самостоятельно адаптироваться под изменения бизнес-процессов организации. Для автоматизации такой сложной задачи планируется использование многоуровневой системы $NNPS$ нейронных сетей $\{NN_i\}$, анализирующих процессы предметной области PSA и генерирующих на их основе графы информационных потоков MG^* и структуры данных SMG_i функциональных блоков для соответствующих компонентов Модели, которые передаются по каналам $NNDC_{NNPS,MI}^T$ и $NNDC_{NNPS,MF}^T$.

Модель посредника MM используется для упрощения взаимодействия между сущностями таким образом, чтобы все потоки данных (как входящие, так и исходящие) проходили через нее, что позволяет сократить количество нейросетевых каналов данных между Управлением и Моделью. Таким образом, принимая от Управления запросы, Модель посредника определяет, к каким компонентам Модели относится этот запрос, после чего через множество каналов $NNC_{MM,X}^T$ (где X — некоторый элемент Модели) получает необходимые данные $Data$ и передает их обратно Управлению в два этапа (сначала от посредника к Модели, потом от Модели к Управлению). Внешне со стороны Управления данный механизм не прослеживается, что позволяет использовать нейросетевой канал запроса данных с разными Моделями, так как всю архитектурную разницу примет на себя Модель посредника. Входящие извне данные также передаются не напрямую в Модель, а через посредника и далее в необходимые компоненты по следующему алгоритму:

$$NNDC_{C,M}^T(Data) \rightarrow NNDC_{M,MM}^T(Data) \rightarrow \{NNDC_{MM,X}^T(Data)\}.$$

Рассмотрим взаимодействие Модели с другими сущностями, а также между ее компонентами. Внешняя связь ограничивается тремя нейросетевыми каналами данных с Управлением. Все основные компоненты связаны с Моделью посредника нейросетевыми каналами передачи данных, чтобы обеспечить передачу информации в двух направлениях между компонентами и Управлением. Два канала передачи данных осуществляют взаимодействие $NNPS$ с MG и MI . Также имеются три внутренних канала, принимающие и передающие данные на границе между Управлением и Моделью. Таким образом, все необходимые операции по передаче информации можно осуществлять через Модель посредника.

Подводя итог, отметим, что сущность Модели имеет самую сложную структуру движения информации, которую удалось упорядочить и сосредоточить в одном месте за счет использования Модели посредника.

Последняя сущность нейросетевой архитектуры — Представление. Ее декомпозиция показана на рис. 6. Основным компонентом Представления является Интерфейс — совокупность набора визуальных элементов v_i , отображаемых на устройстве пользователя в соответствии с требованиями, которые сформированы Управлением.

Формирование графической, текстовой, звуковой или иной информации на основе полученных от Управления данных для визуальных элементов возможна в автоматическом режиме на основе работы нейросетевого генератора интерфейса

$$NNGI(V_{type}, V_{param}) = \{v_i\}.$$

Генератор интерфейса анализирует полученные по каналу управления отображением $NNDC_{C,V}^D$ тип и параметры интерфейса и выбирает из некоторой библиотеки готовые визуальные элементы v_i .

