

КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ТЕХНІЧНА ДІАГНОСТИКА

УДК 681.326:519.613

ЛОГІЧНІ СХЕМИ ТА АРХІТЕКТУРИ
КІБЕРСОЦІАЛЬНОГО КОМП'ЮТИНГУ
ХАХАНОВА Г.В.

Пропонується розвиток технології цифровізації відносин, пов'язане з розробкою кубітного логічного секвенсора для моделювання і розпізнавання social-явищ у великих потоках даних на основі інтерпретативних кубітних матричних моделей, методів і архітектур киберсоціального комп'ютингу, спрямованого на автоматичний синтез і аналіз соціально логічних схем, орієнтованих на моніторинг і управління соціальними процесами і явищами. Удосконалюється архітектура memory-driven киберфізического комп'ютингу, яка відрізняється паралелізмом процедур синтезу і аналізу логічних схем, призначених для моделювання social-процесів і явищ з метою моніторингу і управління. Модернізуються кубітно-матричні моделі логічних схем, які відрізняються унітарним кодуванням багатозначних змінних для синтезу секвенсорів для паралельного аналізу киберсоціальних процесів. Пропонуються кубітно-векторні, матричні методи синтезу і аналізу логічних схем, які характеризуються унітарним кодуванням значень багатозначних змінних для паралельного моделювання social-процесів і явищ. Розробляються кубітно-матричні моделі, структури даних, методи синтезу і аналізу логічних схем social-процесів, які дозволяють моделювати реакцію соціальної системи на прийняття конструктивних і деструктивних рішень людиною, керівником, чиновником, завдяки кубітному опису еталонів поведінки, що дає можливість актуаторно управляти громадянами та інфраструктурою. Ключові слова – киберсоціальний комп'ютинг, соціально-логічні схеми, кубітні структури даних, методи синтезу і аналізу логічних схем, розпізнавання соціальних явищ і процесів.

1. Вступ

Основою класичного комп'ютингу традиційно є таблиця істинності [1-4], яка в явному вигляді задає функціональний опис поведінки технологічного об'єкта, робота, людини, соціальної групи. Інші форми, такі як: аналітичні рівняння, альтернативні графи, кубітно-векторні структури [3-5] орієнтовані на мінімізацію пам'яті для зберігання даних при описі складних функціонально-стей. Історія питання полягає в вичерпній формалізації обчислень, які пов'язані з математико-демократичними процесами, що мають відношення до техніки і технологій [1]. Щодо формального опису діяльності людини або соціальних груп, тут існує практично біла пляма, яке частково покривається машинним навчанням [6-9]. Останнє є не що інше, як еволюційний, а значить повільне, побудова таблиці істинності для будь-якого, ще не вивченого процесу, як правило пов'язаного з біо-соціальними об'єктами.

При синтезі таблиці істинності існує два фази [6, 7]. Перша називається навчанням (Analysis), коли кожному вхідному (соціальному) впливу (Input Data Flow) ставиться у відповідність значення виходу (Truth Table) на підставі введеного критерію кластеризації (подібності-відмінності). Друга – тестування, коли виконується порівняння (Comparison) навченої або синтезованою таблиці істинності з фрагментом вже відомою. Результатом тестування є оцінка якості отриманої моделі (рис. 1), як кількість збігів G , віднесене до загальної кількості тестових впливів (Test Table): $Q = G / T$.

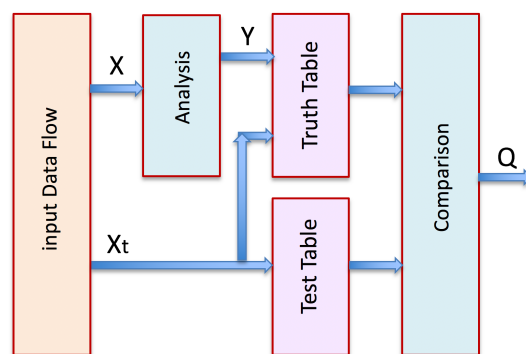


Рис. 1. Machine Learning for Truth Table Synthesis

Таким чином, для організації киберсоціального комп'ютингу необхідно насамперед навчитися аво-автоматично будувати таблиці істинності, які являють собою дискретне опис соціальних процесів і явищ, призначене для моделювання і передбачення наслідків від прийняття рішень зціальними групами, керівниками та співробітниками компаній і університетів [10-14]. Мета дослідження – суттєве підвищення якості моніторингу та управління громадських поцесів за рахунок імплементації у практику моделей, методів, алгоритмів і архітектур киберсоціального комп'ютин-га, які дають можливість моделювати реакцію суспільства на актуаторные впливу, а також розпоряд-знавать і передбачати соціальні явища. Сутність дослідження – розробка моделей, включаючи кубітні структури [15-17], методів, алгоритмів і архітектур киберсоціального комп'ютингу для моделювання реакції суспільства на управлінські впливи, розпізнавання і передбачення соціальних явищ, що дає можливість суттєво підвищити якість моніторингу та управління соціальними поцессами, а також життя людей за збереження екології навколишнього середовища. Об'єкт дослідження – технології синтезу, аналізу, цифрових логічних структур для моделювання і розпізнавання киберсоціальних процесів і явищ на основі створення кубітних структур даних і спеціалізованих обчислювальних архітектур.

Предмет дослідження – моделі, методи і алгоритми синтезу та аналізу, цифрових логічних структур для моделювання і розпізнавання киберсоціальних процесів і явищ на основі кубітних структур даних і спеціалізованих обчислювальних архітектур.

Завдання дослідження: 1) формальний синтез таблиці істинності для киберсоціальної функціональності; 2) синтез і аналіз соціальних логічних схем для моделювання суспільних і виробничих процесів; 3) типові логічні структури функціональностей і комбінаційні кубітні схеми для моделювання і розпізнавання соціальних явлень; 4) структурна модель хог-відношень матричного киберсоціального комп'ютингу; 5) архітектури, методи і алгоритми комп'ютингу для розпізнавання киберсоціальних явищ.

2. Метод синтезу соціально-логічних схем за таблицями істинності

Форми опису киберфізичних функціональностей для подальшого аналізу представлені в тому числі і роботами [18-30]. Таблиці істинності для завдання social-функціональностей формуються на основі позиційного або унітарного кодування значень змінних. При цьому передбачається замкнутість значень змінних у межах social-функціональності, які формують групу логічних функцій, заданих кубітними покриттями таблиць істинності. Конкретна функціональність може оперувати не більш, ніж n значеннями, змінної, які складають алфавіт або універсум примітивів-значень $A = \{A_1, A_2, \dots, A_i, \dots, A_n\}$. Наступна таблиця ілюструє два види кодування універсуму примітивів-значень змінної для формування таблиць істинності трьох логічних функцій, які беруть участь у створенні social-функціональності:

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

Аналогічні таблиці істинності створюються для всіх змінних, що в сукупності формують матрицю таблиць істинності для кожного social-процесу або явища. За фактом, стовпець позиційного кодування P-code не використовується на практиці, але він слугує базовим компонентом для доказу застосовності класических таблиць істинності при описі будь-яких social-процесів. Стовпці P1, P2, P3 тривіально викорис-ся для мінімізації кількості стовпців.

Далі пропонується матрична модель social-процесів на метриці значень змінних. Стовпці-кубіти P1, P2, P3 з попередньої таблиці

трансформуються в рядки-вектори, які представляють собою супер-позицію унітарних кодів значень параметрів, що беруть участь у формуванні social-процесів: PT1 - PT3:

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

Дуалізм інтерпретації даної таблиці формує ієрархію, яку необхідно враховувати при синтезі моделі social-процесів. 1) Отримана таблиця або матриця об'єднання унітарних кодів, розміщених в координатах, являє собою двійковий модель social-функціональності, прив'язану до модельного часу PT1–PT3. Тут істотно, що кожна координата матриці являє собою векторну або кубіт-ву форму опису таблиці істинності. 2) Однак, формат матричної моделі також адекватно створює структури даних для опису сукупного social-процесу, що має місце бути в компанії. При цьому кожен стовпець матриці PTi інтегрально задає social-функціональність.

Модель, алгебра, структура, граф, таблиця, матриця, система, рівняння є еквівалентними матема-демократичними поняттями, в основу яких покладена структура взаємопов'язаних компонентів. При цьому завжди розглядається замкнутий алфавіт або безліч примітивних компонентів, які створюють основу структури або універсум примітивів. Всі можливі зв'язки між елементами алфавіту або універсуму формують сигнатуру або базові операції алгебри. Булева алгебра на найнижчому рівні представлена алфавітом або універсумом примітивних символів $\{0,1\}$, які фігурують у значеннях булевих змінних і функції $Y=f(X)$, де $\{X,Y\}=\{0,1\}$. При цьому поведінка функції визначається таблицею істинності, де впорядкованої двійкової послідовності вхідних змінних ставиться у відповідність двійкове значення функції. Менш поширеною є інтерпретація таблиці істинності, де кожному двійково-му коду або адресою ставиться у відповідність одиничне або нульове значення функції. При цьому сукупність кодів або адрес являє собою універсум примітивних компонентів або підставу алгебри, на якій визначаються базові операції. Таким чином, вводиться алгебра логіки, де багатозначні стану вхідної змінної (алфавіт) кодуються в таблиці істинності двійковими векторами, які представляють собою адреси комірок пам'яті, де зберігаються $\{1,0\}$ -значення функції. Інтегрально 1-значення функції в таблиці істинності формують підмножина існуючих

примітивів $A = \{a, c, e, f\}$ на заданому універсумі $A = \{a, b, c, d, e, f, g, h\}$, яке суперпозиційно створює функціональність:

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

Кубітним покриттям даної функціональності є F-вектор двійкових станів вихідний змінної, розмірність якого дорівнює універсуму примітивних компонентів, що формують функцію, а число одиничних значень одно підмножини примітивів з універсуму, який бере участь у формуванні за-даної функціональності. Слід зазначити, що функціональність формується значеннями суттєвої перемінної у тимчасових кадрах social-процесу або social-патерну. Враховуючи, що кількість суттєво-них змінних, як правило більше 1, то необхідно синтезувати цифрові логічні схеми з кубіт-них покриттів функцій, число яких дорівнює кількості істотних змінних. Таким чином, кінцеве безліч істотних змінних є базовими елементами для синтезу цифрових логічних схем social-функціональності або social-процесу. Далі представлена структура, яка оперує кубітними покриттями примітивів, об'єднаними логікою елементів: and, or (рис. 2).

