

КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

УДК 681.326:519.613

ЛОГИЧЕСКИЕ СХЕМЫ И АРХИТЕКТУРЫ КИБЕРСОЦИАЛЬНОГО КОМПЬЮТИНГА

ХАХАНОВА А.В.

Предлагается развитие технологии цифровизации отношений, связанное с разработкой кубитного логического секвенсора для моделирования и распознавания social-явлений в потоках больших данных на основе интерпретативных кубитных матричных моделей, методов и архитектур киберсоциального компьютеринга, направленного на автоматический синтез и анализ социально-логических схем, ориентированных на мониторинг и управление социальными процессами и явлениями. Совершенствуется архитектура memory-driven киберфизического компьютеринга, которая отличается параллелизмом процедур синтеза и анализа логических схем, предназначенных для моделирования social-процессов и явлений в целях мониторинга и управления. Модернизируются кубитно-матричные модели логических схем, которые отличаются унитарным кодированием многозначных переменных для синтеза секвенсоров в целях параллельного анализа киберсоциальных процессов. Предлагаются кубитно-матричные векторные методы синтеза и анализа логических схем, которые характеризуются унитарным кодированием значений многозначных переменных для параллельного моделирования social-процессов и явлений. Разрабатываются кубитно-матричные модели, структуры данных, методы синтеза и анализа логических схем social-процессов, которые позволяют моделировать реакцию социальной системы на принятие конструктивных и деструктивных решений человека, руководителя, чиновника, благодаря кубитному описанию эталонов поведения, что дает возможность актуаторно управлять гражданами и инфраструктурой.

Ключевые слова: киберсоциальный компьютеринг, социально-логические схемы, кубитные структуры данных, методы синтеза и анализа логических схем, распознавание социальных явлений и процессов.

1. Введение

Основой классического компьютеринга традиционно является таблица истинности [1-4], которая в явном виде задает функциональное описание поведения технологического объекта, робота, человека, социальной группы. Другие формы, такие как: аналитические уравнения, альтернативные графы, кубитно-векторные структуры [3-5] ориентированы на минимизацию памяти для хранения данных при описании сложных функциональностей. История вопроса заключается в исчерпывающей формализации вычислений, которые связаны с математическими процессами, имеющих отношение к технике и технологиям

[1]. Относительно формального описания деятельности человека или социальных групп существует практически белое пятно, которое частично покрывается машинным обучением [6-9]. Последнее есть не что иное, как эволюционное, а значит медленное, построение таблицы истинности для любого, еще не изученного процесса, как правило, связанного с био-социальными объектами. При синтезе таблицы истинности существует две фазы [6, 7]. Первая называется обучением (Analysis), когда каждому входному (социальному) воздействию (Input Data Flow) ставится в соответствие значение выхода (Truth Table) на основании введенного критерия кластеризации (сходства–различия). Вторая – тестирование, когда выполняется сравнение (Comparison) обученной или синтезированной таблицы истинности с фрагментом уже известной. Результатом тестирования является оценка качества полученной модели (рис. 1), как количество совпадений G , относенное к общему числу тестовых воздействий (Test Table): $Q=G/T$.

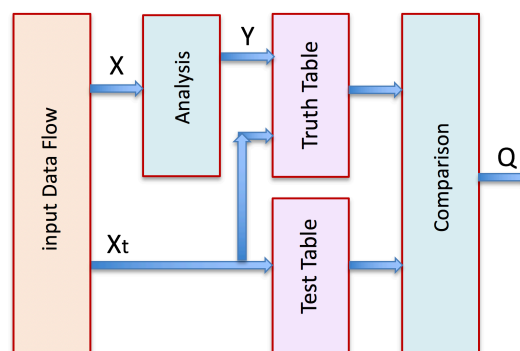


Рис. 1. Machine Learning for Truth Table Synthesis

Таким образом, для организации киберсоциального компьютеринга необходимо, прежде всего, научиться автоматически строить таблицы истинности, которые представляют собой дискретное описание социальных процессов и явлений, предназначенное для моделирования и предсказания последствий от принятия решений социальными группами, руководителями и сотрудниками компаний и университетов [10-14].

Цель исследования – существенное повышение качества мониторинга и управления общественных процессов за счет имплементации в практику моделей, методов, алгоритмов и архитектур киберсоциального компьютеринга, которые дают возможность моделировать реакцию общества на актуаторные воздействия, а также распознавать и предсказывать социальные явления.

Сущность исследования – разработка моделей, включая кубитные структуры [15-17], методов, алгоритмов и архитектур киберсоциального компьютеринга для моделирования реакции общества

на управленческие воздействия, распознавания и предсказания социальных явлений, что дает возможность существенно повысить качество мониторинга и управления социальными процессами, а также качество жизни людей при сохранении экологии окружающей среды.

Объект исследования – технологии синтеза, анализа, цифровых логических структур для моделирования и распознавания киберсоциальных процессов и явлений на основе создания кубитных структур данных и специализированных вычислительных архитектур.

Предмет исследования – модели, методы и алгоритмы синтеза и анализа цифровых логических структур для моделирования и распознавания киберсоциальных процессов и явлений на основе кубитных структур данных и специализированных вычислительных архитектур.

Задачи исследования: 1) Формальный синтез таблицы истинности для киберсоциальной функциональности. 2) Синтез и анализ социальных логических схем для моделирования общественных и производственных процессов. 3) Логические типовые структуры функциональностей и комбинационные кубитные схемы для моделирования и распознавания социальных явлений. 4) Структурная модель хог-отношений матричного киберсоциального компьютеринга. 5) Архитектуры, методы и алгоритмы компьютеринга для распознавания киберсоциальных явлений.

2. Метод синтеза социально-логических схем по таблицам истинности

Формы описания киберфизических функциональностей для последующего анализа представлены в том числе и работами [18-30]. Таблицы истинности для задания social-функциональностей формируются на основе позиционного или унитарного кодирования значений переменных. При этом предполагается замкнутость значений переменных в пределах social-функциональностей, которые формируют группу логических функций, заданных кубитными покрытиями таблиц истинности. Конкретная функциональность может оперировать не более, чем n значениями переменной, которые составляют алфавит или универсум примитивов-значений $A = \{A_1, A_2, \dots, A_i, \dots, A_n\}$. Следующая таблица иллюстрирует два вида кодирования универсума примитивов-значений переменной для формирования таблиц истинности трех логических функций, которые участвуют в создании social-функциональности:

A	Hash 16	P-Code	U-code	P1	P2	P3
A1	0100...10	00	1000	1	0	0
A2	0100...11	01	0100	1	0	1
A3	1100...10	10	0010	0	1	1
A4	0101...11	11	0001	0	1	0

Аналогичные таблицы истинности создаются для всех переменных, что в совокупности формирует матрицу таблиц истинности для каждого social-процесса или явления. По факту, столбец позиционного кодирования P-code не используется на практике, но он служит базовым компонентом для доказательства применимости классической таблицы истинности при описании любых social-процессов. Столбцы P1, P2, P3 тривиально используются для минимизации количества столбцов.

Далее предлагается матричная модель social-процессов на метрике значений переменных. Столбцы-кубиты P1, P2, P3 из предыдущей таблицы трансформируются в строки-векторы, которые представляют собой суперпозицию унитарных кодов значений параметров, участвующих в формировании social-процессов PT1 – PT3:

Pi	PT1	PT2	PT3
P1	1100	1101	1011
P2	0011	1011	0111
P3	0110	0111	0110
P4	1001	1011	1101

Дуализм интерпретации данной таблицы формирует иерархию, которую необходимо учитывать при синтезе модели social-процессов. 1) Полученная таблица или матрица объединения унитарных кодов, размещенных в координатах, представляет собой двоичную модель social-функциональности, привязанную к модельному времени PT1–PT3. Здесь существенно, что каждая координата матрицы представляет собой векторную или кубитную форму описания таблицы истинности. 2) Однако формат матричной модели также адекватно создает структуры данных для описания совокупного social-процесса, имеющего место в компании. При этом каждый столбец матрицы PTi интегрально задает social-функциональность.

Модель, алгебра, структура, граф, таблица, матрица, система, уравнение являются эквивалентными математическими понятиями, в основу которых положена структура взаимосвязанных компонентов. При этом всегда рассматривается замкнутый алфавит или множество примитивных компонентов, которые создают основание структуры или универсум примитивов. Все возможные связи между элементами алфавита или универсума формируют сигнатуру или базовые опе-

рации алгебры. Булева алгебра на самом низком уровне представлена алфавитом или универсумом примитивных символов $\{0,1\}$, которые фигурируют в значениях булевых переменных и функции $Y=f(X)$, где $\{X,Y\}=\{0,1\}$. При этом поведение функции определяется таблицей истинности, где упорядоченной двоичной последовательности входных переменных ставится в соответствие двоичное значение функции. Менее распространенной является интерпретация таблицы истинности, где каждому двоичному коду или адресу ставится в соответствие единичное или нулевое значение функции. При этом совокупность кодов или адресов представляет собой универсум примитивных компонентов или основание алгебры, на которой определяются базовые операции. Таким образом, вводится алгебра логики, где многозначные состояния входной переменной (алфавит) кодируются в таблице истинности двоичными векторами, которые представляют собой адреса ячеек памяти, где хранятся $\{1,0\}$ -значения функции. Интегрально 1-значения функции в таблице истинности формируют подмножество существующих примитивов $A=\{a,c,e,f\}$ на заданном универсуме $A=\{a,b,c,d,e,f,g,h\}$, которое суперпозиционно создает функциональность:

String	Code	Function
a	000	1
b	001	0
c	010	1
d	011	0
e	100	1
f	101	1
g	110	0
h	111	0

Кубитным покрытием данной функциональности является F-вектор двоичных состояний выходной переменной, размерность которого равна универсуму примитивных компонентов, формирующих функцию, а число единичных значений равно подмножеству примитивов из универсума, которое участвует в формировании заданной функциональности. Следует отметить, что функциональность формируется значениями существенной переменной во временных фреймах social-процесса или social-паттерна. Учитывая, что количество существенных переменных, как правило больше 1, то необходимо синтезировать цифровые логические схемы из кубитных покрытий функций, число которых равно количеству существенных переменных. Таким образом, конечное множество существенных переменных является базовыми элементами для синтеза цифровых логических схем social-функциональности

или social-процесса. Далее представлена структура, которая оперирует кубитными покрытиями примитивов, объединенными логикой элементов: and, or (рис. 2). Логические схемы, синтезированные из кубитных форм значений переменных, предназначены для моделирования social-процессов в целях определения поведения киберфизической social-архитектуры компьютеринга на заданных входных рабочих воздействиях. Рабочими воздействиями являются суперпозиции унитарных кодов входных значений переменных. Состояние выхода логической social-схемы, равное единице, свидетельствует о позитивном результате влияния входных значений существенных переменных на ход выполнения social-паттерна или процесса для достижения поставленной цели. Таким образом, вместо цепочки данных, иллюстрирующей последовательность действий в social-функциональности, предлагается принципиально новая форма – комбинационная цифровая логическая схема, параллельно интегрирующая только существенные свойства social-функциональности, где основным отличием является возможность моделирования social-процессов.

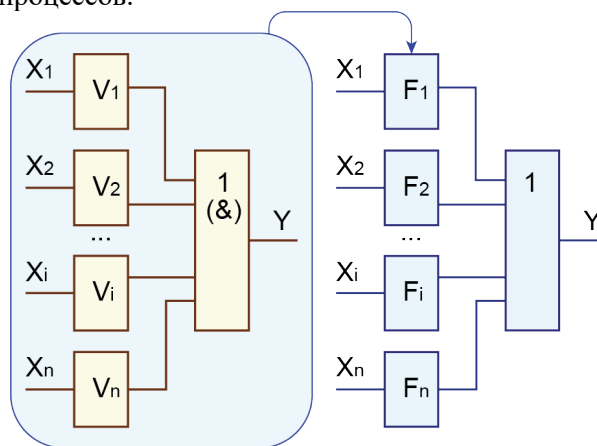


Рис. 2. Logic social-functions

Схемная структура формирует функциональное поведение или модель social-функциональности на основе использования кубитных покрытий переменных, инвариантную ко времени. Совокупность social-функциональностей создает основу для синтеза параллельной цифровой модели social-процесса, которая представляет собой специализированный вычислитель, реализующий кибер-физический компьютеринг для мониторинга, моделирования и управления social-процессами компании. Иерархия social-компьютеринга представлена компонентами:

<значение – переменная – функциональность – процесс> или <value – variable – function – process>.