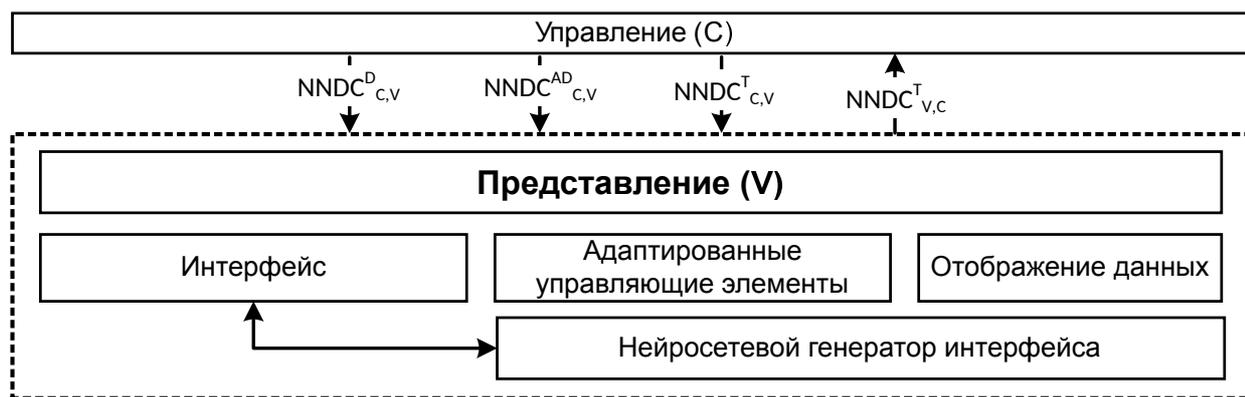


Рис. 6. Декомпозиция сущности Представление

Второй задачей является автоматическая адаптация Представления под характеристики оборудования и используемое программное обеспечение пользователей, которая осуществляется последовательной передачей нужной информации сначала через канал адаптации $NNDC_{E,C}^A$, а потом по каналу управления отображением $NNDC_{C,V}^D$. При этом учитывается мощность оборудования, которое используется при выборе тех средств визуализации интерфейса, что обеспечат наилучшее взаимодействие с системой, а также такие параметры как размеры и разрешение экрана для компоновки элементов представления. Помимо визуальных элементов Представление включает также отображение полученных из Управления данных, выводимых в элементы интерфейса или непосредственно на экран, и адаптированные под визуальные требования управляющие элементы de_i . В нейросетевой архитектуре Представление взаимодействует лишь с Управлением по четырем нейросетевым каналам данных, рассмотренным выше.

На этом декомпозиция NNA завершена, дальнейший анализ будет связан уже с описанием структуры и закономерностей функционирования каждого из рассмотренных модулей, нейронных сетей и генераторов, нейросетевых каналов данных, используемых для создания, обработки и передачи информации.

Область применения представленной архитектуры в настоящее время ограничена системами обработки, хранения и передачи информации, например, электронными архивами, системами документооборота, обучающими и справочными системами, автоматизированными системами управления данными, но в дальнейшем может быть расширена.

Заключение

В статье рассмотрена актуальная задача автоматизации проектирования информационных систем. Рассмотренные подходы к решению данной задачи в различных предметных областях позволили сформулировать перечень требований к структуре систем, а также сделать вывод, что без перехода к новому типу архитектуры решить задачу подобной сложности невозможно. Проведен анализ возможности использования известного шаблона проектирования MVC для решения поставленной задачи автоматизации, однако, в результате обнаружено несколько критических недостатков такого направления.

Сформулирована нейросетевая архитектура информационных систем, построенная на основе взаимодействия пяти основных сущностей: Нейронные сети, Окружение, Модель, Представление и Управление. Отличительными особенностями представленной архитектуры является учет влияния внешней среды и пользователей на структуру и параметры системы, а также реализация процесса автоматизированного взаимодействия модулей по-

средством нейросетевых каналов данных, основанных на технологиях искусственного интеллекта и машинного обучения. Нейронные сети также используются при генерации отдельных компонентов сущностей, что позволит автоматизировать процесс их формирования. В рамках проведенных исследований также рассмотрена декомпозиция нейросетевой архитектуры, описаны закономерности функционирования и взаимодействия компонентов сущностей на основе нейросетевых каналов данных различного типа.

В ходе дальнейших исследований планируется разработка необходимого математического, методологического и алгоритмического обеспечения в рамках предложенной нейросетевой архитектуры, что позволит перейти к реализации необходимого программного обеспечения для решения задачи автоматизации информационных систем.

Финансирование. Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации в рамках проектной части, проект № 8.2906.2017/ПЧ.