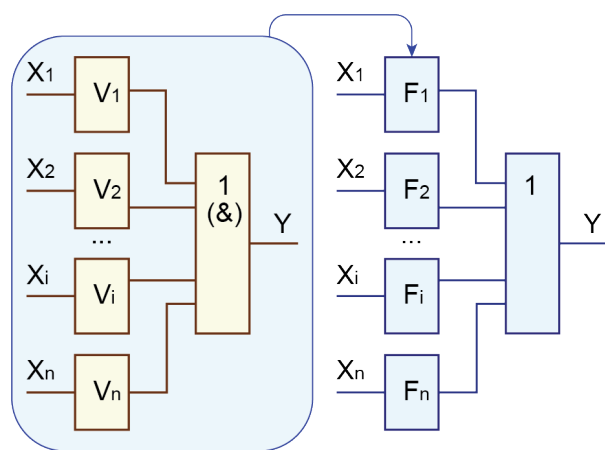


Рис. 2. Logic social-functions

Логічні схеми, синтезовані з кубітних форм значень змінних, призначені для моделювання social-процесів з метою визначення поведінки киберфизической social-архітектури комп'ютингу на заданих вхідних робочих віз-дії. Робочими впливами є суперпозиції унітарних кодів вхідних значень змінних. Стан виходу логічної social-схеми, рівне одиниці, свідчить про позитивному результаті взаємодії вхідних значень істотних

змінних на хід виконання social-патерну або процесу для досягнення поставленої мети. Таким чином, замість ланцюжка даних, що ілюструє послідовність дій social-функціональності, пропонується принципово нова форма – комбінаційна цифрова логічна схема, паралельно інтегруючи лише істотні властивості social-функціональності, де основною відмінністю є можливість моделювання social-процесів.

Схемна структура формує функціональне поведінка або модель social-функціональності на основі використання кубітних покриттів змінних, інваріантну до часу. Сукупність social-функціональностей створює основу для синтезу паралельної цифрової моделі social-процесу, яка являє собою спеціалізований обчислювач який реалізує кібер-фізичний комп'ютинг для моніторингу, моделювання та управління social-процесами компанії. Ієрархія social-комп'ютингу представлена компонентами:

<значення – змінна – функціональність – процес> або <value – variable – function – process>.

Логічна структура або social-процесор інваріантний до часу, створює social-функціональності на основі кубітних форм таблиць істинності логічних елементів, прив'язаних до універсуму примітивних значень змінних. Переваги social-процесора полягають у компактності подання та високій швидкодії логічної схеми social-функціональності, яка визначається кубітними векторами змінних, бітова розмірність яких дорівнює кількості примітивних даних значень кожної змінної. Кубітні структури даних інваріантні до їх hardware реалізації, у вигляді логічної схеми, або software, у вигляді таблиць-матриць опису функціональностей [18-20]. Кубітна двійкова форма опису social-функціональностей дає можливість технологічно моделювати як завгодно складні social-процеси шляхом впливу на схему двійковими вхідними наборами, які відповідають унітарно-кодованим даними, в цілях визначення вихідних станів social-процесора, що виконують роль клас-сификатора та/або актуатора. Позитивним є той факт, що кожен кубітний логічний елемент, заданий у векторному форматі кількості значень змінної, створює образ social-функціональності єдиничними значеннями своїх координат. При цьому число кубітних векторів-елементів, об'єднаних в схему and(or)-елементом, дорівнює кількості змінних синтезованої social-функціональності. Крім того, візуалізація досить компактній логічної схеми завдання social-функціональності дає можливість поль-зователю або керівнику побачити сутність social-функціональностей без

прив'язки до часу, де важливим представляється покрити всі необхідні змінні конструктивними діями користувача для отримання відповідних сервісів.

Недоліком процесорної social-моделі є виключення параметра часу у функціонуванні логічної моделі, що призводить до її неадекватності у разі потреби опису последовательностной граф-схеми алгоритму social-функціональності. Крім того, для моделювання даних з допомогою логиче-ської social-схеми необхідно синтезувати препроцесор текстових даних в унітарні кубитные вектори на основі використання hash-функцій, що суттєво зменшують час порівняння вихідного і вхідного текстового рядку з бібліотечними даними. Необхідний також постпроцесор для інтерпретації станів виходів social-процесора користувачу (автомата), який буде генерувати наступні актюаторные дії. В сукупності, препроцесор і постпроцесор займають близько 10 відсотків пам'яті і визнач-кового часу обробки даних від базового варіанту, пов'язаних з синтезом і аналізом кубитных структур, що реалізують архітектуру social-схеми підприємства.

Формальний синтез логічних схем social-процесу компанії здійснюється шляхом виконання наступних пунктів:

- 1) Формування універсуму примітивних даних по кожному атрибуту social-процесу.
- 2) Заповнення вектор-кубіта одиничними значеннями тих координат, номери яких відповідають даним, використовуваним при формуванні конкретної social-функціональності.
- 3) Об'єднання за and-функції всіх логічних елементів, заданих кубітами задіяних значень змінних, для отримання схеми social-функціональності.
- 4) Об'єднання всіх схем social-функціональностей для отримання логічного social-процесора компанії.

Завдання аналізу, які вирішуються з допомогою логічних social-схем:

- 1) Моделювання вхідного потоку даних в цілях їх класифікації на множині бібліотечних social-функціональностей шляхом використання вектора моделювання (покриття), який дає можливість на кожному кроці визначати стан модельованого social-процесу, а також генерувати керуючі віз-дії, спрямовані на отримання покриття social-функціональності, виходячи з нульових координат вектора моделювання.
- 2) Автоматичне генерування вхідних двійкових векторів на основі аналізу даних для їх підутнього моделювання на social-процесорі.

Для автоматичного створення social-процесора на основі використання вхідного контенту

необхідно: 1) Визначення ключових слів для формування змінних social-процесу і подальшого обчислення універсумов примітивних значень даних. 2) Формування ідеї кожного social-процесу, що створює закінчену social-функціональність у вигляді матриці кубитных покриттів, складених з унітарних кодів, відповідних значень з універсумов примітивів змінних. Інструментом для точного моделювання вхідних впливів на кубітних покриттях social-функціональності є функція приналежності, яка обчислюється шляхом визначення кодового відстані між вхідним двійковим набором, що подається на логічний елемент і кубітним вектором останнього:

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	

Однак приналежність частини social-функціональності визначається операцією логічного множення або перетину, яка повинна бути дорівнює вхідному вектору $I \wedge F = I$, що означає покриття логічним еле-ментом social-функціональності значення вхідного вектора $I \in F$.

Отже, структури даних social-аналітики або архітектура social-процесора, зображеного на рис. 3, містить: 1) метрику у вигляді множини $U = \{U_1, U_2, \dots, U_i, \dots, U_n\}$ кубітних покриттів універсумів примітивних значень, яке має потужність, рівну кількості змінних n . Кожен універсум $U_i = \{U_{i1}, U_{i2}, \dots, U_{ij}, \dots, U_{im}\}$ формує повне безліч примітивних значень змінної; 2) на основі даної метрики, шляхом синтаксичного порівняння, синтезуються еталонні social-функціональності $P = \{P_1, P_2, \dots, P_r, \dots, P_k\}$, де стовпчик $P_r = \{P_{r1}, P_{r2}, \dots, P_{ri}, \dots, P_{rn}\}$ формує social-функціональність, а координата стовпця визначаються-ється двійковим вектором, $P_{ri} = \{P_{ri1}, P_{ri2}, \dots, P_{rij}, \dots, P_{rim}\}$ записаним у форматі кубітного покриття $U_i = \{U_{i1}, U_{i2}, \dots, U_{ij}, \dots, U_{im}\}$ універсуму примітивних значень змінної. При цьому, $P_{rij} = 1$, якщо в social-функціональності присутній символічне значення змінної, рівне U_{ij} . Кубітне покриття social-функціональності $P_r = \{P_{r1}, P_{r2}, \dots, P_{ri}, \dots, P_{rn}\}$ завжди є структурною частиною кубітного покриття універсума примітивів: $P_r \in U$

оскільки завжди працює аксіома $Pr \wedge U = P$; 3) метою створення еталонних social-функціональностей є автоматичне генерування актюаторних впливів $A = (A_1, A_2, \dots, A_p, \dots, A_q)$, які замикають цикл social-аналітики керуючими впливами, перетворюючи її в RPA-структуру.

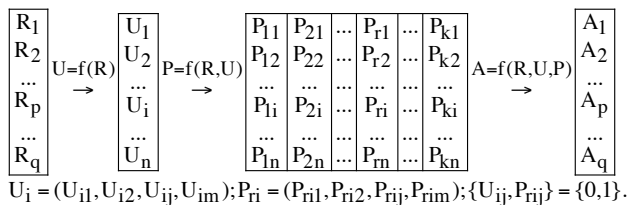


Рис. 3. Матрична архітектура social-комп'ютинг

Матрична архітектура social-комп'ютингу підтримує ієрархію кубітних покриттів $R_p \in Pr \in U$, де R_p -вектор входних значень змінної не може бути більше P_{ri} -кубіта social-змінної, який не може бути більше універсуму U_i значень змінної. Архітектура програмного серверного додатка для синтезу логічних social-функціональностей і по-наступного моделювання на них контенту великих social-даних представлена на рис. 4. Структура містить дві частини, де ліва з них призначена для автоматичного синтезу структур дан-них, логічних елементів і схеми в цілому. Права частина орієнтована на ручне введення схемних елементів і структури social-функціональності на основі використання графічного інтерфейсу. Обидві частини архітектури навантажені на блок кубітного моделювання входного контенту з метою визначення в ньому аналогів social-процесів, конструктивних і деструктивних, синтезованих раніше за інших можливостей. Результати моді-вання зберігаються в бібліотеці, яка містить також і логічні еталони social-функціональностей.

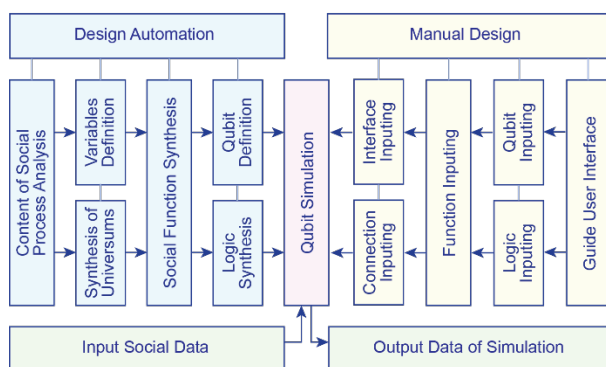


Рис. 4. Архітектура social-процесора

3. Логічні структури social-функціональностей

Пропонується рішення в частині синтезу логічних схем для аналізу social-процесів, пов'язане з вико-

ристанням кубітно-реєстрових змінних, які унітарно кодують безліч примітивних значень з багатозначного універсуму змінної. У цьому випадку будь-яка функціональність може бути представлена логічним елементом and (or, or), який має реєстрові входні змінні, представлені кубітними векторами [2-4]. Середнє арифметичне значення на виході логічного елемента з багатозначними або векторними входами визначається збігом сигналів з кубітними векторами за всіма входними змінними. Для цього необхідно виконувати процедуру, пов'язану з логічним перетином (and-операція) входного сигналу з кубітним вектором і подальшим хог-порівнянням результату перетину з вихідним сигналом. Якщо таке порівняння дорівнює нулю по всіх координатах векторів, то формується середнє арифметичне значення виходу для однієї багатозначної змінної. При наявності одиниць на всіх виходах змінних логічного елемента and, його значення буде дорівнює одиниці. Аналітична модель і схемна реалізація процедури порівняння двох векторів для визначення приналежності одного з них Q_1 до іншого Q_2 має вигляд, представлений на рис. 5.