Логическая структура или social-процессор инвариантен ко времени, создает social-функциональности на основе кубитных форм таблиц истинности логических элементов, привязанных к универсуму примитивных значений переменных. Преимущества social-процессора заключаются в компактности представления и высоком быстродействии логической схемы social-функциональности, которая определяется кубитными векторами переменных, битовая размерность которых равна количеству примитивных данных-значений каждой переменной. Кубитные структуры данных инварианты к их hardware реализации в виде логической схемы, или software, в виде таблиц-матриц описания функциональностей [18-20]. Кубитная двоичная форма описания social-функциональностей дает возможность технологично моделировать сколь угодно сложные social-процессы путем воздействия на схему двоичными входными наборами, которые соответствуют унитарно-кодированным данным, в целях определения выходных состояний social-процессора, выполняющих роль классификатора и/или актюатора. Позитивным представляется тот факт, что каждый кубитный логический элемент, заданный в векторном формате количества значений переменной, создает образ social-функциональности единичными значениями своих координат. При этом число кубитных векторов-элементов, объединенных в схему and(or)-элементом, равно количеству переменных синтезируемой social-функциональности. Кроме того, визуализация достаточно компактной логической схемы задания social-функциональности дает возможность пользователю или руководителю увидеть сущность social-функциональностей без привязки ко времени, где важным представляется покрыть все необходимые переменные конструктивными действиями пользователя для получения соответствующих сервисов.

Недостатком процессорной social-модели является исключение параметра времени в функционировании логической модели, что приводит к ее неадекватности в случае необходимости описания последовательностной граф-схемы алгоритма social-функциональности. Кроме того, для моделирования данных с помощью логической social-схемы необходимо синтезировать препроцессор текстовых данных в унитарные кубитные векторы на основе использования hash-функций, существенно уменьшающих время сравнения исходной входной текстовой строки с библиотечными данными. Необходим также постпроцессор для интерпретации состояний выходов

social-процессора пользователю (автомату), который будет генерировать последующие актюаторные действия. В совокупности препроцессор и постпроцессор занимают порядка 10 процентов памяти и вычислительного времени обработки данных от базового варианта, связанных с синтезом и анализом кубитных структур, реализующих архитектуру social-схемы предприятия.

Формальный синтез логических схем social-процесса компании осуществляется путем выполнения следующих пунктов:

- 1) Формирование универсума примитивных данных по каждому атрибуту social-процесса.
- 2) Заполнение вектор-кубита единичными значениями тех координат, номера которых соответствуют данным, используемым при формировании конкретной social-функциональности.
- 3) Объединение по and-функции всех логических элементов, заданных кубитами задействованных значений переменных, для получения схемы social-функциональности.
- 4) Объединение всех схем social-функциональностей для получения логического social-процессора компании.

Задачи анализа, решаемые с помощью логических social-схем:

- 1) Моделирование входного потока данных в целях их классификации на множестве библиотечных social-функциональностей путем использования вектора моделирования (покрытия), который дает возможность на каждом шаге определять состояние моделируемого social-процесса, а также генерировать управляющие воздействия, целенаправленные на получение покрытия social-функциональности, исходя из нулевых координат вектора моделирования.

- 2) Автоматическое генерирование входных двоичных векторов на основе анализа данных для их последующего моделирования на social-процессоре.

Для автоматического создания social-процессора на основе использования входного контента необходимо: 1) Определение ключевых слов для формирования переменных social-процесса и последующего вычисления универсумов примитивных значений данных. 2) Формирование идеи каждого social-процесса, создающей законченную social-функциональность в виде матрицы кубитных покрытий, составленных из унитарных кодов, соответствующих значениям из универсумов примитивов переменных. Инструментом для точного моделирования входных воздействий на кубитных покрытиях social-функциональности является функция принадлежности, которая вычисляется путем определе-

ния кодового расстояния между входным двоичным набором, подаваемым на логический элемент, и кубитным вектором последнего:

String	Input	Func	Xor	$\mu(I,F)=$
a	1	1	0	0,75
b	0	0	0	
c	0	1	1	
d	0	0	0	
e	1	1	0	
f	1	1	0	
g	0	0	0	
h	1	0	1	

Однако принадлежность части social-функциональности определяется операцией логического умножения или пересечения, которая должна быть равна входному вектору $I \wedge F = I$, что означает покрытие логическим элементом social-функциональности значения входного вектора $I \in F$.

Итак, структуры данных social-аналитики или архитектура social-процессора, изображенного на рис. 3, содержит: 1) Метрику в виде множества $U = \{U_1, U_2, \dots, U_i, \dots, U_n\}$ кубитных покрытий универсумов примитивных значений, которое имеет мощность, равную количеству переменных n . Каждый универсум $U_i = \{U_{i1}, U_{i2}, \dots, U_{ij}, \dots, U_{im}\}$ формирует полное множество примитивных значений переменной. 2) На основе данной метрики, путем синтаксического сравнения, синтезируются эталонные social-функциональности $P = \{P_1, P_2, \dots, P_r, \dots, P_k\}$, где столбец $P_r = \{P_{r1}, P_{r2}, \dots, P_{ri}, \dots, P_{rn}\}$ формирует social-функциональность, а координата столбца определяется двоичным вектором $P_{ri} = \{P_{ri1}, P_{ri2}, \dots, P_{rij}, \dots, P_{rim}\}$, записанным в формате кубитного покрытия $U_i = \{U_{i1}, U_{i2}, \dots, U_{ij}, \dots, U_{im}\}$ универсума примитивных значений переменной. При этом $P_{rij} = 1$, если в social-функциональности присутствует символьное значение переменной, равное U_{ij} . Кубитное покрытие social-функциональности $P_r = \{P_{r1}, P_{r2}, \dots, P_{ri}, \dots, P_{rn}\}$ всегда является структурной частью кубитного покрытия универсума примитивов: $P_r \in U$, поскольку всегда работает аксиома $P_r \wedge U = P_r$. 3) Целью создания эталонных social-функциональностей является автоматическое генерирование актюаторных воздействий $A = (A_1, A_2, \dots, A_p, \dots, A_q)$, которые замыкают цикл social-аналитики управляющими воздействиями, превращая ее в RPA-структуру.

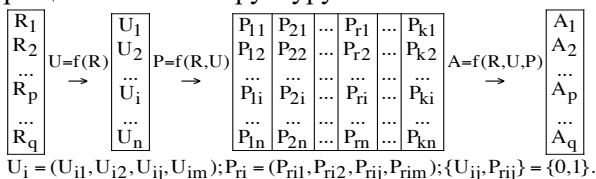


Рис. 3. Матричная архитектура social-компьютинга

Матричная архитектура social-компьютинга поддерживает иерархию кубитных покрытий $P_r \in P_r \in U$, где P_r -вектор входных значений переменной не может быть больше P_{ri} -кубита social-переменной, который не может быть больше универсума U_i значений переменной.

Архитектура программного серверного приложения для синтеза логических social-функциональностей и последующего моделирования на них контента больших social-данных представлена на рис. 4.

Структура содержит две части, где левая из них предназначена для автоматического синтеза структур данных, логических элементов и схемы в целом. Правая часть ориентирована на ручной ввод схемных элементов и структуры social-функциональности на основе использования графического интерфейса. Обе части архитектуры нагружены на блок кубитного моделирования входного контента в целях определения в нем аналогов social-процессов, конструктивных и деструктивных, синтезированных ранее по другому контенту. Результаты моделирования сохраняются в библиотеке, которая содержит также и логические эталоны social-функциональностей.

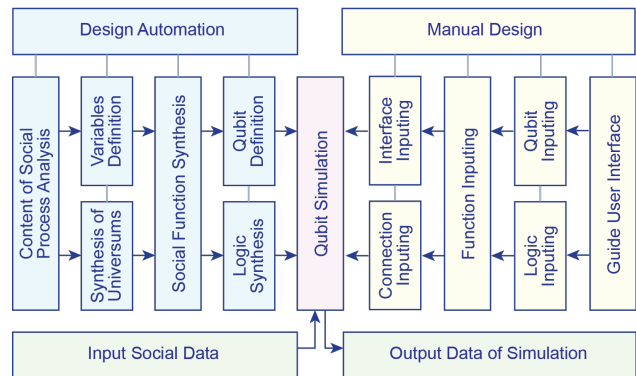


Рис. 4. Архитектура social-процессора

3. Логические структуры social-функциональностей

Предлагается решение в части синтеза логических схем для анализа social-процессов, связанное с использованием кубитно-регистровых переменных, которые унитарно кодируют множество примитивных значений из многозначного универсума переменной. В этом случае любая функциональность может быть представлена логическим элементом and (or , or), который имеет регистровые входные переменные, представленные кубитными векторами [2-4]. Единичное значение на выходе логического элемента с многозначными или векторными входами определяется совпадением сигналов с кубитными векторами по всем входным переменным. Для этого необхо-

димому выполнять процедуру, связанную с логическим пересечением (and-операция) входного сигнала с кубитным вектором, и последующим хог-сравнением результата пересечения с исходным сигналом. Если такое сравнение равно нулю по всем координатам векторов, то формируется единичное значение выхода для одной многозначной переменной. При наличии единиц на всех выходах переменных логического элемента and его значение будет равно единице. Аналитическая модель и схемная реализация процедуры сравнения двух векторов для определения принадлежности одного из них Q1 к другому Q2 имеет вид, представленный на рис. 5.

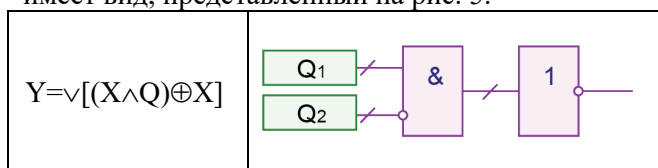


Рис. 5. Логика сравнения данных

Схемная реализация устройства для моделирования социальных процессов X путем сравнения с эталонными функциональностями Q представлена на рис. 6.

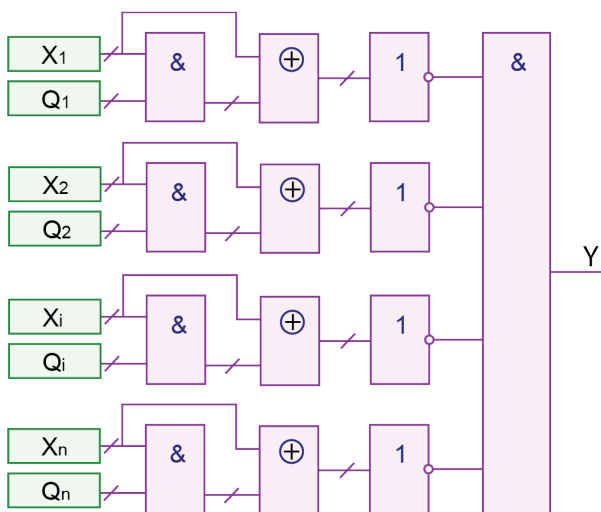


Рис. 6. Схема мониторинга и анализа социальных процессов

Здесь выполняется сравнение двоично-кодированных входных значений X с соответствующими кубитными векторами эталонных значений переменных социальных процессов. Состояние выходной переменной равно $Y=1$, если входные значения равны кубитным векторам по всем координатам. Это означает, что входные данные в совокупности представляют собой наперед заданную эталонную социальную функциональность. Кубитные векторы в совокупности формируют матрицу эталонных значений

социальной функциональности. Быстродействие получения решения на выходе схемы равно пяти структурным тактам. Следует отметить, что в общем случае состояние выхода схемы формируется функцией принадлежности $Y=m(X,Q)=1-d(X,Q)/n$, которая определена в интервале (0,1) числом (несовпадений) единичных координат на регистровом выходе логической and-функции каждой переменной, где n – число координат Q-вектора, $d(X,Q)$ – кодовое расстояние по Хэммингу. Схема формирует функцию принадлежности элемента или подмножества к наперед заданному эталонному вектору примитивов за один автоматный цикл, благодаря векторной форме задания подмножеств одинаковой размерности. Иначе, схема дает возможность ответить на вопрос: принадлежит ли входная двоичная последовательность X эталонной социальной функциональности Q, также представленной в цифровом виде $Y=1$?

Можно использовать схему с нулевым позитивизмом выходного сигнала, когда максимальная принадлежность входного сигнала X к кубитному эталону Q идентифицируется 0-уровнем различия между двумя векторами $Y=a(X,Q)=d(X,Q)/n$.