СПИСОК ЛИТЕРАТУРЫ

1. Obukhov A.D., Krasnyansky M.N., Dedov D.L., Karpushkin S.V. Mathematical model of information processing in electronic document management system // *International Review of Automatic Control*. 2018. Vol. 11. No. 6. P. 336–345. <https://doi.org/10.15866/ireaco.v11i6.15305>
2. Краснянский М.Н., Карпушкин С.В., Остроух А.В., Обухов А.Д., Касатонов И.С., Букреев Д.В., Карпов С.В., Дедов Д.Л. Проектирование информационных систем управления документооборотом научно-образовательных учреждений. Тамбов: Изд-во ТГТУ, 2015. 216 с.
3. Сидихменова Е.И., Ланец С.А. Применение CASE-технологий при проектировании информационных систем // *Научно-техническое и экономическое сотрудничество стран АТР в XXI веке*. 2013. Т. 1. С. 288–294.
4. Bahill A.T., Madni A.M. System design and the design process // *Tradeoff decisions in system design*. Cham: Springer, 2017. P. 1–157. https://doi.org/10.1007/978-3-319-43712-5_1
5. Marca D.A. SADT/IDEF0 for augmenting UML, Agile and usability engineering methods // *International Conference on Software and Data Technologies*. Berlin–Heidelberg: Springer, 2011. P. 38–55. https://doi.org/10.1007/978-3-642-36177-7_3
6. Levin M.Sh. Combinatorial optimization in system configuration design // *Automation and Remote Control*. 2009. Vol. 70. Issue 3. P. 519–561. <https://doi.org/10.1134/S0005117909030187>
7. Карпов Д.В. Гибкая методология разработки программного обеспечения // *Вестник Нижегородского университета им. Н.И. Лобачевского*. 2011. № 3 (2). С. 227–230.
8. Beynon-Davies P., Carne C., Mackay H., Tudhope D. Rapid application development (RAD): an empirical review // *European Journal of Information Systems*. 1999. Vol. 8. Issue 3. P. 211–223. <https://doi.org/10.1057/palgrave.ejis.3000325>
9. Soui M., Diab S., Ouni A., Essayeh A., Abed M. An ontology-based approach for user interface adaptation // *Advances in Intelligent Systems and Computing* / Ed.: N. Shakhovska. Cham: Springer, 2017. P. 199–215. https://doi.org/10.1007/978-3-319-45991-2_13
10. Zeigler B.P. Object-oriented simulation with hierarchical, modular models: intelligent agents and endomorphic systems. Academic Press, 1990. 414 p. <https://doi.org/10.1016/C2013-0-11737-0>
11. Петросов Д.А., Ломазов В.А., Басавин Д.А. Эволюционный синтез систем на основе заданной элементной базы компонентов // *Научные ведомости Белгородского государственного университета. Сер. Экономика. Информатика*. 2015. № 7 (204). Вып. 34/1. С. 116–124.
12. Rakov D.L., Sinyev A.V. The structural analysis of new technical systems based on a morphological approach under uncertainty conditions // *Journal of Machinery Manufacture and Reliability*. 2015. Vol. 44. No. 7. P. 650–657. <https://doi.org/10.3103/S1052618815070110>

13. Цырков А.В. Структурно-параметрический моделлер — основа построения комплексных информационных моделей производственных систем // Информационные технологии в проектировании и производстве. 2005. № 1. С. 51–58.
14. Прошин И.А., Прошин Д.И., Прошина Р.Д. Структурно-параметрический синтез математических моделей объектов исследования по экспериментальным данным // Вестник Астраханского государственного технического университета. Сер. Морская техника и технология. 2009. № 1. С. 110–115.
15. Macik M., Cerny T., Slavik P. Context-sensitive, cross-platform user interface generation // Journal on Multimodal User Interfaces. 2014. Vol. 8. No. 2. P. 217–229.
<https://doi.org/10.1007/s12193-013-0141-0>
16. Terletskyi D.O., Provotar O.I. Mathematical foundations for designing and development of intelligent systems of information analysis // Problems in Programming. 2014. Vol. 15. Nos. 2–3. P. 233–241.
<https://arxiv.org/abs/1510.04183>
17. Reenskaug T. The model-view-controller (MVC). Its past and present. 2003.
https://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC_pattern.pdf
18. Pattanayak S. Natural language processing using recurrent neural networks // Pro deep learning with TensorFlow. Berkeley, CA: Apress, 2017. P. 223–278.
https://doi.org/10.1007/978-1-4842-3096-1_4
19. Iatan I.F. Mathematical aspects of using neural approaches for information retrieval // Issues in the use of neural networks in information retrieval. Cham: Springer, 2017. P. 1–35.
https://doi.org/10.1007/978-3-319-43871-9_1

Поступила в редакцию 01.07.2019

Обухов Артем Дмитриевич, к. т. н., старший преподаватель, кафедра «Системы автоматизированной поддержки принятия решений», Тамбовский государственный технический университет, 392000, Россия, г. Тамбов, ул. Советская, 106.