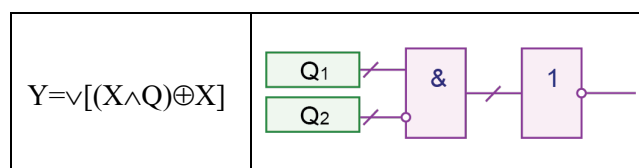


Рис. 5. Логіка порівняння даних

Схемна реалізація пристрою для моделювання соціальних процесів X шляхом порівняння з еталонні-ми функціональностями Q представлена на рис. 6.

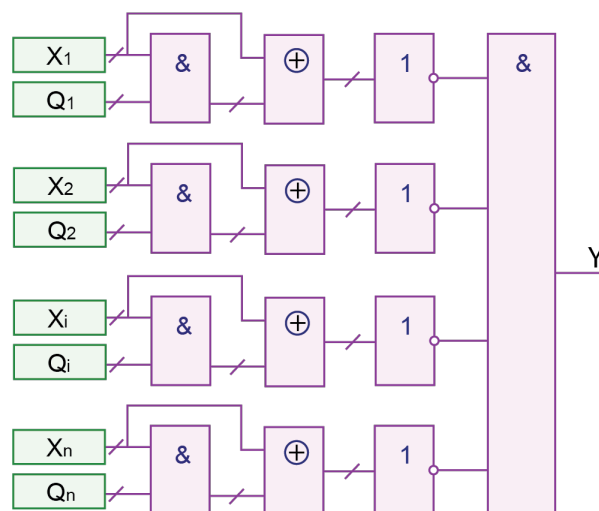


Рис. 6. Схema моніторингу і аналізу соціальних процесів

Тут виконується порівняння двійковий-кодованих вхідних значень X з відповідними кубітними векторами еталонних значень змінних соціальних процесів. Стан вихідної змінної одно $Y = 1$, якщо вхідні значення рівні кубітним векторам по всьому кординат. Це означає, що вхідні дані в сукупності представляють собою наперед задану еталонну соціальну функціональність. Кубітні вектори в сукупності формують матрицю еталонних значень соціальної функціональності. Швидкодія отримання рішення на виході схеми дорівнює п'яти структурним тактам. Слід зазначити, що в загальному випадку стан виходу схеми формується функцією приналежності $Y = m(X, Q) = 1 - d(X, Q) / n$, яка визначена в інтервалі $(0,1)$ числом (розбіжностей) єдиничних координат на регістровому виході логічної and-функції кожної змінної, де n - число координат Q -вектора, $d(X, Q)$ - кодова відстань за Хеммінгом. Схема формує функцію приналежності елемента або підмножини до наперед заданого еталонного вектору примітивів за один автоматний цикл, завдяки векторній формі завдання підмножин однаковою розмірності. Інакше, схема дає можливість відповісти на питання: чи належить вхідні двоичная послідовність X еталонної соціальної функціональності Q , також представленій в цифровому вигляді $Y = 1$.

Можна використовувати схему з нульовим позитивізмом вихідного сигналу, коли максимальна належність вхідного сигналу X до кубітному еталону Q ідентифікується 0-рівнем відмінності між двома векто-рами $Y = a(X, Q) = d(X, Q) / n$.

Це істотно спрощує логічну структуру. Тут $Y = 0$ означає повне співпадання між вхідними даними і кубітним вектором еталонної функціональності, що дає підстави ідентифікувати або класифікувати аналізований процес, як належить еталонного вектора. В результаті отримує-ся, що функція збігу (match function) є інверсною по відношенню до функції приналежності (membership function): $Y = a(X, Q) = d(X, Q) / n = \text{not}[m(X, Q) = 1 - d(X, Q) / n]$.

Сигналами в змінних можуть виступати: символи, літери, слова, речення, цифри, числа, стосунки, структури, малюнки, фотографії, відеофільми, звукові фрагменти, процеси і явища.

Наступна схема, рис. 7, ще більше спрощує логічне моделювання вхідних social-потоків даних, де видалені хог-елементи, але додані U -елементи, що виконують роль таблиці універсуму примітивних значень кожної змінної, щодо яких синтезуються кубітні вектори змінних для формування social-патернів.

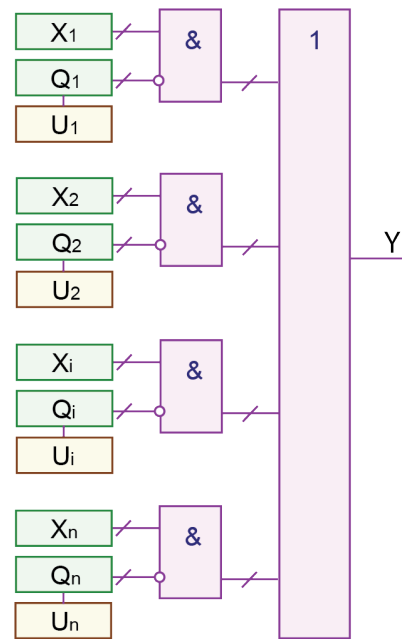


Рис. 7. Спрощена схема social-simulation

Щоб показати схему для формування структур даних social-функціональності (U, Q, X) , необхідно з'єднати вхідний потік даних S для кожної social-змінної, рис. 8.

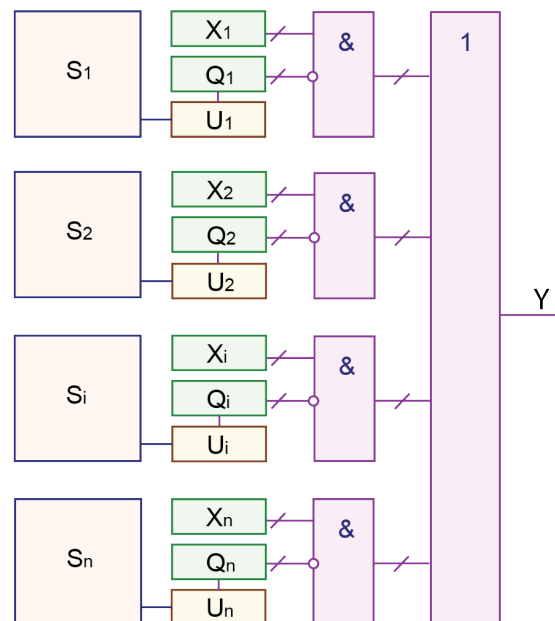


Рис. 8. Схема формування social-патерну

Схема для формування структур даних social-процесу (V, P, Y) з'єднує вхідні потоки даних для змінних F щодо кожної social-функціональності. Приблизна UU -модель social-процесу для аналізу діяльності компанії представлена на рис. 9.

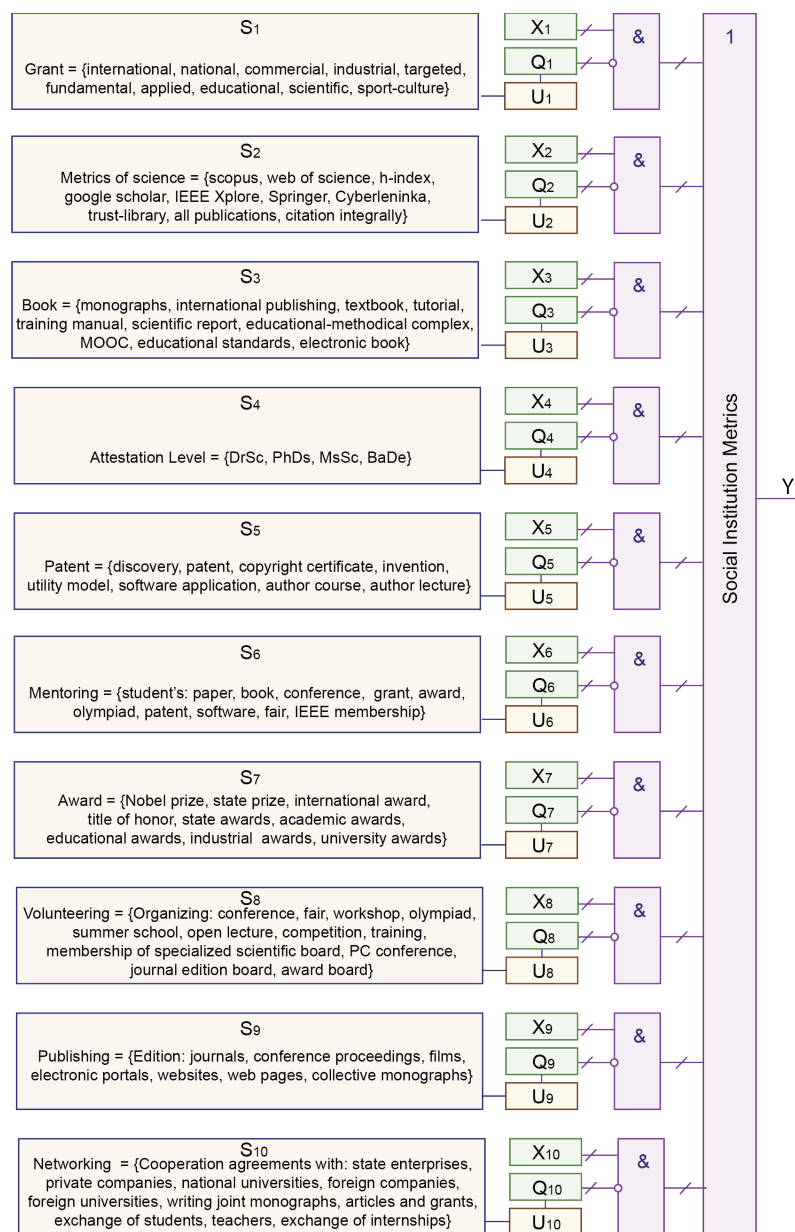


Рис. 9. Метрика-схема social-процесу

Універсум універсумов (UU) являє собою дворівневу модель процесу чи явища, призначена ченню для активного цифрового моделювання в цілях точного розпізнавання великих потоків вхідних даних шляхом метричного порівняння з заданими еталонами.

Кожен з параметрів представлено значеннями, які в сукупності складають універсум другого рівня. У результаті виходить UU-метрика функціональності еталонного процесу або явища, стосовно якого можна вимірювати-моделювати потік вхідних даних, які претендують на цю роль. Для проектування киберсоціального комп'ютингу використовуються два види моделей логічних процесорів.

Кубитно-регістрова модель (KPM) оперує двома входами логічного and-елемента, на які подаються: 1) кубітний вектор-еталон багатозначної

змінної social-функціональності; 2) вхідний вектор даних social-процесу, що підлягає розпізнаванню. Обидва вектора паралельно порівнюються між собою за процедурою $Y = \vee[(X \wedge Q) \oplus X]$, що включає три паралельних операції, що призводить до двійкового результату: $Y=0$, якщо X належить вектору Q , в іншому випадку $Y=1$.