Это существенно упрощает логическую структуру. Здесь $Y=0$ означает полное совпадение между входными данными и кубитным вектором эталонной функциональности, что дает основание идентифицировать или классифицировать анализируемый процесс, как принадлежащий эталонному вектору. В результате получается, что функция совпадения (match function) является инверсной по отношению к функции принадлежности (membership function):

$$Y=a(X,Q)=d(X,Q)/n = \text{not}[m(X,Q)=1-d(X,Q)/n].$$

Сигналами в переменных могут выступать: символы, буквы, слова, предложения, цифры, числа, отношения, структуры, рисунки, фотографии, видеофильмы, звуковые фрагменты, процессы и явления.

Следующая схема, рис. 7, еще более упрощает логическое моделирование входных social-потоков данных, где удалены хог-элементы, но добавлены U-элементы, выполняющие роль таблицы универсума примитивных значений каждой переменной, относительно которых синтезируются кубитные векторы переменных для формирования social-паттернов.

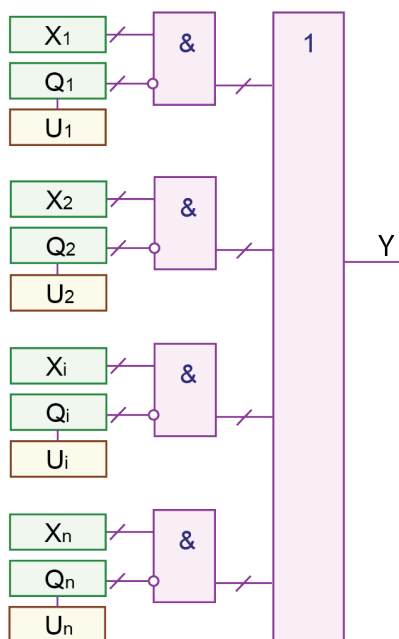


Рис. 7. Упрощенная схема social-simulation

Чтобы показать схему для формирования структур данных social-функциональности (U,Q,X), необходимо соединить входной поток данных S для каждой social-переменной, рис. 8.

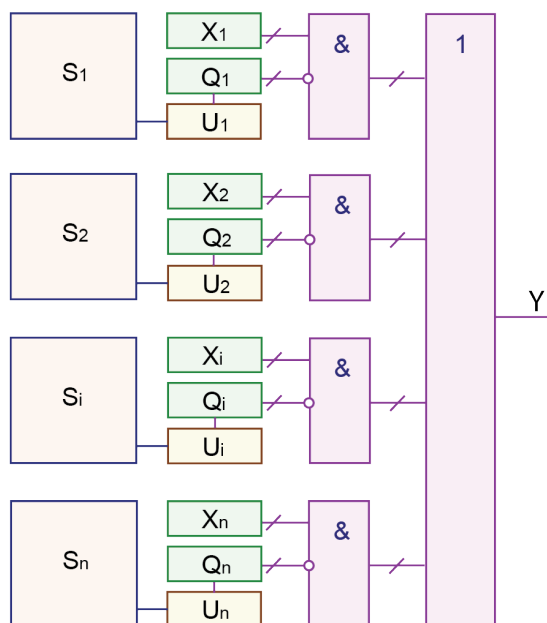


Рис. 8. Схема формирования social-паттерна

Схема для формирования структур данных social-процесса (V,P,Y) соединяет входные потоки данных для переменных F относительно каждой social-функциональности. Примерная UU-модель social-процесса для анализа деятельности компании представлена на рис. 9.

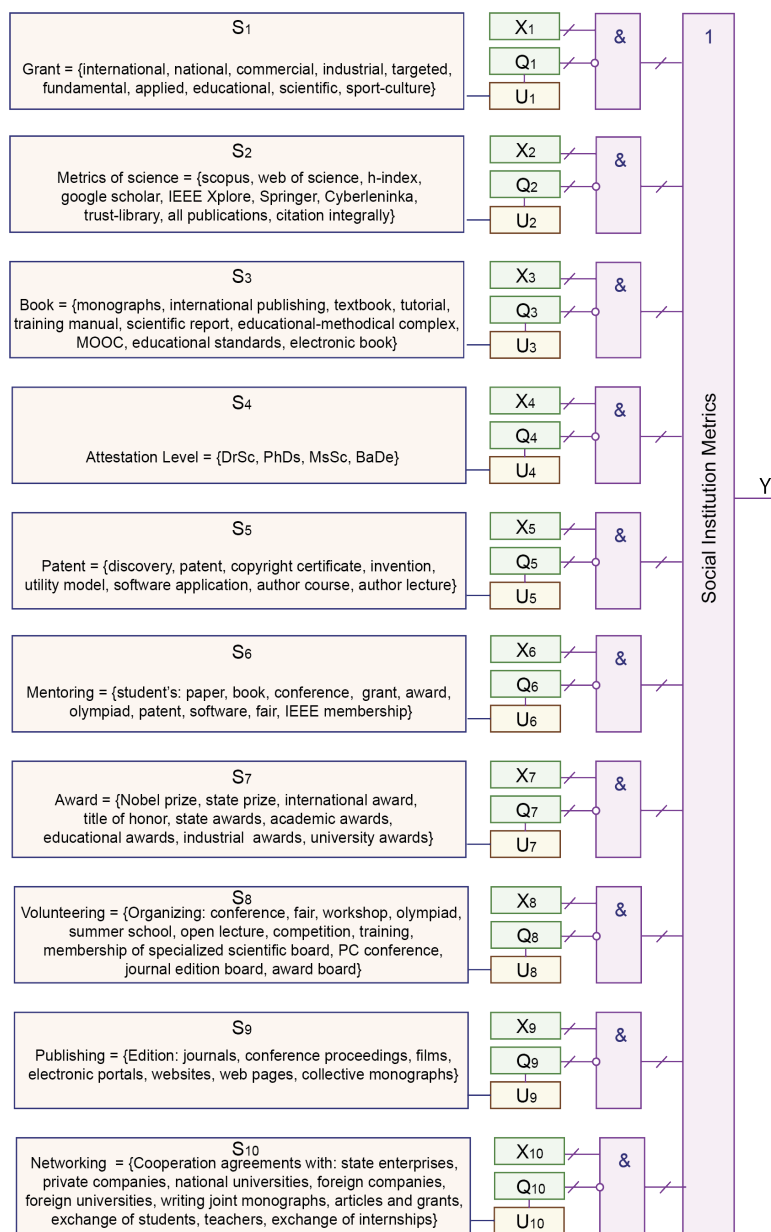


Рис. 9. Метрика-схема social-процесса

Универсум универсумов (UU) представляет собой двухуровневую модель процесса или явления, предназначенную для активного цифрового моделирования в целях точного распознавания больших потоков входных данных путем метрического сравнения с заданными эталонами. Каждый из параметров представлен значениями, которые в совокупности составляют универсум второго уровня. В результате получается UU-метрика функциональности эталонного процесса или явления, относительно которого можно измерять-моделировать поток входных данных, претендующих на данную роль. Для проектирования киберсоциального компьютинга используются два вида моделей логических процессоров.

Кубитно-регистрационная модель (КРМ) оперирует двумя входами логического and-элемента, на которые подаются: 1) кубитный вектор-эталон многозначной переменной social-функциональности; 2) входной вектор данных social-процесса, подлежащего распознаванию. Оба вектора параллельно сравниваются между собой по процедуре $Y = \sqrt{(X \wedge Q) \oplus X}$, включающей три параллельных операции, что приводит к двоичному результату: $Y=0$, если X принадлежит вектору Q , в противном случае $Y=1$.

Кубитно-логическая модель (КЛМ) оперирует одним входом логического and-элемента, на который подается значение-адрес из данных social-процесса, соответствующее данной переменной. По адресу, полученному в результате кодирования текстового фрагмента, определяется координата кубитного вектор-эталона многозначной переменной social-функциональности, значение которой формирует состояние выхода Y для распознавания значения входных данных относительно заданного кубитного вектора, формирующего логический эталон. Другими словами, одна итерация моделирования на кубитном элементе КЛМ-модели social-функциональности определяет принадлежность входного значения к кубитному вектору логического эталона. КРМ модель переменной social-функциональности дает возможность параллельно, за один автоматный цикл, определять принадлежность входного вектора к кубитному вектору логического эталона. В этом их различие. Естественно, что входные воздействия для обеих моделей формируются из текстовых фрагментов входных данных путем их унитарного кодирования на универсальном множестве значений каждой переменной киберсоциальной функциональности.

4. Комбинационные кубитные схемы social-компьютинга

Одним из ярких представителей memory-driven social-компьютинга выступает компания Wave Computing Inc, Silicon Valley, которая соединяет AI-ML с dataflow-based MIPS RISC (конвейер с ограниченной системой команд) architecture. Девиз компании – “follow the data with AI” в целях предоставления deep learning по любому месту нахождения больших данных – от дата центров до терминалов облака [31].

Цель – существенное повышение качества распознавания социальных процессов путем создания логического кубитного процессора параллельного моделирования и распознавания social-паттернов во входных потоках больших данных, полученных на основе метрического мониторинга инфраструктурных компонентов, для цифро-

вого актюаторного управления социальными процессами.

Сущность заключается в синтезе логической схемы social-явления для моделирования (сравнения, распознавания) входного потока данных в целях формирования на выходе процессора актюаторных значений (сигналов), направленных на устранение различий между входными данными и эталоном.

Отличие инновации от существующих решений мониторинга и сравнительного пассивного отображения social-процесса заключается в online генерировании управляющих воздействий, активирующих компьютерные процедуры по приведению social-процесса к идеальному алгоритму или social-траектории.

Исследование поддерживает три фазы social-аналитики (синтез, анализ, управление): 1) Синтез структур данных и логических схем, формирующих кубитные модели social-процесса и social-паттернов на основе анализа входных потоков больших данных – фаза обучения. 2) Анализ или моделирование входных потоков больших данных на синтезированных логических схемах social-паттернов в целях определения принадлежности потоковых фрагментов (распознавания) к заданным social-функциональностям. 3) Управление инфраструктурными киберфизическими компонентами с помощью актюаторных сигналов, полученных в результате анализа входных потоков данных в целях уменьшения временных затрат предоставления сервисов.

Достижение цели дает возможность получить следующие функциональности, практически ориентированные на точное решение задач исчерпывающего мониторинга и цифрового управления social-процессами:

1) Автоматический синтез базисного кубита переменных $P = (P_1, P_2, \dots, P_i, \dots, P_n)$ social-процесса путем поиска или определения универсума непересекающихся классов эквивалентных переменных на основе семантического и синтаксического анализа входных потоков больших данных.

2) Автоматический синтез базисных кубитов значений каждой переменной $Q_i = (Q_{i1}, Q_{i2}, \dots, Q_{ij}, \dots, Q_{im})$ social-процесса путем поиска или определения универсума непересекающихся классов эквивалентных значений для каждой переменной на основе семантического и синтаксического анализа входных потоков больших данных.

3) Автоматический синтез кубитной Q -матрицы универсумов $Q=[Q_{ij}]$ social-процесса на основе объединения кубит-универсумов значений каж-

дой переменной под эгидой кубит-универсума переменных. Q-матрица, как мастер-шаблон, порождает все возможные social-паттерны путем маскирования ее координат значениями {0,1}.

4) Автоматический синтез актуальной кубитной Q-матрицы и логической схемы social-паттерна на основе маскирования кубит-универсумов переменных и их значений путем семантического и синтаксического анализа входных потоков больших данных.

5) Автоматический синтез кубитного логического процессора для моделирования и распознавания social-паттернов во входных потоках больших данных на основе кубитных Q-матриц путем объединения актуальных логических схем social-функциональностей.

6) Автоматический синтез функций выходов логических паттерн-схем для актуаторного управления компонентами инфраструктуры social-процесса на основе моделирования входных потоков больших данных, использующего кубитные Q-матрицы social-функциональностей.

7) Автоматический синтез GUI на основе Q-матрицы дает возможность визуализировать online процедуры моделирования входных потоков данных на логических схемах social-процессора в целях определения состояний выходов, формирующих актуаторные воздействия для инфраструктурных компонентов и работников компании.

Решение перечисленных задач связано с созданием модели, метода и интерпретативного гибкого процессора киберсоциального social-компьютинга, направленного на автоматический синтез и анализ кубитных логических схем, ориентированных на мониторинг, моделирование, распознавание и управление social-процессами и явлениями:

1) Процессор киберсоциального social-компьютинга на основе синтеза кубитных логических схем для мониторинга, моделирования, распознавания и управления social-процессами.