E-mail: obuhov.art@gmail.com

Краснянский Михаил Николаевич, д. т. н., профессор, ректор, Тамбовский государственный технический университет, 392000, Россия, г. Тамбов, ул. Советская, 106.

E-mail: kras@tambov.ru

Цитирование: А. Д. Обухов, М. Н. Краснянский. Нейросетевая архитектура информационных систем // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2019. Т. 29. Вып. 3. С. 438–455.

A. D. Obukhov, M. N. Krasnyansky

Neural network architecture of information systems

Keywords: neural network architecture, neural network data channel, information systems design automation, artificial intelligence, adaptability.

MSC2010: 68T01, 68T05

DOI: [10.20537/vm190312](https://doi.org/10.20537/vm190312)

The problem of process automation for the development of information systems based on the application of the original neural network architecture is considered. An analysis of existing approaches to the automation of information systems design is carried out. Recommendations for the information systems architecture, aimed at reducing the negative impact of human factor, are formulated. A general concept of neural network architecture in the form of a structural model is presented. Definitions of the main entities and components are given. The key differences of the neural network architecture are: the independence of the key entities of information systems and the possibility of automation of their design and interaction based on the use of neural networks; isolation of the mathematical software of architecture; separation of models of information processes and functional elements from control systems and information representation systems; taking into account the influence of the environment on the processes of movement of information flows, the elements of control and system visualization; the possibility of adapting structural units of information systems to the characteristics of the subject area, the parameters of user equipment without the need to make significant changes to the architecture. The concept of a neural network data channel, its structure and generalized mathematical software are considered. The decomposition of the structural mode is implemented. The structural diagrams of each entity of the neural network architecture of information systems, the description of the main components, the neural network data channels used to connect the entities and their components are presented. The scope of application of the neural network architecture is analyzed.

Funding. The study was funded by the Ministry of Science and Higher Education of the Russian Federation in the framework of the project part, project no. 8.2906.2017/ПЧ.

REFERENCES

1. Obukhov A.D., Krasnyansky M.N., Dedov D.L., Karpushkin S.V. Mathematical model of information processing in electronic document management system, *International Review of Automatic Control*, 2018, vol. 11, no. 6, pp. 336–345. <https://doi.org/10.15866/ireaco.v11i6.15305>
2. Krasnyansky M.N., Karpushkin S.V., Ostroukh A.V., Obukhov A.D., Kasatonov I.S., Bukreev D.V., Karpov S.V., Dedov D.L. *Proektirovanie informatsionnykh sistem upravleniya dokumentooborotom nauchno-obrazovatel'nykh uchrezhdenii* (Information systems design for document flow management in scientific and educational institutions), Tambov: Tambov State Technical University, 2015, 216 p.
3. Sidikhmenova E.I., Lanets S.A. The use of CASE-technologies in the design of information systems, *Nauchno-tekhnicheskoe i ekonomicheskoe sotrudnichestvo stran ATR v XXI veke*, 2013, vol. 1, pp. 288–294 (in Russian).
4. Bahill A.T., Madni A.M. System design and the design process, *Tradeoff decisions in system design*, Cham: Springer, 2017, pp. 1–157. https://doi.org/10.1007/978-3-319-43712-5_1
5. Marca D.A. SADT/IDEF0 for augmenting UML, Agile and usability engineering methods, *International Conference on Software and Data Technologies*, Berlin–Heidelberg: Springer, 2011, pp. 38–55. https://doi.org/10.1007/978-3-642-36177-7_3
6. Levin M.Sh. Combinatorial optimization in system configuration design, *Automation and Remote Control*, 2009, vol. 70, issue 3, pp. 519–561. <https://doi.org/10.1134/S0005117909030187>