Кубитно-логічна модель (КЛМ) оперує одним входом логічного and-елемента, на який подається значення-адреса даних social-процесу, відповідне даної змінної. За адресою, одержаного в результаті кодування текстового фрагмента, визначається координата кубітного вектор-еталона багатозначної змінної social-функціональності, значення якої формує стан виходу Y для розпізнавання значення вхідних даних відносно заданого кубітного вектора,

формує логічний еталон. Іншими словами, одна ітерація моделювання на кубитному елементі КЛМ-моделі social-функціональності визначає належність вхідного значення до кубитному вектору логічного еталону. КРМ модель змінної social-функціональності дає можливість паралельно, за один автоматний цикл, визначати належність вхідного вектора до кубитному вектору логічного еталону. В цьому їх відмінність. Природно, що вхідні впливи для обох моделей формуються з текстових фрагментів вхідних даних шляхом їх унітарного кодування на універсальній множині значень кожної змінної кібер-соціальної функціональності.

4. Комбінаційні кубітні схеми social-комп'ютингу

Одним з яскравих представників memory-driven social-комп'ютингу виступає компанія Wave Computing Inc, Silicon Valley, яка з'єднує AI-ML з dataflow-based MIPS RISC (конвеєр з обмеженою системою команд) architecture. Девіз компанії – "follow the data with AI" в цілях надання deep learning по будь-якому місцю знаходження великих даних – від дата центрів до терміналів хмари [31].

Мета – суттєве підвищення якості розпізнавання соціальних процесів шляхом створення логічного кубитного процесора паралельного моделювання і розпізнавання social-патернів у вхідних потоках великих даних, отриманих на основі метричного моніторингу інфраструктурних компонентів, для цифрового актуаторного управління соціальними процесами.

Сутність полягає в синтезі логічної схеми social-явища для моделювання (порівняння, розпізнавання) вхідного потоку даних в цілях формування на виході процесора актуаторних значень (сигналів), спрямованих на усунення відмінностей між вхідними даними і еталонам.

Відмінність інновації від існуючих рішень моніторингу та порівняльного пасивного відображення social-процесу, полягає в online генеруванні керуючих впливів, активуючих комп'ютинго-правові процедури щодо приведення social-процесу до ідеального алгоритмом або social-траєкторії.

Дослідження підтримує три фази social-аналітики (синтез, аналіз, управління):

- 1) синтез структур даних і логічних схем, що формують кубитные моделі social-процесу і social-патернів на основі аналізу вхідних потоків великих даних – фаза навчання;
- 2) аналіз або моделювання вхідних потоків великих даних на синтезованих логічних схемах social-патернів з метою визначення комплектуючості поточкових фрагментів

(розпізнавання) до заданих social-функціональностям;

- 3) управління інфраструктурними кіберфізическими компонентами за допомогою актуаторних сигналів, отриманих в результаті аналізу вхідних потоків даних у цілях зменшення часових витрат надання сервісів.

Досягнення мети дає можливість отримати наступні функціональності, практично орієнтовані на точне рішення завдань вичерпного моніторингу та цифрового управління social-процесами:

- 1) автоматичний синтез базисного кубіта змінних $P = (P_1, P_2, \dots, P_i, \dots, P_n)$ social-процесу шляхом пошуку або визначення універсуму непересічних класів еквівалентних змінних на основі семантичного і синтаксичного аналізу вхідних потоків великих даних;

- 2) автоматичний синтез базисних кубітів значень кожної змінної $Q_i = (Q_{i1}, Q_{i2}, \dots, Q_i, \dots, Q_{im})$ social-процесу шляхом пошуку або визначення універсуму непересічних класів еквівалентних значень для кожної змінної на основі семантичного і синтаксичного аналізу вхідних потоків великих даних;

- 3) автоматичний синтез кубитной Q-матриці універсумов $Q = [Q_i]$ social-процесу на основі об'єднання кубіт-універсумов значень кожної змінної під егідою кубіт-універсуму змінних. Q-матриця, як майстер-шаблон, породжує всі можливі social-патерни шляхом маскуваннн її значеннями координат $\{0,1\}$;

- 4) автоматичний синтез актуальною кубитной Q-матриці і логічної схеми social-патерну на основі маскуваннн кубіт-універсумов змінних і їх значень шляхом семантичного і синтаксичного аналізу вхідних потоків великих даних;

- 5) автоматичний синтез кубитного логічного процесора для моделювання і розпізнавання social-патернів у вхідних потоках великих даних на основі кубитных Q-матриць шляхом об'єднання актуальних логічних схем social-функціональностей;

- 6) автоматичний синтез функцій виходів логічних патерн-схем для актуаторного управління компонентами інфраструктури social-процесу на основі моделювання вхідних потоків великих даних, що використовують кубітні Q-матриці social-функціональностей;

- 7) автоматичний синтез GUI на основі Q-матриці дає можливість візуалізувати online процедури моделювання вхідних потоків даних на логічних схемах social-процесора в цілях визначення станів виходів, формують актуаторные впливи для інфраструктурних компонентів і працівників компанії.

Рішення перерахованих завдань пов'язане з створенням моделі, методу та інтерпретативного гнучкого процес-ра киберсоціального social-комп'ютингу, спрямованого на автоматичний синтез і аналіз кубитних логічних схем, орієнтованих на моніторинг, моделювання, розпізнавання та управління social-процесами і явищами:

- 1) процесор киберсоціального social-комп'ютингу на основі синтезу кубитних логічних схем для моніторингу, моделювання, розпізнавання та управління social-процесами;
 - 2) кубітно-матрична модель універсуму змінних і універсуму значень кожної багатозначною змінної для синтезу логічних схем social-патернів, орієнтованих на аналіз і управління social-процесами;
 - 3) кубітно-матричний метод синтезу логічної схеми для моделювання і розпізнавання social-функціональностей на основі унітарного кодування значень багатозначних змінних в цілях оптимального управління social-процесом;
 - 4) кубітно-матричний інтерпретативний метод моделювання вхідних потоків social-даних social-процесів на основі використання еталонних логічних елементів social-функціональностей з унітарною кодуванням багатозначних змінних. Метод відрізняється від розумних блокчейн-контрактів, що представляють собою виконуваний код, інтерпретативним аналізом гнучких структур даних, що дає можливість змінювати модель social-процесу в online режимі реального часу;
 - 5) тестування і верифікація кубітно-матричної архітектури social-комп'ютингу на прикладах аналізу і розпізнавання процесів у вхідних потоках великих даних, пов'язаних з кіберфізическим відображенням і управлінням діяльністю компанії.
- Кубітно-матрична інтерпретативная модель social-патернів і її унітарно-кодований екземпляр для моделювання і актюаторного управління social-процесом компанії представлені на рис. 10.

		U-Values					
U-parameters	P1	Q11	Q12	...	Q1j	...	Q1m
	P2	Q21	Q22	...	Q2j	...	Q2m
	...	...	...	...	...	...	...
	Pi	Qi1	Qi2	...	Qij	...	Qim
	...	...	...	...	...	...	...
	Pn	Qn1	Qn2	...	Qnj	...	Qnm

Q-matrix (UU-model of business process)

		U-Values					
U-parameters	1	1	0	0	0	0	1
	0	0	0	0	0	0	0
	1	1	0	0	1	1	1
	1	0	1	1	0	1	1
	0	0	0	0	0	0	0
	1	1	1	0	0	1	1

Q-matrix (UU-model of actual pattern)

Рис. 10. Кубітна матриця social-процесу

Фактично, Q-матриця являє собою формальну запис протоколу, анкети або ідеального блокчейн-контракту, який додатково виробляє актюаторные сигнали (заповнити активні поля, по-ставити підпис, сканувати паспорт, зробити фото, оцінити якість послуг), які спонукають людини виконувати певні дії для досягнення мети – ліквідації відмінностей між зразком і реальністю. Таким чином, прямі і безпосередні інтеракції між роботом-автоматом і користувачем створюють social-комп'ютинг, використовує Q-матрицю і логічні схеми, спрямований на отримання продуктів або сервісів за мінімальний час без участі посередників у формі чиновників, паперових наказів, положень і законів.

5. Кубітний метод моделювання

Вихідні дані:

- 1) S-data – вхідний потік великих даних для виконання social-аналітики;
- 2) U-матриця – універсум змінних для опису social-процесу з допомогою універсумов вербальних значень кожної змінної;
- 3) X-матриця – векторний формат вхідних даних за кожною змінною для одного віть-рації моделювання;
- 4) Q-матриця – сукупність кубитних двійкових векторів для опису еталонного патерну по всім перемінним;
- 5) U-вектор – універсум вербальних значень змінної для опису social-патерну;
- 6) X-вектор – впорядкована послідовність двійкових вхідних даних з однієї змінної для виконання ітерації моделювання;
- 7) Q-вектор – кубитний двійковий вектор для опису еталонного патерну з однією змінною;
- 8) Функція приналежності – чисельне значення кількості різних дво-кових однойменних координат в векторах X, Q, визначається за допомогою операції $(X \wedge \text{not} Q) = Y$;
- 9) актюаторные сигнали – одиничні значення координат вектора Y, які ініціюють процедури, спрямовані на усунення розбіжностей між X-вектором і еталонним Q-вектором.

Структура методу: пристрій містить п'ять шарів вертикальних компонентів, де перший з них являє собою вхідні потоки social-даних, що надходять від сенсорних пристроїв інфраструктури компанії на тріади елементів другого шару, де перший елемент тріади U являє собою універсум вербальних примітивів-значень кожної змінної, соединений с елементом Q, який є двійковим кубітним вектором, об'єднуючим значення змінної для синтезу social-патерну у форматі універсуму U, який також служить шаблоном для формування двійкових значень кубитного вхідного вектора даних X, призначеного для спільного паралельного моделювання з кубіт-

універсумом Q , яке заключається в їх порівнянні один з одним шляхом виконання реєстрової логічної операції and , в третьому шарі елементів, над вмістом кубіт-вектора X і інверсним станом кубіт-вектора Q , що дає можливість отримати на виході and -елемента 0 -вектор, якщо кубіт-вектор X належить кубіт-універсуму Q , і $\text{not-}0$ -вектор – в іншому випадку, що дає можливість використовувати отримані двійкові вектори від всіх логічних and -елементів відповідних змінних для їх логічного об'єднання на 0 -елементі четверто-го шару, який формує інтегральний сигнал $Y=0(1)$ для визначення приналежності (неприналежності) вхідних векторів даних X social-паттерну , що визначається кубіт-векторами Q всіх змінних, що, у свою чергу, дає можливість формувати з допомогою елемента-таблиці з п'ятого шару актуаторні сигнали-впливу, призначені для усунення суперечностей між координатами кубіт-векторів Q social-паттерну і вхідних векторів даних.