2) Кубитно-матричная модель универсума переменных и универсума значений каждой многозначной переменной для синтеза логических схем social-паттернов, ориентированных на анализ и управление social-процессами.

3) Кубитно-матричный метод синтеза логической схемы для моделирования и распознавания social-функциональностей на основе унитарного кодирования значений многозначных переменных в целях оптимального управления social-процессом.

4) Кубитно-матричный интерпретативный метод моделирования входных потоков social-данных

social-процессов на основе использования эталонных логических элементов social-функциональностей с унитарным кодированием многозначных переменных. Метод отличается от умных блокчейн-контрактов, представляющих собой исполняемый код, интерпретативным анализом гибких структур данных, что дает возможность изменять модель social-процесса в online режиме реального времени.

5) Тестирование и верификация кубитно-матричной архитектуры social-компьютинга на примерах анализа и распознавания процессов во входных потоках больших данных, связанных с киберфизическим отображением и управлением деятельностью компании.

Кубитно-матричная интерпретативная модель social-паттернов и ее унитарно-кодированный экземпляр для моделирования и актуаторного управления social-процессом компании представлены на рис. 10.

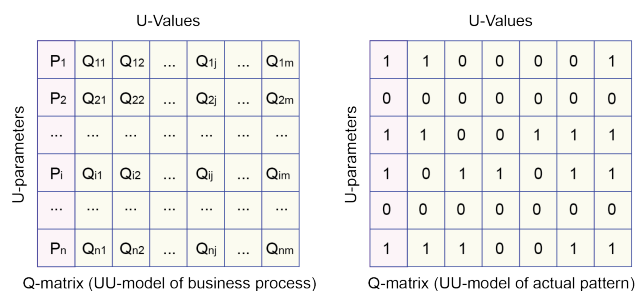


Рис. 10. Кубитная матрица social-процесса

Фактически, Q-матрица представляет собой формальную запись протокола, анкеты или идеального блокчейн-контракта, который дополнительно вырабатывает актуаторные сигналы (заполнить активные поля, поставить подпись, сканировать паспорт, сделать фото, оценить качество услуг), побуждающие человека выполнять определенные действия для достижения цели – ликвидации различий между образцом и реальностью. Таким образом, прямые и непосредственные интеракции между роботом-автоматом и пользователем создают social-компьютинг, использующий Q-матрицу и логические схемы, направленный на получение продуктов или сервисов за минимальное время без участия посредников в виде чиновников, бумажных приказов, положений и законов.

5. Кубитный метод моделирования

Исходные данные: 1) S-data – входной поток больших данных для выполнения social-аналитики. 2) U-матрица – универсум переменных для описания social-процесса с помощью универсумов вербальных значений каждой пере-

менной. 3) X-матрица – векторный формат входных данных по каждой переменной для одной итерации моделирования. 4) Q-матрица – совокупность кубитных двоичных векторов для описания эталонного паттерна по всем переменным. 5) U-вектор – универсум вербальных значений переменной для описания social-паттерна. 6) X-вектор – упорядоченная последовательность двоичных входных данных по одной переменной для выполнения итерации моделирования. 7) Q-вектор – кубитный двоичный вектор для описания эталонного паттерна по одной переменной. 8) Функция принадлежности – численное значение количества различных двоичных одноименных координат в векторах X, Q, определяемая посредством операции $(X \wedge \text{not}Q) = Y$. 9) Актуаторные сигналы – единичные значения координат вектора Y, которые инициируют процедуры, направленные на устранение различий между X-вектором и эталонным Q-вектором.

Структура метода: устройство содержит пять слоев вертикальных компонентов, где первый из них представляет собой входные потоки social-данных, поступающие от сенсорных устройств инфраструктуры компании на триады элементов второго слоя. Здесь первый элемент триады U представляет собой универсум вербальных примитивов-значений каждой переменной, соединенный с элементом Q, который является двоичным кубитным вектором, объединяющим значения переменной для синтеза social-паттерна в формате универсума U. Он также служит шаблоном для формирования двоичных значений кубитного входного вектора данных X, предназначенного для совместного параллельного моделирования с кубит-универсумом Q, которое заключается в их сравнении друг с другом путем выполнения регистровой логической операции and, в третьем слое элементов, над содержимым кубит-вектора X и инверсным состоянием кубит-вектора Q. Это дает возможность получить на выходе and-элемента 0-вектор, если кубит-вектор X принадлежит кубит-универсуму Q, и not-0-вектор – в противном случае, что позволяет использовать полученные двоичные векторы от всех логических and-элементов соответствующих переменных для их логического объединения на or-элементе четвертого слоя. Он формирует интегральный сигнал $Y=0(1)$ для определения принадлежности (непринадлежности) входных векторов данных X к social-паттерну, определяемому кубит-векторами Q всех переменных. Это, в свою очередь, дает возможность формировать с помощью элемента-таблицы из пятого слоя актуаторные сигналы-воздействия, предна-

значенные для устранения противоречий между координатами кубит-векторов Q social-паттерна и входных векторов данных.

Метод: определяет последовательность выполнения операций и процедур для распознавания social-паттернов во входных потоках данных и формирования актуаторных сигналов-воздействий для устранения противоречий между ними. Он начинается с процедуры синтаксического анализа входного потока social-данных, который заключается в извлечении из него 1) универсума ключевых слов или переменных-примитивов, где для каждой переменной извлекается 2) универсум значений, что в совокупности составляет 3) U-матрицу универсума универсумов. Последняя служит платформой для последующего 4) синтеза двух основных двоичных векторов: кубитного Q-вектора и кубитного X-вектора входного потока данных каждой переменной social-паттерна. Далее они используются для выполнения 5) процедуры моделирования, которая заключается в параллельном сравнении координат кубитных векторов $(X \wedge \text{not}Q) = Y$. Это дает возможность получить 6) результирующий вектор $Y = (Y_1, Y_2, \dots, Y_i, \dots, Y_n)$, где каждая 1-координата идентифицирует различие между соответствующими разрядами векторов X и Q. Это позволяет 7) определять функцию принадлежности входного X-вектора к Q-вектору паттерна для каждой переменной, что интегрально, по всем переменным формирует 8) функцию принадлежности входного X-паттерна к эталонному Q-паттерну путем подсчета во всех результирующих Y-векторах 1-координат, которые дают возможность 9) формировать актуаторные сигналы, направленные на устранение противоречий в координатах между актуальным входным вектором X и эталонным паттерном Q. Это является конечной целью social-компьютинга, ориентированного на точное моделирование социальных процессов за счет цифровизации, автоматизации и пространственно-временной оптимизации social-процесса сервисного обслуживания.

Сигналы ветвления в social-паттернах. Актуаторные сигналы формируют ветвления в классической граф-схеме алгоритма управления social-процессом и создают условные процедуры реализации social-паттерна. В данном случае существуют две структуры данных для реализации переменных в social-паттернах, обозначающих условия (X): 1) Каждой X-переменной ставится в соответствие отдельная строка в матрице или кубите универсума переменных. 2) Всем X-переменным ставится в соответствие одна строка в матрице, где все значения различных X-

переменных будут представлены унитарными кодами. Учитывая тот факт, что условные вершины в граф-схеме алгоритма инициируют определенные альтернативные действия, то выходная (логическая) функция для всех условных X-переменных должна быть определена в многозначном алфавите. Это означает, что $Y=\{0,1\}$ является лишь частным случаем при формировании таблицы истинности X-переменной. Общий случай определяется множеством значений, выполняющих актуаторные воздействия на субъект или компонент social-процесса: $Y=\{\text{fill in this gap, please do the signature, try again the spelling, correct the wording, do scanning the passport, press the enter-button, scan the right eye}\}$.

Таким образом, каждая логическая схема social-паттерна имеет многозначные входные переменные и функции выхода, которые формируют функциональное отношение, направленное на инициирование или актуацию субъектов инфраструктуры или пользователей, как реакцию на входной поток больших данных, полученных путем мониторинга social-процесса для их моделирования:

$$Y(Y_1, Y_2, \dots, Y_i, \dots, Y_n) = f(X_1, X_2, \dots, X_i, \dots, X_n).$$

Назначение (многозначной) функции выхода логической схемы social-паттерна заключается в активации процедур social-процесса, направленных на достижение цели в виде получения продукта или сервиса в рамках инфраструктуры киберфизического social-компьютинга.

По существу реакция выхода логической схемы social-паттерна при моделировании входного потока данных представляет собой вектор актуаторных сигналов, направленных на устранение различий между входными данными и social-эталонном.

Логические схемы social-паттернов интегрируются с помощью og-элемента в социальный процессор (рис. 11), который также имеет многозначные входные переменные и функции выхода, формирующие функциональное отношение, направленное на инициирование или актуацию субъектов инфраструктуры или пользователей, как реакцию на входной поток больших данных, полученных путем мониторинга social-процесса для их моделирования:

$$Y(Z_1, Z_2, \dots, Z_i, \dots, Z_n) = f(Y_1, Y_2, \dots, Y_i, \dots, Y_n).$$

Синтез кубитной матрицы social-процесса. Существенной представляется инвариантность social-паттернов относительно времени, что приводит к созданию компактных комбинационных логических схем, где каждый логический эле-

мент отвечает за все используемые значения существенной переменной. Базовым является вариант, где имеется большое количество скриншотов, которые создают последовательности логических элементов, затрудняющие анализ social-процессов для выработки управляющих актуаторных сигналов. Последовательные цепочки экранов приводят к необходимости группирования экранов в рамках формирования модели комбинационной схемы social-паттерна.

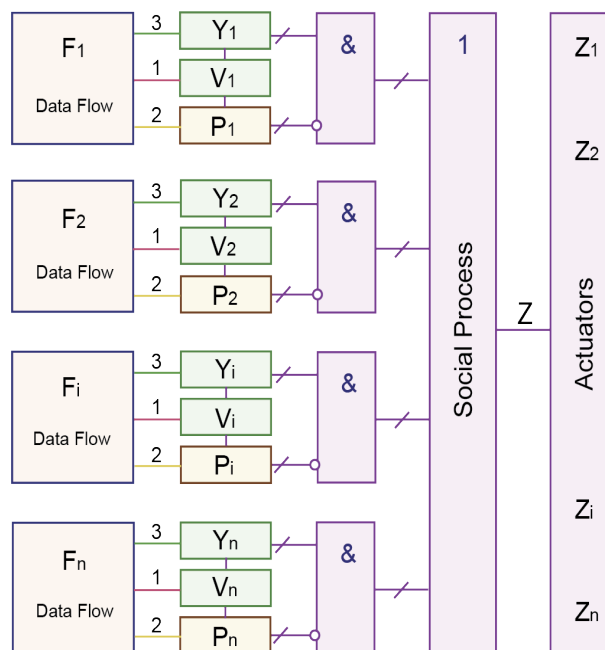


Рис. 11. Логическая схема social-процесса

Таким образом, имеются два пути синтеза social-паттерн-схемы: 1) На основе анализа social-потока сформировать универсум значений логических переменных и кубитных векторов social-паттерна, к которым применить алгоритм моделирования для распознавания потока данных. 2) На последовательности фреймов выполнять процедуру группирования экранов, используя неизменяемость существенных переменных на временном интервале в целях синтеза комбинационной логической схемы, которая интегрирует отношения между значениями существенных переменных во времени и пространстве. Social-процесс, social-паттерны, social-переменные и social-значения автоматически (вручную) определяются стратегией нисходящего проектирования: от ключа-сущности к синонимам путем формирования непересекающихся классов эквивалентностей, соответствующих переменным, которые в совокупности создают универсумы примитивов.

Синтез или извлечение универсумов. Система классов-переменных обладает следующими свойствами: 1) Образует разбиение. 2) Любые два элемента из одного класса эквивалентны [1]. 3) Любые два элемента из разных классов не эквивалентны. 4) Объединение классов создает универсум примитивов – значений, переменных, social-паттернов. 5) Пересечение классов равно пустому множеству. Следующая таблица иллюстрирует взаимодействие классов эквивалентных подмножеств множества P:

P	a b c	d e	f x y	z
a	1 1 1			
b	1 1 1			
c	1 1 1			
d		1 1		
e		1 1		
f			1 1 1	
x			1 1 1	
y			1 1 1	
z				1

Процедура синтеза классов переменных или значений, имеющих отношения эквивалентностей на основе разбиения входного потока данных, содержит следующие пункты:

1) Выбор элемента $a_1 \in P$ и образование подмножества (класса) $K_1 \subset P$, состоящего из a_1 и всех элементов, эквивалентных ему:

$$\forall a_1 \in P \exists K_1 \subset P: K_1 = [a_1] = \{x \in P: x \sim a_1\}.$$

2) Выбор элемента $a_2 \in P$, $a_2 \neq a_1$ и образование класса $K_2 \subset P$, состоящего из a_2 и всех элементов, эквивалентных ему:

$$\forall a_2 \in P, a_2 \notin K_1 \exists K_2 \subset P: K_2 = [a_2] = \{x \in P, x \notin K_1: x \sim a_2\}.$$

3) В результате получается система эквивалентных классов, объединение которых равно множеству P.