7. Karpov D.V. Agile methodology of software development, *Vestnik Nizhegorodskogo Universiteta imeni N.I. Lobachevskogo*, 2011, no. 3 (2), pp. 227–230 (in Russian).
8. Beynon-Davies P., Carne C., Mackay H., Tudhope D. Rapid application development (RAD): an empirical review, *European Journal of Information Systems*, 1999, vol. 8, issue 3, pp. 211–223.
<https://doi.org/10.1057/palgrave.ejis.3000325>
9. Soui M., Diab S., Ouni A., Essayeh A., Abed M. An ontology-based approach for user interface adaptation, *Advances in Intelligent Systems and Computing*, Ed.: N. Shakhovska. Cham: Springer, 2017, pp. 199–215. https://doi.org/10.1007/978-3-319-45991-2_13
10. Zeigler B.P. *Object-oriented simulation with hierarchical, modular models: intelligent agents and endomorphic systems*, Academic Press, 1990, 414 p.
<https://doi.org/10.1016/C2013-0-11737-0>
11. Petrosov D.A., Lomazov V.A., Basavin D.A. Evolutionary synthesis of systems based on a pre assigned element base of components, *Nauchnye Vedomosti Belgorodskogo Gosudarstvennogo Universiteta. Seriya: Ekonomika. Informatika*, 2015, no. 7 (204), issue 34/1, pp. 116–124 (in Russian).
12. Rakov D.L., Sinyev A.V. The structural analysis of new technical systems based on a morphological approach under uncertainty conditions, *Journal of Machinery Manufacture and Reliability*, 2015, vol. 44, no. 7, pp. 650–657. <https://doi.org/10.3103/S1052618815070110>
13. Tsyrov A.V. Structural-parametric Modeler is the basis for the construction of complex information models of production systems, *Informatsionnye Tekhnologii v Proektirovanii i Proizvodstve*, 2005, no. 1, pp. 51–58 (in Russian).
14. Proshin I.A., Proshin D.I., Proshina R.D. Structural-parametric synthesis of mathematical models of objects of research on experimental data, *Vestnik Astrakhanskogo Gosudarstvennogo Tekhnicheskogo Universiteta. Seriya: Morskaya Tekhnika i Tekhnologiya*, 2009, no. 1, pp. 110–115 (in Russian).
15. Macik M., Cerny T., Slavik P. Context-sensitive, cross-platform user interface generation, *Journal on Multimodal User Interfaces*, 2014, vol. 8, no. 2, pp. 217–229.
<https://doi.org/10.1007/s12193-013-0141-0>
16. Terletskiy D.O., Provotar O.I. Mathematical foundations for designing and development of intelligent systems of information analysis, *Problems in Programming*, 2014, vol. 15, nos. 2–3, pp. 233–241.
<https://arxiv.org/abs/1510.04183>
17. Reenskaug T. *The model-view-controller (MVC). Its past and present*, 2003.
https://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC_pattern.pdf
18. Pattanayak S. Natural language processing using recurrent neural networks, *Pro deep learning with TensorFlow*, Berkeley, CA: Apress, 2017, pp. 223–278.
https://doi.org/10.1007/978-1-4842-3096-1_4
19. Iatan I.F. Mathematical aspects of using neural approaches for information retrieval, *Issues in the use of neural networks in information retrieval*, Cham: Springer, 2017, pp. 1–35.
https://doi.org/10.1007/978-3-319-43871-9_1

Received 01.07.2019

Obukhov Artem Dmitrievich, Candidate of Engineering, Senior Lecturer, Department of Automated Systems of Decision-Making Support, Tambov State Technical University, ul. Sovetskaya, 106, Tambov, 392000, Russia.

E-mail: obuhov.art@gmail.com

Krasnyansky Mikhail Nikolaevich, Doctor of Engineering, Rector, Tambov State Technical University, ul. Sovetskaya, 106, Tambov, 392000, Russia.

E-mail: kras@tambov.ru

Citation: A.D. Obukhov, M.N. Krasnyansky. Neural network architecture of information systems, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2019, vol. 29, issue 3, pp. 438–455.