Метод: визначає послідовність виконання операцій і процедур для розпізнавання social-паттернів у вхідних потоках даних і формування актуаторних сигналів-впливів для усунення протиріч між ними, який починається з процедури синтаксичного аналізу вхідного потоку social-даних , який полягає у вилученні з нього (1) універсуму ключових слів або змінних-примітивів, де для кожної змінної витягується (2) універсум значень, що в сукупності становить (3) U -матрицю універсуму універсумов, яка служить платформою для подальшого (4) синтезу двох основних двійкових векторів: кубітного Q -вектора і кубітного X -вектора вхідного потоку даних кожної змінної social-паттерну , які далі використовуються для виконання (5) процедури моделювання, яка полягає в паралельному порівнянні координат кубітних векторів $(X \wedge \text{not}Q) = Y$, що дає можливість отримати (6) результуючий вектор $Y = (Y_1, Y_2, \dots, Y_i, \dots, Y_n)$, де кожна 1 -координата ідентифікує відмінність між со-ответствующими розрядами векторів X і Q , що дає можливість (7) визначати функцію приналежності вхідного X -вектора до Q -вектору паттерну для кожної змінної, що інтегрально, по всім змінним формує (8) функцію приналежності вхідного X -паттерну до еталонного Q -паттерну шляхом підрахунку в усіх результуючих Y -вектори 1 -координат, які дають можливість (9) формувати актуаторні сигнали, спрямовані на усунення суперечностей в координатах між актуальним вхідним вектором X і еталонним $\text{паттерном } Q$, що є кінцевою метою $\text{social-комп'ютингу}$, орієнтованого на точне моделювання соціальних процесів за рахунок

цифровізації, автоматизації та просторово-часової оптимізації social-процесу сервісного обслуговування.

Сигнали розгалуження в social-паттернах . Актуаторні сигнали формують розгалуження в класичній граф-схемою алгоритму управління social-процесом і створюють умовні процедури реалізації social-паттерну . В даному випадку існують дві структури даних для реалізації змінних у social-паттернах , що позначає умови (X):

1) кожній X -змінній ставиться у відповідність окремий рядок в матриці або кубіти універсуму змінних;

2) для всіх X -змінних ставиться у відповідність один рядок в матриці, де всі значення різних X -змінних будуть представлені унітарними кодами. Враховуючи той факт, що умовні вершини у граф-схемі алгоритму ініціюють певні альтернативні дії, то вихідна (логічно-ська) функція для всіх умовних X -змінних повинна бути визначена в міждисциплінарному алфавіті. Це означає, що $Y=\{0,1\}$, є лише приватним випадком при формуванні таблиці істинності X -змінною. Загальний випадок визначається безліччю значень, які виконують актуаторні впливи на суб'єкт або компонент social-процесу : $Y=\{\text{fill in this gap, please do the signature, try again the spelling, correct the wording, do scanning the passport, press the enter-button, scan the right eye}\}$. Таким чином, кожна логічна схема social-паттерну має багатозначні значення вхідних змінних і функції виходу, які формують функціональне ставлення, спрямоване на ініціювання або актуацію суб'єктів інфраструктури або користувачів, як реакцію на вхідний потік великих даних, отриманих шляхом моніторингу social-процесу для їх моделювання:

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

Призначення (багатозначною) функції виходу логічної схеми social-паттерну полягає в активації процедур social-процесу , спрямовані на досягнення мети у вигляді отримання продукту або сервісу в рамках інфраструктури кіберфізичного $\text{social-комп'ютингу}$.

По суті реакція виходу логічної схеми social-паттерну при моделюванні вхідного потоку даних являє собою вектор актуаторних сигналів, спрямованих на усунення відмінностей між вхідними даними та social-еталоном .

Логічні схеми social-паттернів інтегруються за допомогою 0 -елемента в соціальний процесор (рис. 11), який також має багатозначні значення вхідних змінних і функції виходу, формують функціональне ставлення, спрямоване на ініціювання або актуацію суб'єктів інфраструктури або користувачів, як реакцію на

вхідний потік великих даних, отриманих шляхом моніторингу social-процесу для їх моделювання: $Y(Z_1, Z_2, \dots, Z_i, \dots, Z_n) = f(Y_1, Y_2, \dots, Y_i, \dots, Y_n)$. Синтез кубітної матриці social-процесу. Суттєвою представляється інваріантність social-патернів щодо часу, що призводить до створення компактних комбінаційних логічних схем, де кожен логічний елемент відповідає за всі використовувані значення істотної змінної. Базовим є варіант, де є велика кількість скріншотів, які створюють послідовності логічних елементів, що утрудняють аналіз social-процесів для вироблення керуючих актюаторних сигналів. Послідовні ланцюжки екранів призводять до необхідності групування екранів в рамках формування моделі комбінаційної схеми social-патерну.

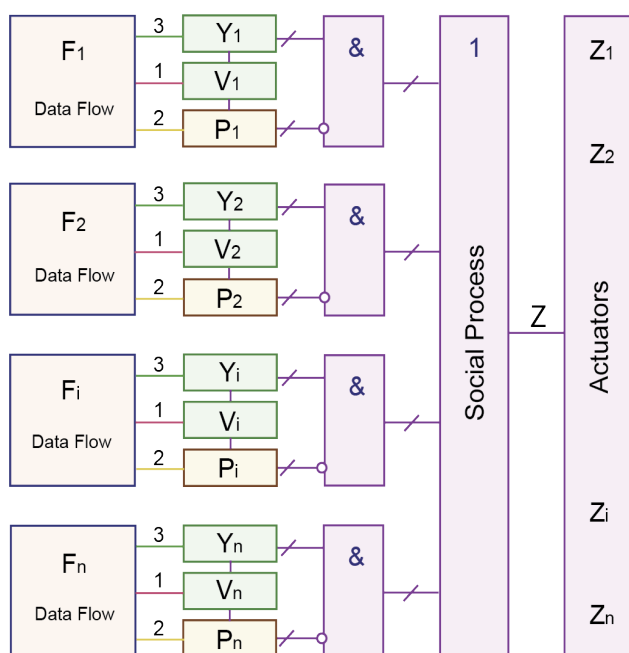


Рис. 11. Логічна схема social-процесу

Таким чином, є два шляхи синтезу social-патерн-схеми: 1) на основі аналізу social-потіку сформувати універсум значень логічних змінних і кубітних векторів social-патерну, до яких застосувати алгоритм моделювання для розпізнавання потоку даних; 2) на послідовності фреймів виконувати процедуру групування екранів використовуючи незмінюваність істотних змінних на тимчасовому інтервалі з метою синтезу комбінаційної логічної схеми, яка інтегрує відносини між значеннями істотних змінних в часі і просторі.

Social-процес, social-патерни, social-змінні і social-символи будуть автоматично (вручну) визначаються стратегією спадного проектування: від ключа-суті до синонімів шляхом формування непересекаючихся класів еквівалентності,

відповідних змінним, які в сукупності створюють Універсам-ми примітивів.

Синтез або витяг універсумів. Система класів-змінних має такі властивості: 1) утворює розбиття; 2) будь-які два елементи з одного класу еквівалентні [1]; 3) будь-які два елементи з різних класів не еквівалентні; 4) об'єднання класів створює універсум примітивів - значень, змінних, social-патернів; 5) перетин класів дорівнює порожній множині. Наступна таблиця ілюструє взаємодію класів еквівалентних підмножин множини 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 синтезу класів змінних або значень, які мають відношення еквівалентностей на основі розбиття вхідного потоку даних, що містить наступні пункти.

1. Вибір елемента $a_1 \in P$ і освіта підмножини (класи) $K_1 \subset P$ що складається з a_1 і всіх елементів, еквівалентних йому:

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

2. Вибір елемента $a_2 \in P$, $a_2 \neq a_1$ і освіта класу $K_2 \subset P$, що складається з a_2 і всіх елементів, еквівалентних йому:

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

3. У результаті виходить система еквівалентних класів, об'єднання яких одно числу P.

Тут сукупність актюаторних впливів формує універсум примітивів для кожної змінної, з яких створюються кубітні логічні функціональності для управління киберсоціальними процесами і явищами, які використовують компараторні процедури або алгоритми для порівняння з еталонами. Універсум актюаторних примітивів дає можливість управляти змінної з допомогою синонімів, кожен з яких активує певний процес або явище. Наприклад, змінна активізації довільного руху об'єкта має такі варіанти, складові універсум: {поїхали, рушили, покотилися, поскакали, полетіли, помчали, потяглися, попрямували, погналі, побігли, пішли, стартовув}. Обов'язковою умовою в підборі синонімів для змінної є їх непересекаємость з

універсумами інших змінних щоб уникнути неоднозначності поведінки об'єкта. Природно, що змінні активізації своїми універсумами створюють команди для управління процесами, пов'язаними з людиною, твариною, машиною, роботом, дроном, комп'ютером, зброєю, технікою, будинком, робочим місцем.

Таким чином, логічна схема для аналізу соціального social-процесу або явища здатна верифікувати створену метрику або структуру, моделювати позитивні або негативні рішення керівника, передбачати соціальні явища в майбутньому, включаючи деструктивні акції, катаклізми, колізії, злети і падіння громадян, соціальних груп і держав. Соціальний social-комп'ютинг, як і універсальний обчислювач, здатний визначати точні рішення з допомогою логічних схем еталонного поведінки людини, соціальної групи або держави. Проблема підлягає вирішенню в майбутньому, полягає у створенні бази алгоритмів або схем, з яких можна технологічно просто синтезувати киберсоціальні метричні social-процесори для моніторингу, аналізу та актуаторного управління медициною, транспортом, наукою, освітою, державою, виробництвом, екологією, юриспруденцією, фінансами. Інша справа, що чиновники усіх рівнів будуть перешкоджати проникненню непідкупної та об'єктивної цифровізації в сферу їх діяльності зі зрозумілих причин. Чим вище киберсоціальна невігластво бюрократичного апарату, тим складніше і довший шлях до процвітання держави і її громадян у сучасному світі. Шлях один, тривалий і еволюційний перманентно вчити майбутню і актуальну політичну еліту елементів моральної кіберкультури управління на основі вичерпного метричного моніторингу соціальних процесів.

6. Матричний секвенсор для розпізнавання соціальних явищ

Проблема і рішення. Проблема киберсоціального комп'ютингу полягає в складності формалізації емоційної логіки людських відносин, які необхідно привести до цифрового детермінізму функціональних відповістей, що виключає ймовірність і невизначеність. Піти від евристики в бік автоматизації синтезу і аналізу логічних схем соціального процесингу для моделювання і передбачення social-колізій від некомпетентних рішень – завдання, актуальна для ринку. Для її вирішення пропонується трансформувати функціонально закінчений потік великих даних до структурованої матричної двійкової формі, яка унітарно кодує безліч ключових понять-змінних і всі можливі їх значення, складові універсум примітивів. Метод для рішення проблеми

представлений синтезом класів еквівалентних відносин на сукупності змінних P :

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

де кожна з них $P_i = \{P_{i1}, P_{i2}, \dots, P_{ik}, \dots, P_{ir}, \dots, P_{in}\}$ приймає універсальну множину вербальних значень, що створюють між собою еквівалентні відносини $P_{ik} \sim P_{ir}$ і утворюють при цьому порожні перетину $P_{ik} \cap P_{ir} = \emptyset$. Таким чином, щоб побудувати непересічні класи еквівалентних вербальних значень кінцевого безлічі змінних, необхідно визначити ключові слова, які не повинні мати між собою смислового перетину, а потім для кожного ключового слова вибрати з тексту синоніми, які не повинні бути присутніми в значеннях інших змінних.