Здесь множество актуаторных воздействий формирует универсум примитивов для каждой переменной, из которых создаются кубитные логические функциональности для управления киберсоциальными процессами и явлениями, используемыми компараторные процедуры или алгоритмы для сравнения с эталонами. Универсум актуаторных примитивов дает возможность управлять переменной с помощью синонимов, каждый из которых активизирует процесс или явление. Например, переменная активизации движения произвольного объекта имеет следующие варианты, составляющие универсум: {поехали, двинулись, покатались, поскакали, полетели, помчались, потащились, направились, погнались, побежали, пошли, стартанули}. Обязательным

условием в подборе синонимов для переменной является их непересекаемость с универсумами других переменных во избежание неоднозначности поведения объекта. Естественно, что переменные активизации своими универсумами создают команды для управления процессами, связанными с человеком, животным, автомобилем, роботом, дроном, компьютером, оружием, техникой, домом, рабочим местом.

Таким образом, логическая схема для анализа социального social-процесса или явления способна верифицировать созданную метрику или структуру, моделировать позитивные или негативные решения руководителя, предсказывать социальные явления в будущем, включая деструктивные акции, катаклизмы, коллизии, взлеты и падения граждан, социальных групп и государств. Социальный social-компьютинг, как и универсальный вычислитель, способен определять точные решения с помощью логических схем эталонного поведения человека, социальной группы или государства. Проблема, подлежащая решению в будущем, заключается в создании базы алгоритмов или схем, из которых можно технологически просто синтезировать киберсоциальные метрические social-процессоры для мониторинга, анализа и актуаторного управления медициной, транспортом, наукой, образованием, государством, производством, экологией, юриспруденцией, финансами. Другое дело, что чиновники всех уровней будут препятствовать проникновению неподкупной и объективной цифровизации в сферу их деятельности по понятным причинам. Чем выше киберсоциальная невежественность бюрократического аппарата, тем сложнее и длиннее путь к процветанию государства и его граждан в современном мире. Путь один, длительный и эволюционный – перманентно учить будущую и актуальную политическую элиту элементам нравственной киберкультуры управления на основе исчерпывающего метрического мониторинга социальных процессов.

6. Матричный секвенсор для распознавания социальных явлений

Проблема и решение. Проблема киберсоциального компьютеринга заключается в сложности формализации эмоциональной логики человеческих отношений, которые необходимо привести к цифровому детерминизму функциональных соответствий, исключая вероятность и неопределенность. Уйти от эвристики в сторону автоматизации синтеза и анализа логических схем социального процессинга для моделирования и предсказания social-коллизий от некомпетентности

тентных решений – задача, актуальная для рынка. Для ее решения предлагается трансформировать функционально законченный поток больших данных к структурированной матричной двоичной форме, которая унитарно кодирует множество ключевых понятий-переменных и все возможные их значения, составляющие универсум примитивов. Метод для решения проблемы представлен синтезом классов эквивалентных отношений на совокупности переменных P:

$$P = \{P_1, P_2, \dots, P_i, \dots, P_j, \dots, P_n\}, P_i \cap P_j = \emptyset,$$

где каждая из них $P_i = \{P_{i1}, P_{i2}, \dots, P_{ik}, \dots, P_{ir}, \dots, P_{in}\}$ принимает универсальное множество вербальных значений, создающих между собой эквивалентные отношения $P_{ik} \sim P_{ir}$ и образующих при этом пустые пересечения $P_{ik} \cap P_{ir} = \emptyset$. Таким образом, чтобы построить непересекающиеся классы эквивалентных вербальных значений конечного множества переменных, необходимо определить ключевые слова, которые не должны иметь между собой смыслового пересечения, а затем для каждого ключевого слова выбрать из текста синонимы, которые не должны присутствовать в значениях других переменных.

Метрика: определения, аксиомы и уравнение social-компьютинга

1) Метрика есть способ измерения расстояний $d_i \in D$ между процессами и явлениями в пространстве заданных параметров с выполнением аксиомы циклического конволюционного замыкания [2]:

$$D = \sum_{i=1}^n d_i = 0$$

2) Измерение есть процедура определения расстояния между конечным множеством процессов или явлений, отличных от нуля.

3) Параметрами выступают логические переменные $P = \{P_1, P_2, \dots, P_i, \dots, P_n\}$, которые с заданной степенью адекватности описывают процесс или явление.

4) Переменные определяются посредством их значений $P_i = \{P_{i1}, P_{i2}, \dots, P_{ij}, \dots, P_{in}\}$, число которых не может быть меньше двух.

5) Матрица универсумов есть упорядоченная совокупность переменных и их значений $U = [P_{ij}] = [U_{ij}]$ для метрического измерения процесса или явления. Унитарно кодированная матрица U имеет единичные значения по всем координатам. Матрица универсумов U может быть представлена одним вектором путем конкатенации ее строк

$$P = (P_1 * P_2 * \dots * P_i * \dots * P_n).$$

6) Матрица паттерна $Q = [Q_{ij}]$ есть подмножество значений переменных универсальной матрицы $Q \in U$, которое формирует образец конкретного процесса или явления. Наиболее удобной формой задания паттерна является матрица унитарного двоичного кодирования значений переменных.

7) Матрица входных данных $X = [X_{ij}]$ есть подмножество (двоичных) значений переменных универсальной матрицы $X \in U$, которое формирует фрагмент реального процесса или явления.

8) Матрица измерения $Y = [Y_{ij}] = [Q_{ij}] \oplus [X_{ij}]$ есть подмножество $Y \in U$ (двоичных) значений переменных универсальной матрицы, формирующее различия в одноименных координатах матриц Q и X путем выполнения параллельной хог-операции между паттерном и фрагментом реального процесса или явления.

9) Расстояние есть скалярная оценка количества различий, отмеченных единицами, в одноименных координатах матриц измеряемых процессов или явлений. Расстояние определяется путем подсчета единичных координат в матрице измерения:

$$d(Q, X) = \sum_{j=1, m}^{i=1, n} Y_{ij}.$$

10) Функция различия есть отношение расстояния (количества различий в одноименных координатах) к общему числу координат матрицы (измерения):

$$\mu = \frac{d(Q, X)}{n \times m} = \frac{\sum_{j=1, m}^{i=1, n} Y_{ij}}{n \times m}.$$

Функция различия μ не дифференцирует отношения между матрицами X, Q, которые качественно необходимо разделить на 5 вариантов взаимодействия двух множеств (векторов, матриц) (рис. 12).

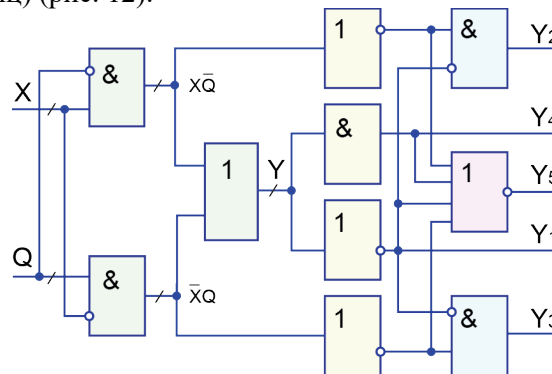


Рис. 12. Матричный метрический процессор
Здесь левая часть схемы состоит из трех элементов $Y = XQ \vee \bar{X}\bar{Q}$ и представляет собой регистровую или матричную реализацию хог-операции, где регистровые выходы всех трех элементов далее используются для формирования уже булевых выходов, которые определяют: 1) пере-

менную $Y_2 = [\sqrt{X\bar{Q}}]$, равную 1, если $X \subset Q$; 2) переменную $Y_3 = [\sqrt{\bar{X}Q}]$, равную 1, если $X \supset Q$; 3) переменную $Y_4 = [\wedge Y] = [\wedge(X\bar{Q}\sqrt{\bar{X}Q})]$, равную единице, если результат фиксирует несовпадения (единицы) по всем координатам выхода Y ; 4) переменную $Y_1 = [\vee Y] = [\vee(X\bar{Q}\sqrt{\bar{X}Q})]$, равную единице, если результат фиксирует совпадения (нули) по всем координатам вектора Y ; 5) переменную $Y_5 = (\bar{Y}_1\vee Y_2\vee Y_3\vee Y_4)$, равную единице, если результат фиксирует нулевые значения булевых переменных $Y_1, Y_2, Y_3, Y_4 = 0$, что означает наличие частичного пересечения между компонентами двух векторов или матриц. Средняя часть схемы представлена четырьмя элементами, которые преобразуют регистровые переменные в булевы по правилам операций and, or. Последний элемент or-not выполняет функцию определения самого сложно-формализуемого взаимодействия между векторами $X \cap Q \neq \emptyset$, методом исключения, когда все предыдущие четыре отношения не выполняются. Интегрально все пять выходных функций формируют метрику измерения качества взаимодействия пары процессов и явлений, представленных в формате матриц или векторов:

$$Y = X\bar{Q}\sqrt{\bar{X}Q};$$

$$Y_1(X = Q) = [\vee Y] = [\vee(X\bar{Q}\sqrt{\bar{X}Q})];$$

$$Y_2(X \subset Q) = [\sqrt{X\bar{Q}}];$$

$$Y_3(X \supset Q) = [\sqrt{\bar{X}Q}];$$

$$Y_4(X \cap Q = \emptyset) = [\wedge Y] = [\wedge(X\bar{Q}\sqrt{\bar{X}Q})];$$

$$Y_5(X \cap Q \neq \emptyset) = (\bar{Y}_1\vee Y_2\vee Y_3\vee Y_4).$$

Последнее равенство вычисляет самое сложное взаимодействие двух матриц или векторов по методу исключения: если все предыдущие четыре переменные не равны единицам, то это означает, что два объекта имеют частичное пересечение друг с другом.

Вычислительная сложность определения качества метрического взаимодействия двух векторов или матриц равна: $Q=5Q(Y)+5Q(Y_i)$, где первое слагаемое формирует время выполнения пяти регистровых логических операций, а второе – время реализации пяти булевых функций.

Следующие таблицы иллюстрируют вычислительный процесс для определения качества взаимодействия между различными парами двоичных векторов X, Q , где первые 4 колонки представлены векторами, а остальные представляют собой двоичные бинарные скалярные результаты

X	Q	X \bar{Q}	$\bar{X}Q$	Y	Y ₄	X	Q	X \bar{Q}	$\bar{X}Q$	Y	Y ₂	X	Q	X \bar{Q}	$\bar{X}Q$	Y	Y ₃	X	Q	X \bar{Q}	$\bar{X}Q$	Y	Y ₅
1	0	1	0	1	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	
0	1	0	1	1		0	1	0	1	1		0	0	0	0	0		0	1	0	1	1	
0	1	0	1	1		1	1	0	0	0		1	1	0	0	0		1	1	0	0	0	
0	1	0	1	1		1	1	0	1	1		1	1	0	0	0		1	0	1	0	1	
1	0	1	0	1		0	0	0	0	0		1	0	1	0	1		1	0	1	0	1	
1	0	1	0	1		0	1	0	1	1		1	0	1	0	1		1	1	0	0	0	

{0,1} взаимодействия векторов.

Решить проблему неоднозначности при идентификации отношения эквивалентности между векторами X, Q можно с помощью двух элементов коррекции состояния выходов Y_2, Y_3 . Таким образом, модифицированный процессор дает возможность однозначно идентифицировать все возможные виды взаимодействия двух матриц (векторов, множеств, текстов) при распознавании паттернов, образов, процессов или явлений. Матричный процессор дает возможность уйти от необходимости считать кодовое расстояние в четырех из пяти случаев качественного взаимодействия матриц или векторов. Остается только один вариант, при $Y_5=1$, для уточнения взаимодействия векторов X, Q путем подсчета функции различия $\mu = \frac{d(Q,X)}{n \times m}$. Это необходимо для последующего корректирования матриц Q, X в целях приведения одной из них к другой, когда их отношения трансформируются в одну из четырех форм: $X = Q; X \subset Q; X \supset Q; X \cap Q = \emptyset$.

Практическая направленность функции различия связана с получением скалярных оценок взаимодействия матриц или векторов по всем видам отношений:

1. Параметры U-матрицы универсумов, количество переменных и сумма всех значений каждой переменной, что в совокупности создает абсолютную оценку мощности актуальных значений унитарной матрицы и коэффициент использования координат U-матрицы:

$$N(U) = \sum_{j=1,m}^{i=1,n} U_{ij};$$

$$K(U) = \frac{N(U)}{n \times m} = \frac{\sum_{j=1,m}^{i=1,n} U_{ij}}{n \times m}.$$

2. Оценка мощности единичных значений координат унитарной Q-матрицы и коэффициент использования значений U-матрицы универсумов:

$$N(Q) = \sum_{j=1,m}^{i=1,n} Q_{ij};$$

$$K(Q) = \frac{N(Q)}{N(U)} = \frac{\sum_{j=1,m}^{i=1,n} Q_{ij}}{\sum_{j=1,m}^{i=1,n} U_{ij}}.$$

3. Оценка мощности единичных значений координат унитарной X-матрицы, коэффициент отношения мощностей единичных значений и функция различия между X- и Q-матрицами для вариантов взаимодействия

$$Y_1((X = Q), (X \subset Q), (X \supset Q), (X \cap Q = \emptyset), (X \cap Q \neq \emptyset));$$

$$N(X) = \sum_{j=1,m}^{i=1,n} X_{ij}.$$

$$K(X) = \frac{N(X)}{N(Q)} = \frac{\sum_{j=1, m}^{i=1, n} X_{ij}}{\sum_{j=1, m}^{i=1, n} Q_{ij}},$$

$$\mu = \frac{d(Q, X)}{n \times m}.$$

Скалярные оценки взаимодействия предоставляют численные доказательства для качественных утверждений: «совпадает», «включает», «покрывает», «не пересекается», «различается». При этом макроструктура social-компьютинга может быть представлена xor-взаимодействием матриц X, Q, где затем последовательно измеряется качественная и количественная оценки их отношений, что представлено на рис. 13.

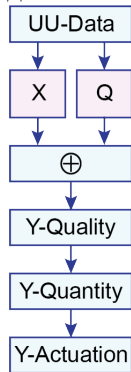


Рис. 13. Макроструктура social-компьютинга

11) Функция принадлежности есть величина, обратная функции различия:

$$\mu' = 1 - \mu = 1 - \frac{d(Q, X)}{n \times m}.$$

12) Распознавание есть процесс моделирования для определения различий между матрицей входных данных и матрицами эталонных паттернов путем выполнения параллельной xor-операции в целях получения матриц измерений и функций различия для каждого паттерна. Процесс моделирования матрицы входных данных X по матрице паттерна Q, в целях вычисления матрицы измерения Y и функции μ , представлен двумя формулами:

$$[Y_{ij}] = [Q_{ij}] \oplus [X_{ij}], \mu = \sum_{j=1, m}^{i=1, n} Y_{ij}.$$

13) Модель M матричного киберсоциального social-компьютинга (распознавания паттернов) использует матрицу U универсумов переменных и их значений для формирования метрического xor-взаимодействия триады производных матриц (рис. 14): входных данных X, паттернов Q и измерения (ошибок) Y, которое оценивается скалярными функциями различия μ :

$$M = \langle U, Q, X, Y, \mu \rangle, Q \oplus X \oplus Y = 0, U = [U_{ij}]$$

$$Q \subseteq U, X \subseteq U, Y \subseteq U, \mu = f\{(Q, X), (Q, Y), (X, Y)\}.$$

14) Циклическое конволюционное замыкание есть неотъемлемое дополнение метрики, предназначенное для проверки валидности измерения процессов или явлений. Конволюция превращает множество всех подмножеств универсума при-

митивов (идентификаторов), а также расстояния между элементами-вершинами цикла (идентификаторов) в пустое множество (ноль) при их обработке с помощью операции симметрической разности (xor). Симметрическая разность (xor) всех элементов (двоичных кодов) и их универсума равна пустому множеству (нулю).

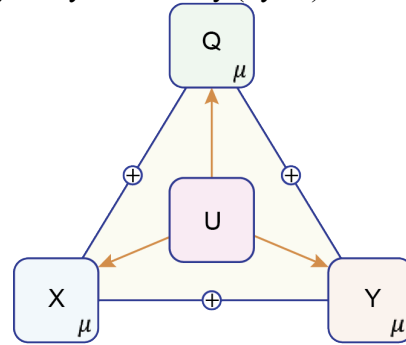


Рис. 14. Модель отношений матричного киберсоциального social-компьютинга

15) Метрическое конволюционное транзитивное взаимодействие трех компонентов процесса или явления составляет сущность уравнения киберсоциального (бизнес) компьютеринга:

$$Q \oplus X \oplus Y = 0 \rightarrow Y = Q \oplus X, X = Q \oplus Y, Q = Y \oplus X.$$

Например, далее приводится конволюционное xor-взаимодействие трех матриц Q, X, Y:

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \oplus \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \oplus \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \oplus \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Очевидно, что по двум известным матрицам можно восстановить любую третью, например, по Q, Y получается матрица X с помощью xor-операции.

Таким образом, уравнение киберсоциального обратимого social-компьютинга дает возможность параллельно, с помощью одной xor-операции, за один автоматный такт получать любой из трех компонентов social-процесса при двух известных.

Основные преимущества кубитного метода: 1) Высокое быстродействие social-компьютинга за счет параллельной обработки однородных двоичных двумерных данных с помощью двух матричных операций: xor, cardinale. 2) Простота архитектуры social-компьютинга, которая определяется xor-взаимодействием трех матриц одинаковой размерности. 3) Универсальность архитектуры xor-взаимодействия матриц способна решать широкий спектр задач social-компьютинга, киберсоциального компьютеринга, online управления движущимися объектами, распознавания об-

целях устранения различий между ними и начинается с блока 1) инициирования процедуры смыслового 2) анализа входных данных S, поступающих от сенсоров F цифровой инфраструктуры, при этом выполняется 3) проверка наличия U-matrix в целях перехода на блок 6, а в случае ее отсутствия активирует процедуру разбиения текстовых фрагментов на непересекающиеся классы эквивалентностей для 4) извлечения универсума ключевых слов или переменных-примитивов, где для каждой из них определяется универсум значений, что в совокупности формирует U-матрицу универсума универсумов, которая служит шаблоном для последующего 5) синтеза кубитной Q-матрицы social-паттерна и 6) кубитной X-матрицы входных данных, которые далее используются для выполнения 7) процедуры моделирования, которая заключается в выполнении &-операции параллельного сравнения координат кубитных матриц $(X \wedge \text{not}Q) = Y$, что дает возможность получить результирующую матрицу $Y = [Y_{ij}]$, где каждая 1-координата идентифицирует различие между соответствующими рядами матриц X и Q, что дает возможность 8) определять численное значение функции принадлежности μ входной X-матрицы к Q-матрице social-паттерна путем подсчета в Y-матрице 1-координат, которые дают возможность 9) формировать вербальную W-матрицу актуаторных сигналов, направленную на устранение различий в координатах между актуальной X-матрицей входных данных и Q-матрицей эталонного паттерна, если 10) функция принадлежности больше нуля, путем выработки управляющих воздействий A для активации 11) инфраструктуры E, сенсоры которой связаны обратной связью с блоком входных данных S, а при отсутствии различий между матрицами Q и X, когда $\mu=0$, выполняется переход к блоку 12) окончания работы алгоритма, целью которого является social-компьютинг, ориентированный на получение дополнительной прибыли компании за счет цифровизации, автоматизации и пространственно-временной оптимизации social-процесса при создании сервиса или продукта.

Логическая структура, представленная на рис. 17, характеризуется конечным множеством μ -матриц, вычисляемых параллельно, которое формирует универсум функций принадлежности очередного входного воздействия паттернам, среди которых выбирается минимальная оценка различий.

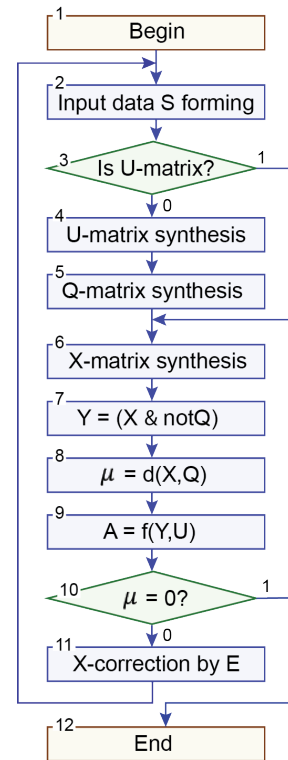


Рис. 16. Алгоритм social-компьютинга

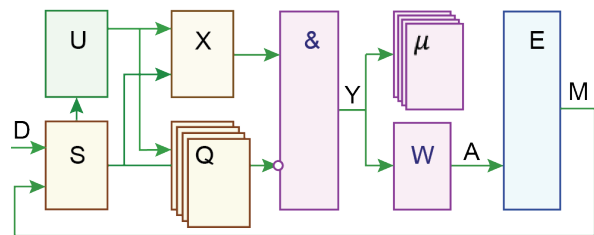


Рис. 17. Логическая матричная схема social-процесса
Логическая структура, представленная на рис. 18, характеризуется компьютерной архитектурой social-аналитики, которая имеет все восемь компонентов модели универсального вычислителя, взаимодействующих между собой по формуле: вход R иницирует команды для выполнения social-процесса, который начинается с получения D-ресурсов: финансовых, кадровых, информационных, поступающих на исполнительный механизм E. Это активирует сенсоры, которые передают информацию по шине мониторинга M для последующего формирования потока S-данных, предназначенного для синтеза U-матрицы универсума переменных для описания social-процесса с помощью универсумов вербальных значений каждой переменной. Последняя служит базовым форматом для синтеза X-матрицы векторов входных данных по каждой переменной для выполнения итерации моделирования относительно Q-матрицы, задающей множество кубитных двоичных векторов для описания эталонного social-паттерна по всем переменным. Это дает возможность на основе &-

операции вычислять функцию принадлежности μ , имеющую выход визуализации состояния V бизнес процесса и соответствующую Y , в виде численного значения количества различных двоичных одноименных координат, полученного при сравнении матриц X и Q , определяемом посредством параллельной операции $(X \wedge \text{not} Q) = Y$. Она дает возможность синтезировать актуаторные вербальные сигналы W , соответствующие единичным значениям координат выходной матрицы Y в формате U -матрицы, которые по шине A иницируют вычислительные процедуры, направленные на устранение различий между X -матрицей и эталонной Q -матрицей social-паттерна путем коррекции координат X -матрицы с помощью инфраструктуры E , обеспечивающей исполнение social-процесса компании для получения сервисов или продукции P .

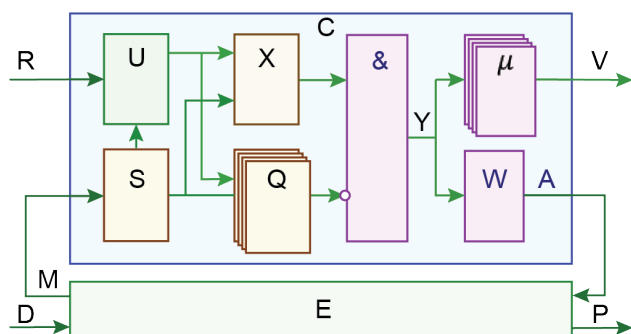


Рис. 18. Логическая матричная параллельная схема social-компьютинга

Алгоритм, соответствующий архитектуре, представленной на рис. 18, характеризуется использованием кубитных матриц данных для выполнения операций social-компьютинга. Его структура содержит механизмы управления C и исполнения E , соединенные шинами мониторинга M и актуации A , где блок управления имеет внешний вход команд R и выход визуализации состояния social-процесса V , а блок исполнения содержит внешний вход данных D и выход сервисов или продукции P . Это в совокупности составляет восемь компонентов, характеризующих social-компьютинг. Он начинается с определения входного потока данных S , поступающего от сенсоров исполнительного механизма, который обслуживает social-процесс компании и предназначен для синтеза U -матрицы универсума переменных и описания social-процесса с помощью универсумов вербальных значений каждой переменной, которая служит базовым шаблоном для синтеза X -матрицы входных данных и последующего моделирования совместно с уже определенными Q -матрицами, задающими множество эталонных social-паттернов. Это дает возмож-

ность на основе $\&$ -операции параллельно вычислять n матриц $Y_i = (X \wedge \text{not} Q_i)$, $(i=1, n)$ моделирования, анализ которых на минимальное число единичных координат $\min(Y_i=1)$, $(i=1, n)$, дает возможность определить номер i матрицы, имеющей минимальное значение из n функций принадлежности

$$\mu = \min_i \mu_i \leftarrow \mu_i = \sum_{j=1, k}^{r=1, m} Y_{ijr},$$

Которая формирует численные значения различных двоичных одноименных координат, полученных при сравнении матрицы X и n матриц Q . Это дает возможность синтезировать матрицу актуаторных вербальных сигналов W , соответствующую единичным значениям координат выходной матрицы Y , которая имеет $\min \mu$, в формате вербальных значений U -матрицы, иницирующие вычислительные процедуры в исполнительном механизме E , направленные на устранение различий между X -матрицей и Q -матрицей social-паттерна с $\min \mu$, путем коррекции координат X -матрицы, обеспечивающей оптимальное исполнение цели P social-процесса компании.