Метрика: визначення, аксіоми і рівняння social-комп'ютингу.

1. Метрика є спосіб вимірювання відстаней $d_i \in D$ між процесами і явищами в просторі заданих параметрів з виконанням аксіоми циклічного конволюційного замикання [2]:

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

2. Вимірювання є процедура визначення відстані між кінцевим безліччю процесів або явищ, відмінних від нуля.

3. Параметрами виступають логічні змінні $P = \{P_1, P_2, \dots, P_i, \dots, P_n\}$, які з заданою ступенем адекватності описують процес або явище.

4. Змінні визначаються за допомогою їх значень $P_i = \{P_{i1}, P_{i2}, \dots, P_{ij}, \dots, P_{in}\}$, число яких не може бути менше двох.

5. Матриця універсумов є упорядкована сукупність змінних і їх значень $U = [P_{ij}] = [U_{ij}]$ для метричного виміру процесу або явища. Унітарно кодований матриця U має одиничні значення за всіма координатами. Матриця універсумов U може бути представлена одним вектором шляхом конкатенації її рядків

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

6. Матриця патерну $Q = [Q_{ij}]$, є підмножина значень змінних універсальної матриці $Q \in U$, яке формує зразок конкретного процесу чи явища. Найбільш зручною формою завдання патерну є матриця унітарного двійкового кодування значень змінних.

7. Матриця вхідних даних $X = [X_{ij}]$ є підмножина (двійкових значень змінних універсальної матриці $X \in U$, яке формує фрагмент реального процесу або явища.

8. Матриця вимірювання $Y = [Y_{ij}] = [Q_{ij}] \oplus [X_{ij}]$ є підмножина $Y \in U$ (двійкових значень змінних універсальної матриці, що формує відмінності в однойменних координатах матриць Q і X шляхом

виконання паралельної хог-операції між паттерном і фрагментом реального процесу або явища.

9. Відстань є скалярна оцінка кількості відмінностей, зазначених одиницями, в однойменних координатах матриць вимірюваних процесів або явищ. Відстань визначається шляхом підрахунку одиничних координат в матриці виміру:

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

10. Функція відмінності є відношення відстані (кількості відмінностей в однойменних координатах) до загального числа координат матриці (вимірювання):

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

Функція відмінності μ не диференціює відношення між матрицями X , Q , які якісно необхідно розділяти на 5 варіантів взаємодії двох множин (векторів, матриць) (рис. 12).

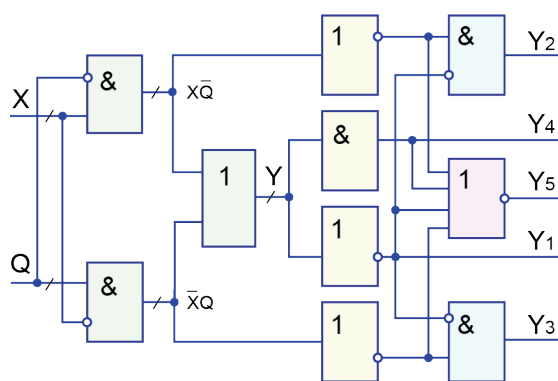


Рис. 12. Матричний метричний процесор

Тут ліва частина схеми складається з трьох елементів

$$Y = X\bar{Q} \vee \bar{X}Q,$$

і являє собою реєстрову або матричну реалізацію хог-операції, де реєстрові виходи всіх трьох елементів далі використовуються для формування вже булевих виходів, які визначають:

1) змінну $Y_2 = [\vee(X\bar{Q})]$, яка дорівнює 1, якщо $X \subset Q$;

2) змінну $Y_3 = [\vee(\bar{X}Q)]$, яка дорівнює 1, якщо $X \supset Q$;

3) змінну $Y_4 = [\wedge Y] = [\wedge(X\bar{Q} \vee \bar{X}Q)]$ яка дорівнює одиниці, якщо результат фіксує розбіжності (одиниці) по всіх координатах виходу Y ;

4) змінну $Y_1 = [\vee Y] = [\vee(X\bar{Q} \vee \bar{X}Q)]$, ка дорівнює одиниці, якщо результат фіксує збіги (нулі) по всіх координатах вектора Y ;

5) змінну $Y_5 = (\bar{Y}_1 \vee \bar{Y}_2 \vee \bar{Y}_3 \vee \bar{Y}_4)$, яка дорівнює одиниці, якщо результат фіксує нульові значення булевих змінних $Y_1, Y_2, Y_3, Y_4 = 0$, що означає наявність часткового перетину між компонентами двох векторів або матриць. Середня частина схеми представлена чотирма елементами, які перетворюють реєстрові змінні в булеві за правилами операцій and, or. Останній елемент or-not виконує функцію визначення самого складно-сформованої взаємодії між векторами $X \cap Q \neq \emptyset$, методом виключення, коли всі попередні чотири відношення не виконуються.

Інтегрально всі п'ять вихідних функцій формують метрику вимірювання якості взаємодії пари процесів і явищ, представлених у форматі матриць або векторів:

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

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

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

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

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

$$Y_5(X \cap Q \neq \emptyset) = (\bar{Y}_1 \vee \bar{Y}_2 \vee \bar{Y}_3 \vee \bar{Y}_4).$$

Остання рівність обчислює найскладнішу взаємодію двох матриць або векторів по методу вилучення: якщо всі попередні чотири змінні на рівні одиницям, то це означає, що два об'єкти мають частинного перетин один з одним.

Обчислювальна складність визначення якості метричного взаємодії двох векторів або матриць дорівнює: $Q = 5Q(Y) + 5Q(Y_i)$, де перший доданок формує час виконання п'яти реєстрових логічних операцій, а друге – час реалізації п'яти булевих функцій.

Наступні таблиці ілюструють обчислювальний процес для визначення якості взаємодії між-ду різними парами довічних векторів X, Q , де перші 4 колонки представлені векторами, а решта представляють собою двійкові бінарні скалярні результати $\{0,1\}$ взаємодії векторів:

X	Q	XQ	X̄Q	Y	Y ₄
1	0	1	0	1	1
0	1	0	1	1	
0	1	0	1	1	
0	1	0	1	1	
1	0	1	0	1	
1	0	1	0	1	

X	Q	XQ	X̄Q	Y	Y ₂
1	1	0	0	0	1
0	1	0	1	1	
1	1	0	0	0	
1	1	0	1	1	
0	0	0	0	0	
0	1	0	1	1	

X	Q	XQ	X̄Q	Y	Y ₃
1	1	0	0	0	1
0	0	0	0	0	
1	1	0	0	0	
1	1	0	0	0	
1	0	1	0	1	
1	0	1	0	1	

X	Q	XQ	X̄Q	Y	Y ₅
1	1	0	0	0	1
0	1	0	1	1	
1	1	0	0	0	
1	0	1	0	1	
1	0	1	0	1	
1	1	0	0	0	

Вирішити проблему неоднозначності при ідентифікації відносини еквівалентності між векторами X, Q можна за допомогою двох елементів корекції стану виходів Y2, Y3. Таким чином, модифікований процесор дає можливість однозначно ідентифікувати всі можливі види взаємодії двох матриць (векторів, множин, текстів) при розпізнаванні патернів, образів, процесів або явищ. Матричний процесор дає можливість піти від необхідності вважати кодове відстань в чотирьох з п'яти випадків якісної взаємодії матриць або векторів. Залишається тільки один варіант, при Y5 = 1, для уточнення взаємодії векторів X, Q шляхом підрахунку функції відмінності $\mu = \frac{d(Q,X)}{n \times m}$. Це необхідно для подальшого коригування матриць Q, X з метою приведення однієї з них до іншої, коли їх відносини трансформуються в одну з чотирьох форм: $X = Q$; $X \subset Q$; $X \supset Q$; $X \cap Q = \emptyset$. Практична спрямованість функції відмінності пов'язана з отриманням скалярних оцінок взаємодії матриць або векторів за всіма видами відносин:

1. Параметри U-матриці універсумів, кількість змінних і сума всіх значень кожної змінної, що в сукупності створює абсолютну оцінку потужності актуальних значень унітарної матриці і коефіцієнт використання координат U-матриці:

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

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

2. Оцінка потужності одиничних значень координат унітарної Q-матриці і коефіцієнт використання значень U-матриці універсумів:

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

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

3. Оцінка потужності одиничних значень координат унітарної X-матриці, коефіцієнт відношення потужностей одиничних значень і функцію відмінності між X- і Q-матрицями для варіантів взаємодії

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

$$(X \cap Q \neq \emptyset):$$

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

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

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

Скалярні оцінки взаємодії надають чисельні докази для якісних тверджень: «збігається», «включає», «покриває», «не перетинається», «різниться». При цьому макроструктура social-комп'ютингу може бути представлена хог-взаємодією матриць X, Q, де потім послідовно вимірює-ся якісна і кількісна оцінки їх відносин, що представлено на рис. 13.

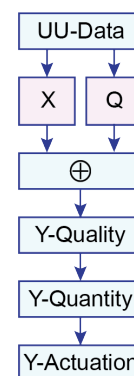


Рис. 13. Макроструктура social-комп'ютингу

4. Функція приналежності є величина, зворотна функції відмінності

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

5. Розпізнавання є процес моделювання для визначення відмінностей між матрицею вхідних даних і матрицями еталонних патернів шляхом виконання паралельної хог-операції з метою отримання матриць вимірювань і функцій відмінності для кожного патерну. Процес моделювання матриці вхідних даних X по матриці паттерна Q, з метою обчислення матриці вимірювання Y і функції μ , представлений двома формулами

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

6. Модель M матричного кіберсоціального social-комп'ютингу (розпізнавання патернів) використовує матрицю U універсумів змінних і їх значень для формування метричного хог-взаємодії тріади похідних матриць (рис. 14): вхідних даних X, патернів Q і вимірювання

(помилки) Y , яке оцінюється скалярними функціями відмінності μ :

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

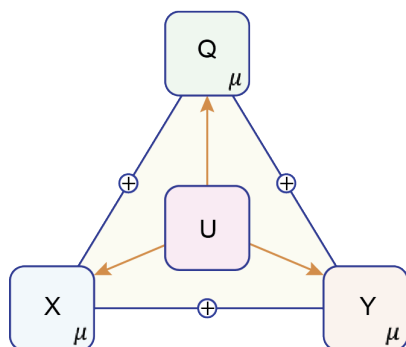


Рис. 14. Модель відносин матричного кіберсоціального social-комп'ютингу

7. Циклічне конволюційне замикання є невід'ємне доповнення метрики, призначене для перевірки валідності вимірювання процесів або явищ. Конволюція перетворює безліч всіх підмножин універсуму примітивів (ідентифікаторів), а також відстані між елементами-вершинами циклу (ідентифікаторов) в пусте безліч (нуль) при їх обробці з допомогою операції симетричною різниці (xor). Симетрична різниця (xor) всіх елементів (двійкових кодів) і їх універсуму дорівнює порожнього безлічі (нулю).

8. Метричний конволюційне транзитивне взаємодія трьох компонентів процесу або явища становить сутність рівняння кібер-соціального (бізнес) комп'ютингу

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

Наприклад, далі наводиться конволюційна хог-взаємодія трьох матриць 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, що по двом відомим матрицям можна відновити будь-яку третю, наприклад, Q, Y виходить матриця X з допомогою хог-операції.

Таким чином, рівняння кібер-соціального оборотного social-комп'ютингу дає можливість паралельно, з допомогою однієї хог-операції, за один автоматний такт отримувати будь-якої з

трьох компонентів social-процесу при двох відомих.

Основні переваги кубітного методу: 1) висока швидкодія social-комп'ютингу за рахунок хронлельной обробки однорідних двійкових двовимірних даних за допомогою двох матричних операцій: xor, car-dinale; 2) простота архітектури social-комп'ютингу, яка визначається хог-взаємодією трьох матриць однакової розмірності; 3) універсальність архітектури хог-взаємодії матриць здатна вирішувати широкий спектр завдань social-комп'ютингу, кіберсоціального комп'ютингу, online управління рухомими об'єктами, розпізнавання образів і цілей, діагностування несправностей; 4) система команд складається з однієї універсальної хог-операції, що дає можливість доставляти процесинг у великі дані, за місцем їх знаходження; 5) орієнтація на memory-driven комп'ютинг дає можливість на 2-3 порядки підвищити швидкодію рішення завдань big data analytics; 6) оборотність комп'ютингу дає можливість визначати будь-яку з трьох матриць за хог-взаємодії двох відомих; 7) структуризація матриці вимірювання дає можливість за одиничним координатах точно коригувати помилкові рішення, пов'язані з неоптимальним над виконанням social-процесу; 8) матриця, як сукупність кубітних покриттів логічних елементів, може бути використана для синтезу і моделювання комбінаційної логіки класичного процесора, тому запропонована матрична архітектура є універсальною не тільки для створення спеціалізованих пристроїв, але і для обчислювальних виробів загального призначення; 9) тріада взаємодіючих матриць кубітних покриттів, як інтерпретативна модель, що володіє властивостями гнучкості, адаптивності і перена-страйваемости, є ринково привабливою для реалізації online комп'ютингу при вирішенні завдань електронного документообігу в рамках технології blockchain smart contract; 10) створення ергономічного інтерфейсу візуалізації процесу моделювання, подібного системі квантового моделювання логічних схем, дає можливість вводити матриці Q, X, Y, U в зручному для користувача форматі, виконувати моделювання-порівняння двох будь-яких матриць в цілях отримання третьої, обчислювати функції відмінності і формувати актюаторные сигнали для виконавчих механізмів enterprise architecture; 11) архітектура матричного social-комп'ютингу являє собою привабливу технологію для electronic Enterprise Architecture (EA) в цілях реалізації human-free моделювання та прийняття правильних рішень проблеми, визначені компанією Gartner Inc.: занадто багато в кіберпросторі аббревіатур, жаргону, спамовських

відволікань і мало досвіду у лідерів в області цифровізації та комунікації компанії. Рішення: врахувати історію, створити метрику бізнесу компанії, планувати дії після проведення аналітики, підтримати лідера у прийнятті виважених рішень щодо можливостей інновацій у бізнесі, використовувати нові технологічні можливості в цифровому бізнесі і в візуалізації планів – роль корпоративного архітектора [https://emtemp.gcom.cloud/ngw/eventassets/en/conferences/epaeu19/documents/gartner-enterprise-architecture-uk-ea-decision-making-2019.pdf].

Алгоритм, що відповідає архітектурі, представленої на рис. 15, характеризується матричною структурою кубитних даних для виконання операцій social-комп'ютингу, який починається з визначення вхідного потоку даних S, що надходить від сенсорів, які обслуговують social-процес компанії, призначеного для синтезу U-матриці універсуму змінних і опису social-процесу з допомогою універсумов вербальних значень кожної змінної, яка слугує базовим форматом для синтезу X-матриці векторів вхідних даних по кожній змінній виконання ітерації моделювання щодо Q-матриці, за дає безліч кубитних двійкових векторів для опису еталонного social-патерну по всіх змінних, що дає можливість на основі &-операції обчислювати функцію приналежності μ , відповідну Y, у вигляді числового значення кількості різних двійкових однойменних координат, отриманого при порівнянні матриць X і Q, визначається за допомогою паралельної операції $(X \wedge \text{not} Q) = Y$, яка дає можливість синтезувати актуаторні вербальні сигнали W, що відповідають одиничним значенням координат вихідної матриці Y у форматі U-матриці, які ініціюють обчислювальні процедури, спрямовані на усунення відмінностей між X-матрицею і еталонної Q-матрицею social-патерну шляхом корекції координат X-матриці з допомогою інфраструктури E, забезпечує виконання social-процесу компанії.

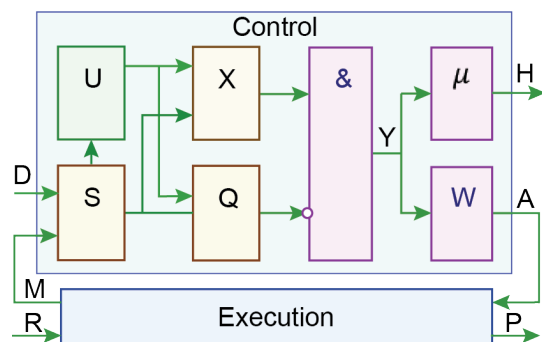


Рис. 15. Логічна матрична схема social-явища

Алгоритм функціонування, представлений на рис. 16, визначає послідовність виконання обчислювальних процедур у часі і просторі для розпізнавання social-патернів у вхідних потоках даних і формування актуаторних сигналів-впливів в цілях усунення відмінностей між ними і починається з блоку (1) ініціювання процедури смислового (2) аналізу вхідних даних S, що надходять від сен-соров F цифрової інфраструктури, при цьому виконується (3) перевірка наявності U-matrix метою переходу на блок 6, а в разі її відсутності активує процедуру розбиття текстових фрагментів на непересічні класи еквівалентностей (4) видобування універсуму ключових слів або змінних-примітивів, де для кожної з них визначається універсум значень, що в сукупності формує U-матрицю універсуму універсумов, яка служить шаблоном для подальшого (5) синтезу кубитної Q-матриці social-патерну і 6) кубитної X-матриці вхідних даних, які далі використовуються для виконання (7) процедури моделювання, яка полягає у виконанні &-операції паралельного порівняння координат кубитних матриць $(X \wedge \text{not} Q) = Y$, що дає можливість отримати результуючу матрицю $Y = [Y_{ij}]$, де кожна 1-координата ідентифікує відмінність між відповідними розрядами матриць X і Q, що дає можливість (8) визначати чисельне значення функції належності μ вхідній X-матриці до Q-матриці social-патерну шляхом підрахунку в Y-матриці 1-координат, які дають можливість (9) формувати вербальну W-матрицю актуаторних сигналів, спрямовану на усунення відмінностей в координатах між актуальною X-матрицею вхідних даних і Q-матрицею еталонного патерну, якщо (10) функція належності більше нуля, шляхом виробки керуючих впливів A для активації (11) інфраструктури E, сенсори якої пов'язані зворотним зв'язком з блоком вхідних даних S, а при відсутності відмінностей між матрицями Q і X, коли $\mu=0$, виконується перехід до блоку (12) закінчення роботи алгоритму, метою якого є social-комп'ютинг, орієнтований на отримання додаткового прибутку за рахунок цифровізації, автоматизації і просторово-часової оптимізації social-процесу при створенні сервісу або продукту.

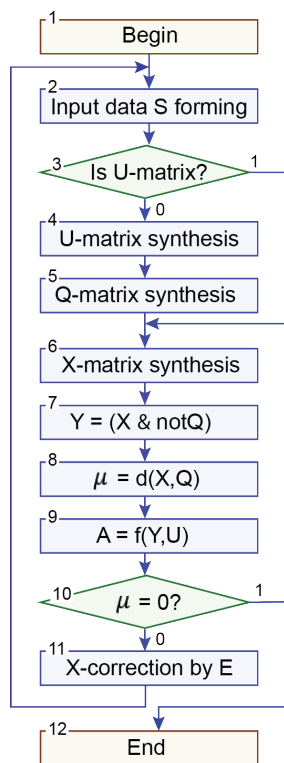


Рис. 16. Алгоритм social-комп'ютингу

Логічна структура, представлена на рис. 17, характеризується кінцевою множиною μ -матриць, обчислюється паралельно, яке формує універсум функцій приладдя чергового вхідного воздей-наслідком паттернам, серед яких вибирається мінімальна оцінка відмінностей.

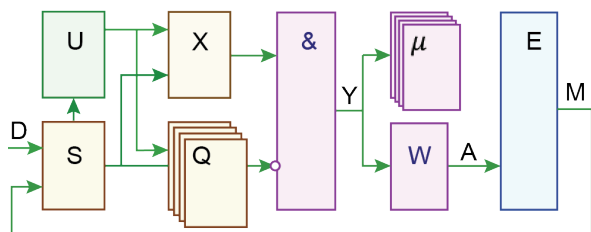


Рис. 17. Логічна матрична схема social-процесу

Логічна структура, представлена на рис. 18, характеризується комп'ютинговою архітектурою social-аналітики, яка має всі вісім компонентів моделі універсального обчислювача, взаємодіючих між собою за формулою: вхід R ініціює команди для виконання social-процесу, який починається з отримання D-ресурсів: фінансових, кадрових, інформаційних, вступників на виконавчий механізм E, що активує сенсори, які передають інформацію по шині моніторингу M для подальшого формування потоку S-даних, призначеного для синтезу U-матриці універсуму змінних для опису social-процесу з допомогою універсумів вербальних значень кожної змінної, яка слугує базовим

форматом для синтезу X-матриці векторів вхідних даних по кожній змінній для виконання ітерації моделювання щодо Q-матриці, що задає безліч кубитних двійкових векторів для опису еталонного social-патерну по всім змінним, що дає можливість на основі &-операції обчислювати функцію приналежності μ , що має вихід візуалізації стану V бізнес-процесу, і відповідну Y, у вигляді числового значення кількості різних двійкових однойменних координат, отриманого при порівняння-нді матриць X і Q, визначається за допомогою паралельної операції $X \wedge \text{not} Q = Y$, яка дає можливість синтезувати актюаторные вербальні сигнали W, що відповідають одиничним значенням координат вихідної матриці Y у форматі U-матриці, які по шині A ініціюють обчислювальні процедури, спрямовані на усунення розбіжностей між X-матрицею і еталонної Q-матрицею social-патерну шляхом корекції координат X-матриці з допомогою інфраструктури E, забезпечує виконання social-процесу компанії для отримання сервісів або продукції P.