Алгоритм, представленный на рис. 19, определяет последовательность процедур компьютеринга во времени и пространстве для распознавания social-паттернов во входных потоках данных и выработки актуаторных сигналов-воздействий в целях устранения различий между ними. Это начинается с блока 1) иницирования алгоритма с последующим переходом к 2) блоку смыслового анализа входных данных S , поступающих от входа D и сенсоров M цифровой инфраструктуры. При этом выполняется 3) проверка наличия U -matrix в целях перехода на блок 6. В случае ее отсутствия алгоритм активирует процедуру разбиения текстовых фрагментов на непересекающиеся классы эквивалентностей для 4) извлечения универсума ключевых слов или переменных-примитивов, где для каждой из них определяется универсум значений. Это в совокупности формирует U -матрицу универсума универсумов, которая служит шаблоном для последующего 5) синтеза кубитных Q -матриц social-паттернов и 6) кубитной X -матрицы входных данных. Далее они используются для выполнения 7) параллельной $\&$ -операции и получения n матриц $Y_i = (X \wedge \text{not} Q_i)$, $i=1, n$ моделирования, в которых подсчитывается число единичных координат $\min(Y_i=1)$, $i=1, n$. Это дает возможность 8) определить номер i матрицы, имеющей минимальное значение из n функций принадлежности

$$\mu = \min_i \mu_i \leftarrow \mu_i = \sum_{j=1, k}^{r=1, m} Y_{ijr},$$

которые формируют численные значения различных двоичных одноименных координат, полу-

ченных при сравнении матрицы X и n матриц Q. Это позволяет 9) синтезировать матрицу актюаторных вербальных сигналов W, соответствующую единичным значениям координат выходной матрицы Y, имеющей $\min \mu$, в формате вербальных значений U-матрицы, которые, 10) при $\mu \neq 0$, инициируют 11) вычислительные процедуры в исполнительном механизме E, направленные на устранение различий между X-матрицей и Q-матрицей social-паттерна с $\min \mu$, путем коррекции координат X-матрицы, обеспечивающей оптимальное исполнение цели P social-процесса компании, а при существовании $\mu \neq 0$ выполняется завершение работы алгоритма на заданной матрице входных воздействий, который обеспечивает реализацию social-компьютинга, ориентированного на получение дополнительной прибыли компании за счет цифровизации, автоматизации и пространственно-временной оптимизации social-процесса при создании сервиса или продукта.

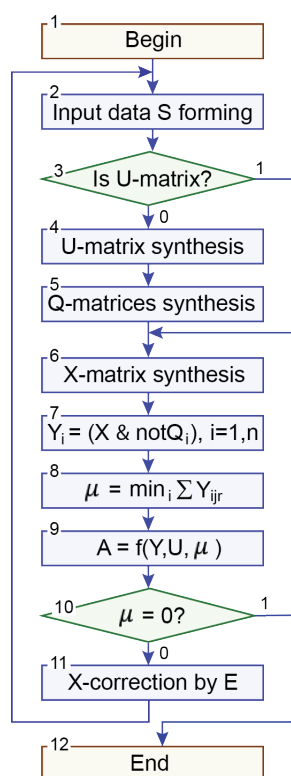


Рис. 19. Алгоритм social-компьютинга

Таким образом, матричная структура данных для параллельного выполнения операций social-компьютинга дает возможность существенно повысить быстродействие интерпретативного параллельного моделирования входных потоков данных для распознавания в них social-паттернов в целях генерирования актюаторных сигналов, устраняющих различия в X-матрице по отношению к одному из эталонных social-паттернов.

7. Заключение

Научная новизна публикации заключается в разработке кубитного логического секвенсора для моделирования и распознавания social-явлений в потоках больших данных на основе создания интерпретативных кубитных матричных моделей, методов и архитектур киберсоциального social-компьютинга, направленного на автоматический синтез и анализ логических схем, ориентированных на анализ, мониторинг и управление социальными процессами и явлениями:

1) Усовершенствована архитектура memory-driven киберфизического компьютеринга, которая отличается параллелизмом процедур синтеза и анализа логических схем, предназначенных для моделирования social-процессов и явлений в целях мониторинга и управления.

2) Усовершенствованы кубитно-матричные модели логических схем, которые отличаются унитарным кодированием многозначных переменных для синтеза секвенсоров в целях параллельного анализа киберсоциальных процессов.

3) Впервые предложены кубитно-матричные векторные методы синтеза и анализа логических схем, которые характеризуются унитарным кодированием значений многозначных переменных для параллельного моделирования social-процессов и явлений.

4) Разработанные кубитно-матричные модели, структуры данных, методы синтеза и анализа логических схем social-процессов позволяют моделировать реакцию социальной системы на принятие конструктивных и деструктивных решений человека, руководителя, чиновника, благодаря кубитному описанию эталонов поведения, что дает возможность актюаторно управлять гражданами и инфраструктурой.

5) Направления будущих исследований. Так же как биологические вирусы деструктурируют человека, социальные вирусы (коррупция, воровство, терроризм, загрязнение планеты, бунты, революции, войны) поражают организм человечества в масштабах планеты, унося в мир иной миллионы жизней. Причиной этому всегда является безнравственность и некомпетентность политической элиты государственных образований, которая формирует социальные болезни «грязных рук». Если учесть, что исцелить политическую элиту можно путем длительного процесса ее «санитарного» образования, необходимо искать альтернативные технологии борьбы против социальных вирусов. Одним из возможных вариантов может быть социальный иммунитет, как киберфизический нравственный социальный компьютеринг метрического исчерпывающего мо-

мониторинга всех процессов и явлений для цифрового human-free управления гражданами на основе моделирования и предсказания последствий от принятия решений. Компьютинг всегда, везде и во всем объясняет природу процессов и явлений, помогает решать актуальные вопросы гармонического и нравственного развития человечества. Вот несколько примеров компьютеринга. Вселенная имеет гармонический геном циклического изменения в метрике: материя и энергия, пространство и время – Cosmological Computing. Человечество также стремится к гармоническому изменению нравственных отношений для достижения социальной справедливости путем цифрового мониторинга и автоматического управления при наличии ресурсов в метрике: материя и энергия, пространство и время – Humanity Computing. Геном человека формирует гармонический полцикл изменения путем зарождения, развития, старения и умирания в метрике ресурсов: материя и энергия, пространство и время – Human Computing. Геном компьютеринга также имеет гармоническую форму развития технологий, элементной базы, системных, алгоритмических, программных и архитектурных решений в метрике: материя и энергия, пространство и время – HW-SW Computing. Кибермедицинский компьютеринг (КМК) – пожизненный мониторинг душевного и физического здоровья каждого человека с момента его рождения в целях активного управления его поведением в формате 24/7 для предотвращения болезней путем создания цифрового ассистента, помогающего принимать оптимальные решения по стратегии и тактике поведения для обеспечения высокого качества жизни. КМК является альтернативой для стратегии современной медицины, заключающейся в лечении болезней, полученных в результате неправильного выбора повседневных и долговременных решений, которые связаны с незнанием функциональных особенностей своего организма и влияния на него окружающей действительности. Предотвращать болезни путем моделирования возможных вариантов поведения, а не объяснять, почему они случились, обеспечивать качество жизни, а не качество лечения на основе перманентного метрического мониторинга состояния души, тела и окружающей среды в целях цифрового оптимального управления поведением человека. При этом коррекция природных ошибок и полученных травм является лишь полезным дополнением к средствам обеспечения качества жизни человека. Биокомпьютинг, как мониторинг и управление биологическими процессами, является наиболее существенной обла-

стью в познании жизни (синтез и анализ), которая наименее изучена учеными и практиками. Аддитивный компьютеринг представляет собой безотходную технологию выращивания или 3(4)D-печатания компьютеров и компонентов, технических конструкций, домов, продуктов питания, биоинженерных частей человеческого тела.

б) Отдельные выводы. Принимая во внимание дизапторную аксиому, что компьютеринг, как процесс, является первичным по отношению к явлениям, которые он объясняет, порождает, обслуживает и использует, можно сделать несколько существенных практически ориентированных выводов. Вместо универсального целевого критерия эффективности: время-деньги-качество вводится измерение процесса или явления в метрике двух взаимодействующих пар: пространство-время, материя-энергия. Любой процесс направлен на изменение явления в метрике параметров пространство-время, материя-энергия для достижения цели. Процесс реализуется, реально или виртуально, в архитектуре или модели компьютеринга, который определяется восемью взаимодействующими компонентами: цель, отношения, визуализация, управление, исполнение, ресурсы, мониторинг, актуация. Любой социальный процесс может и должен быть реализован в формате компьютеринга, где главным отличием от всей предыстории человечества является цифровой мониторинг и актуаторное human-free online управление. Киберсоциальный компьютеринг имеет смысл лишь в случае прямого и непосредственного взаимодействия каждого гражданина с любыми сервисами мониторинга и управления, устраняющими армию чиновников. Гражданин не должен вступать в отношения с чиновниками при получении сервисов, только кибер-роботы-автоматы обслуживают человека. Киберсоциальный компьютеринг своей нравственностью исключает средневековую дискриминацию граждан по расе, национальности, религии, истории, культуре, языкам, возрасту, полу и месту рождения. Издательство Springer анонсировало предложение авторам по написанию серии книг в направлении Computational Social Sciences. Такой интерес издательства подогревается цифровым рынком, который требует более адекватного поведения политической элиты государств по отношению к своим народам, а также коррекции политических решений, которая вполне возможна в рамках реализации Cyber Social Computing [32].

- Литература:** 1. *Abramovici M.* Digital System Testing and Testable Design / M. Abramovici, MA Breuer and AD Friedman. Comp. Sc. Press. 1998. 652 p. 2. *Vladimir Hahanov.* Cyber Physical Computing for IoT-driven Services. New York. Springer. 2018. 279p. 3. *Hahanov V.I.* Qubit technologies for analysis and diagnosis of digital devices / V.I. Hahanov, T. Bani Amer, S.V. Chumachenko, E.I. Litvinova // Electronic Modeling. Vol. 37, no. 3. 2015. P. 17-40. 4. *Хаханов В.И.* Кубитные структуры данных вычислительных устройств / В. И. Хаханов, В. Гариби, Е. И. Литвинова, А. С. Шкиль // Электронное моделирование. 2015. Т. 37, № 1. С. 76-99. 5. *Reinsalu U., Raik J., Ubar R. and Ellervee P.* Fast RTL Fault Simulation Using Decision Diagrams and Bitwise Set Operations // In 2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems. Vancouver, BC. 2011. P. 164-170. 6. *Rodrigo Fernandes de Mello, Moacir Antonelli Ponti.* Machine Learning. Practical Approach on the Statistical Learning Theory. Springer, 2018. 362 p. 7. *Forsyth David.* Applied Machine Learning. Springer, 2019. 478 p. 8. *Шумский С.А.* Машинный интеллект. Очерки по теории машинного обучения и искусственного интеллекта. Москва: РИОР, 2019. 340 с. 9. *Zhengbing Hu, Yevgeniy V. Bodyanskiy, Oleksii Tyshchenko.* Self-learning and Adaptive Algorithms for Business Applications: A Guide to Adaptive Neuro-fuzzy Systems for Fuzzy Clustering Under Uncertainty Conditions (Emerald Points) Paperback. June 25, 2019. 120 p. 10. *Zhu C., Leung VCM, Shu L. and Ngai ECH.* Green Internet of Things for Smart World // in IEEE Access, 2015. Vol. 3. P. 2151-2162. 11. *Zanella A., Bui N., Castellani A., Vangelista L. and Zorzi M.* Internet of Things for Smart Cities // in IEEE IoT Journal. Feb. 2014. Vol. 1, no. 1. P. 22-32. 12. *Frahim J.* Securing the Internet of Things: A Proposed Framework / J. Frahim // Cisco White Paper. 2015. 13. *Kharchenko V.* Green IT Engineering: Concepts, Models, Complex Systems Architectures / V. Kharchenko, Y. Kondratenko, J. Kacprzyk (Eds.) // In the book series "Studies in Systems, Decision and Control" (SSDC). Berlin, Heidelberg: Springer International Publishing, 2017. Vol. 1. 14. *Kharchenko V.* Green IT Engineering: Components, Networks and Systems Implementation / V. Kharchenko, Y. Kondratenko, J. Kacprzyk (Eds.) // In the book series "Studies in Systems, Decision and Control" (SSDC). Berlin, Heidelberg: Springer International Publishing, 2017. Vol. 2. 15. *Memory-Driven Computing.* [Online]. Available: <https://www.labs.hpe.com/next-next/mdc> 16. *Benenti G., Casati G., Strini G.* Principles of Quantum Computation and Information. Vol. 1: Basic Concepts. World Scientific, 2004. 256 p. 17. *Imai Hiroshi, Hayashi Masahito.* Quantum Computation and Information. From Theory to Experiment. Springer, 2006. 234 p. 18. *Hahanov V.* Cloud-driven Cyber Managing Resources / V. Hahanov, S. Chumachenko, E. Litvinova, O. Mishchenko, I. Yemelyanov, Bani Amer Tamer // Australian Journal of Scientific Reseach. № 1(5). 2014. С. 202-215. 19. *Hahanov I.* QuaSim – Cloud Service for Digital Circuits Simulation / I. Hahanov, W. Gharibi, I. Iemelianov, T. Bani Amer // Proc. of IEEE East-West Design & Test Symposium. 2016. Yerevan, Armenia. P. 363-370. 20. *Tarraf Danielle C.* Control of Cyber-Physical Systems. Workshop held at Johns Hopkins University, March 2013, Springer, 2013. 378 p. 21. *Mohammad A. Khan, Hillol Debnath, Cristian Borcea.* Balanced Content Replication in Peer-to-Peer Online Social Networks // 2016 IEEE International Conferences on Big Data and Cloud Computing, Social Computing and Networking, 2016. P. 274 – 283. 22. *Maria R. Lee, Tsung Teng Chen.* Understanding Social Computing Research. IT Professional. 2013. Vol. 15, Issue 6. P. 56 – 62. 23. *Jerry Higg, Varadraj Gurupur, Murat Tanik.* A Transformative Software Development Framework: Reflecting the paradigm shift in social computing // Proceedings of IEEE Southeastcon. 2011. P. 339 – 344. 24. *Jun Jason Zhang.* Cyber-Physical-Social Systems: The State of the Art and Perspectives / Jun Jason Zhang; Fei-Yue Wang; Xiao Wang; Gang Xiong; Fenghua Zhu; Yisheng Lv; Jiachen Hou; Shuangshuang Han; Yong Yuan; Qingchun Lu; Yishi Lee // IEEE Transactions on Computational Social Systems. 2018. Vol. 5, Issue 3. P. 134-144. 25. *Christopher M. Bishop.* Pattern Recognition and Machine Learning. Springer, 2006. 738 p. 26. *Susan Stepney.* Computational Matter / Susan Stepney, Steen Rasmussen, Martyn Amos // Springer, 2018. 336 p. 27. *Zhuge Hai.* Cyber-Physical-Social Intelligence. On Human-Machine-Nature Symbiosis. Springer. Singapore. 2020. 336 p. 28. *Elias G. Carayannis.* Handbook of Cyber-Development, Cyber-Democracy, and Cyber-Defense / Elias G. Carayannis. David F. J. Campbell. Marios P. Efthymiopoulos. 2018. 1089 p. 29. *Vladimir Hahanov.* Cyber-physical social monitoring and governance for the state structures / Vladimir Hahanov; Svetlana Chumachenko; Eugenia Litvinova; Anastasia Hahanova // IEEE 9th International Conference on Dependable Systems, Services and Technologies. 2018. P.123-127. 30. *Vladimir Hahanov.* Cyber Social Computing / Vladimir Hahanov; Svetlana Chumachenko; Eugenia Litvinova; Abdullayev Vugar Hacimahmud; Anastasia Hahanova; Tetiana Soklakova // 2018 IEEE East-West Design & Test Symposium (EWDTS). 2018. 31. <https://globenewswire.com/news-release/2018/12/03/1661174/0/en/Wave-Computing-Appoints-Industry-Veteran-Art-Swift-As-President-of-its-Recently-Acquired-MIPS-Licensing-Business.html> 32. <https://www.springer.com/generic/linked-items?SGWID=0-1741513-93-0-0>
- Транслитерированный список литературы:**
1. *Abramovici M.* Digital System Testing and Testable Design / M. Abramovici, MA Breuer and AD Friedman. Comp. Sc. Press. 1998. 652 p.
 2. *Vladimir Hahanov.* Cyber Physical Computing for IoT-driven Services. New York. Springer. 2018. 279p.
 3. *Hahanov V.I.* Qubit technologies for analysis and diagnosis of digital devices / V.I. Hahanov, T. Bani Amer, S.V. Chumachenko, E.I. Litvinova // Electronic Modeling. Vol. 37, no. 3. 2015. P. 17-40.
 4. *Hahanov V.I.* Кубитные структуры данных вычислительных устройств / В. И. Хаханов, В. Гариби, Е. И. Литвинова, А. С. Шкиль // Электронное моделирование. 2015. Т. 37, № 1. С. 76-99.

5. *Reinsalu U., Raik J., Ubar R. and Ellervee P.* Fast RTL Fault Simulation Using Decision Diagrams and Bitwise Set Operations // In 2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems. Vancouver, BC. 2011. P. 164-170.
6. *Rodrigo Fernandes de Mello, Moacir Antonelli Ponti.* Machine Learning. Practical Approach on the Statistical Learning Theory. Springer, 2018. 362 p.
7. *Forsyth David.* Applied Machine Learning. Springer, 2019. 478 p.
8. *Shumskij S.A.* Mashinnyj intellekt. Ocherki po teorii mashinnogo obuchenija i iskusstvennogo intellekta. Moskva: RIOR, 2019. 340 s.
9. *Zhengbing Hu, Yevgeniy V. Bodyanskiy, Oleksii Tyshchenko.* Self-learning and Adaptive Algorithms for Business Applications: A Guide to Adaptive Neuro-fuzzy Systems for Fuzzy Clustering Under Uncertainty Conditions (Emerald Points) Paperback. June 25, 2019. 120 p.
10. *Zhu C., Leung VCM, Shu L. and Ngai ECH.* Green Internet of Things for Smart World // in IEEE Access, 2015. Vol. 3. P. 2151-2162.
11. *Zanella A., Bui N., Castellani A., Vangelista L. and Zorzi M.* Internet of Things for Smart Cities // in IEEE IoT Journal. Feb. 2014. Vol. 1, no. 1. P. 22-32.
12. *Frahim J.* Securing the Internet of Things: A Proposed Framework / J. Frahim // Cisco White Paper. 2015.
13. *Kharchenko V.* Green IT Engineering: Concepts, Models, Complex Systems Architectures / V. Kharchenko, Y. Kondratenko, J. Kacprzyk (Eds.) // In the book series "Studies in Systems, Decision and Control" (SSDC). Berlin, Heidelberg: Springer International Publishing, 2017. Vol. 1.
14. *Kharchenko V.* Green IT Engineering: Components, Networks and Systems Implementation / V. Kharchenko, Y. Kondratenko, J. Kacprzyk (Eds.) // In the book series "Studies in Systems, Decision and Control" (SSDC). Berlin, Heidelberg: Springer International Publishing, 2017. Vol. 2.
15. *Memory-Driven Computing.* [Online]. Available: <https://www.labs.hpe.com/next-next/mdc>
16. *Benenti G., Casati G., Strini G.* Principles of Quantum Computation and Information. Vol. 1: Basic Concepts. World Scientific, 2004. 256 p.
17. *Imai Hiroshi, Hayashi Masahito.* Quantum Computation and Information. From Theory to Experiment. Springer, 2006. 234 p.
18. *Hahanov V.* Cloud-driven Cyber Managing Resources / V. Hahanov, S. Chumachenko, E. Litvinova, O. Mishchenko, I. Yemelyanov, Bani Amer Tamer // Australian Journal of Scientific Reseach. № 1(5). 2014. С. 202-215.
19. *Hahanov I.* QuaSim – Cloud Service for Digital Circuits Simulation / I. Hahanov, W. Gharibi, I. Iemelianov, T. Bani Amer // Proceedings of IEEE East-West Design & Test Symposium. 2016. Yerevan, Armenia. P. 363-370.
20. *Tarraf Danielle C.* Control of Cyber-Physical Systems. Workshop held at Johns Hopkins University, March 2013, Springer, 2013. 378p.
21. *Mohammad A. Khan, Hillol Debnath, Cristian Borcea.* Balanced Content Replication in Peer-to-Peer Online Social Networks // 2016 IEEE International Conference on Big Data and Cloud Computing, Social Computing and Networking, 2016. P. 274 – 283.
22. *Maria R. Lee, Tsung Teng Chen.* Understanding Social Computing Research. IT Professional. 2013. Vol. 15, Issue 6. P. 56 – 62.
23. *Jerry Higg, Varadraj Gurupur, Murat Tanik.* A Transformative Software Development Framework: Reflecting the paradigm shift in social computing // Proceedings of IEEE Southeastcon. 2011. P. 339 – 344.
24. *Jun Jason Zhang.* Cyber-Physical-Social Systems: The State of the Art and Perspectives / Jun Jason Zhang; Fei-Yue Wang; Xiao Wang; Gang Xiong; Fenghua Zhu; Yisheng Lv; Jiachen Hou; Shuangshuang Han; Yong Yuan; Qingchun Lu; Yishi Lee // IEEE Transactions on Computational Social Systems. 2018. Vol. 5, Issue 3. P. 134-144.
25. *Christopher M. Bishop.* Pattern Recognition and Machine Learning. Springer, 2006. 738 p.
26. *Susan Stepney.* Computational Matter / Susan Stepney, Steen Rasmussen, Martyn Amos // Springer, 2018. 336 p.
27. *Zhuge Hai.* Cyber-Physical-Social Intelligence. On Human-Machine-Nature Symbiosis. Springer. Singapore. 2020. 336 p.
28. *Elias G. Carayannis.* Handbook of Cyber-Development, Cyber-Democracy, and Cyber-Defense / Elias G. Carayannis. David F. J. Campbell. Marios P. Efthymiopoulos. 2018. 1089 p.
29. *Vladimir Hahanov.* Cyber-physical social monitoring and governance for the state structures / Vladimir Hahanov; Svetlana Chumachenko; Eugenia Litvinova; Anastasia Hahanova // IEEE 9th International Conference on Dependable Systems, Services and Technologies. 2018. P.123-127.
30. *Vladimir Hahanov.* Cyber Social Computing / Vladimir Hahanov; Svetlana Chumachenko; Eugenia Litvinova; Abdullayev Vugar Hacimahmud; Anastasia Hahanova; Tetiana Soklakova // 2018 IEEE East-West Design & Test Symposium (EWDTS). 2018.
31. <https://globenewswire.com/news-release/2018/12/03/1661174/0/en/Wave-Computing-Appoints-Industry-Veteran-Art-Swift-As-President-of-its-Recently-Acquired-MIPS-Licensing-Business.html>
32. <https://www.springer.com/generic/linked+items?SGWID=0-1741513-93-0-0>

Поступила в редколлегию 03.06.2020

Рецензент: д-р техн. наук, проф. Дрозд О.В.

Хаханова Анна Владимировна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: сжатие и восстановление двоичной информации, киберфизический и киберсоциальный компьютеринг. Адрес: Украина, 61166, Харьков, пр. Науки, 14. e-mail: anna.hahanova@nure.ua

Khakhanova Anna Vladimirovna, Ph.D., Associate Professor, Design Automation Department, NURE. Scientific Interests: compressed and binary information recovery. Address: Ukraine, 61166, Kharkov, Nauky Ave., 14, e-mail: anna.hahanova@nure.ua