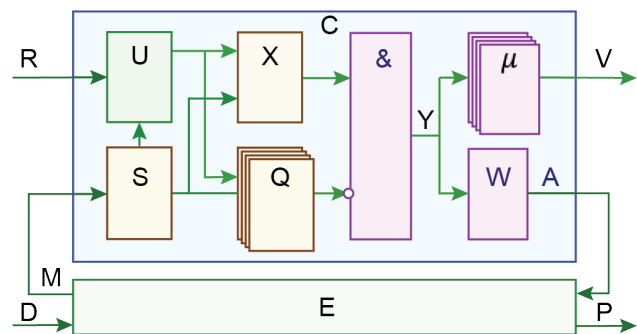


Рис. 18. Логічна матрична паралельна схема social-комп'ютингу

Алгоритм, відповідний архітектурі, представленої на рис. 18, характеризується використанням ку-бітних матриць даних для виконання операцій social-комп'ютингу, структура якого містить механізми-ми управління C і виконання E, з'єднані шини моніторингу M і актюації A, де блок управління має зовнішній вхід команд R і вихід візуалізації стану social-процесу V, а блок виконання містить зовнішній вхід даних D і вихід сервісів або продукції P, що в сукупності становить вісім компонен-тів, що характеризують social-комп'ютинг, який починається з визначення вхідного потоку даних S, по-ступає від сенсорів виконавчого механізму, обслуговуючого social-процес компанії, який призначений для синтезу U-матриці універсуму змінних і опису social-процесу за допомогою універсумів вербальних значень кожної змінної, яка слугує базовим шаблоном для синтезу X-

матриці вхідних даних і подальшого моделювання спільно з уже певними Q-матрицями, які задають безліч еталонних social-патернів, що дає можливість на основі &-Операції паралельно обчислювати n матриць $Y_i = (X \wedge \text{not} Q_i)$, ($i = 1, n$) моделювання, аналіз яких на мінімальне число одиничних координат $\min(Y_i = 1)$, ($i = 1, n$), дає можливість визначити номер i матриці, що має мінімальне значення з n функцій приладдя

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

формують чисельні значення відмінностей двійкових однойменних координат, отриманих при порівнянні матриці X і n матриць Q, що дає можливість синтезувати матрицю актюаторних вербальних сигналів W, відповідну одиничним значенням координат вихідної матриці Y, що має $\min \mu$, у форматі вербальних значень U-матриці, які ініціюють обчислювальні процедури у виконавчому механізмі E, спрямовані на усунення розбіжностей між X-матрицею і Q-матрицею social-патерну с $\min \mu$, шляхом корекції координат X-матриці, що забезпечує оптимальне виконання мети P social-процесу компанії.

Алгоритм, представлений на рис. 19, визначає послідовність процедур комп'ютингу у часі і просторі для розпізнавання social-патернів у вхідних потоках даних і вироблення актюаторних сигналів-впливів в цілях усунення відмінностей між ними, що починається з блоку 1) ініціювання алгоритму з подальшим переходом до 2) блоку смислового аналізу вхідних даних S, що надходять від входу D і сенсорів M цифрової інфраструктури, при цьому виконується 3) перевірка наявності U-matrix в цілях перехід на блок 6, а в разі її відсутності алгоритм активує процедуру розбиття текстових фрагментів на непересічні класи еквівалентностей 4) видобування універсуму ключових слів або змінних-примітивів, де для кожної з них визначається універсум значень, що в сукупності формує U-матрицю універсуму універсумов, яка служить шаблоном для подальшого 5) синтезу кубітних Q-матриць social-патернів і 6) кубітної X-матриці вхідних даних, які далі використовуються для виконання 7) паралельної &-операції і отримання n матриць $Y_i = (X \wedge \text{not} Q_i)$, ($i=1, n$) моделювання, в яких під-зчитується число одиничних координат $\min(Y_i=1)$, ($i=1, n$), дає можливість 8) визначити номер i матриці, що має мінімальне значення з n функцій належностей

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

формують чисельні значення відмінностей довічних однойменних координат, отриманих при порівнянні матриці X і n матриць Q, що дає можливість 9) синтезувати матрицю актюаторних

вербальних сигналів W, відповідну одиничним значенням координат вихідній матриці Y, що має $\min \mu$, в форматі вербальних значень U-матриці, які, 10) при $\mu \neq 0$, ініціюють 11) обчислювальні процедури в виконавчому механізмі E, спрямовані на усунення розходжень між X-матрицею і Q-матрицею social-патерну с $\min \mu$, шляхом корекції координат X-матриці, що забезпечує оптимальне виконання мети P social-процесу компанії, а при існуванні $\mu \neq 0$ виконується завершення роботи алгоритму на заданій матрице вхідних впливів, який забезпечує реалізацію social-комп'ютингу, орієнтованого на отримання додаткового прибутку компанії за рахунок цифровізації, автоматизації та просторово часової оптимізації social-процесу при створенні сервісу або продукту.

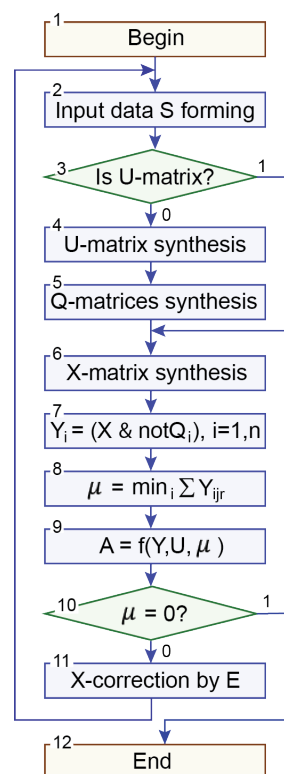


Рис. 19. Алгоритм social-комп'ютингу

Таким чином, матрична структура даних для паралельного виконання операцій social-комп'ютингу дає можливість суттєво підвищити швидкодію інтерпретативного паралельного моделювання вхідних потоків даних для розпізнавання в них social-патернів метою генерування актюаторних сигналів, усувають відмінності в X-матриці по відношенню до одного з еталонних social-патернів.

7. Висновок

Наукова новизна публікації полягає в розробці кубитного логічного секвенсора для моделювання і розпізнавання social-явищ у великих потоках даних на основі створення інтерпретативних кубитних матричних моделей, методів і архітектур киберсоціального social-комп'ютингу, спрямованого на автоматичний синтез і аналіз логічних схем, орієнтованих на аналіз, моніторинг і управління соціальними процесами і явищами:

1) Удосконалено архітектура memory-driven киберфізического комп'ютингу, яка відрізняється паралелізмом процедур синтезу і аналізу логічних схем, призначених для моделювання social-процесів і явищ в цілях моніторингу і управління.

2) Удосконалено кубитно-матричні моделі логічних схем, які відрізняються унітарною кодуванням багатозначних змінних для синтезу секвенсорів з метою паралельного аналізу киберсоціальних процесів.

3) Вперше запропоновано кубитно-векторні, матричні методи синтезу і аналізу логічних схем, які характеризуються унітарним кодуванням значень багатозначних змінних для паралельного моделювання social-процесів і явищ.

4) Розроблені кубитно-матричні моделі, структури даних, методи синтезу і аналізу логічних схем social-процесів дозволяють моделювати реакцію соціальної системи на прийняття конструктивних і деструктивних рішень людиною, керівником, чиновником, завдяки кубитному опису еталонів поведінки, що дає можливість актуально управляти громадянами та інфраструктурою.

5) Напрями майбутніх досліджень. Так само як біологічні віруси деструктують людини, з-спеціальні віруси (корупція, злочинство, тероризм, забруднення планети, бунти, революції, війни) але-ють організм людства в масштабах планети, несучи в світ інший мільйони життів. Причиною цього завжди є аморальність і некомпетентність політичної еліти державних утворень, яка формує соціальні хвороби «брудних рук». Якщо врахувати, що зцілити політичну еліту можна шляхом тривалого процесу її «санітарного» освіти, необхідно шукати альтернативні технології боротьби проти соціальних вірусів. Одним з можливих варіантів може бути соціальний імунітет, як киберфізический моральний соціальний комп'ютинг метричного вичерпного моніторингу всіх процесів і явищ для цифрового human-free управління громадянами на основі моделювання і передбачення остан-наслідків від прийняття рішень. Комп'ютинг завжди, скрізь і в усьому

пояснює природу процесів та явищ, допомагає вирішувати актуальні питання гармонійного та морального розвитку людства. Ось кілька при-
мерів комп'ютингу. Всесвіт має гармонійний геном циклічної зміни в метриці: матерія і енергія, простір і час – Космологічна Computing. Людство прагне до гармонійного зраді-ню моральних відносин для досягнення соціальної справедливості шляхом цифрового моніторингу та автоматичного управління за наявності ресурсів у метриці: матерія і енергія, простір і час – Humanity Computing. Геном людини формує гармонійний напівцикл зміни шляхом зародження, розвитку, старіння і вмирання в метриці ресурсів: матерія і енергія, простір і час – Human Computing. Геном комп'ютингу також має гармонійну форму розвитку технологій, елементної бази, системних, алгоритмічних, програмних і архітектурних рішень у метриці: матерія і енергія, простір і час – HW-SW Computing. Кибермедицинський комп'ютинг (КМК) – довічний моніторинг душевного і фізичного здоров'я кожної людини з моменту його народження в цілях активного керування його поведінкою у форматі 24/7 для запобігання хвороб шляхом створення цифрового асистента, допомагає приймати оптимальні рішення щодо стратегії і тактики поведінки для забезпечення високої якості життя. КМК є альтернативою для стратегії сучасної медицини, що полягає у лікуванні хвороб, отриманих у результаті неправильного вибору повсякденних і довгострокових рішень, пов'язаних з незнанням функціональних особливостей свого організму і впливу на нього навколишнього дійсності. Запобігати хворобам шляхом моделювання можливих варіантів поведінки, а не пояснювати, чому вони сталися), забезпечувати якість життя, а не якість лікування на основі перманентного метричного моніторингу стану душі, тіла і навколишнього середовища в цілях цифрового оптимального управління поведінкою людини. При цьому корекція природних помилок і отриманих травм є лише корисним доповненням до засобів забезпечення якості життя людини. Біокомп'ютинг, як моніторинг та управління біологічними процес-сами, є найбільш суттєвою областю у пізнанні життя (синтез і аналіз), яка найменш вивчена вченими і практиками. Адитивний комп'ютинг являє собою безвідходну технологію вирощування або 3(4)D-друкування комп'ютерів і компонентів, технічних конструкцій, будинків, продуктів харчування, біоінженерних частин людського тіла.

6) Окремі висновки. Беручи до уваги дизрапторную аксіому, що комп'ютинг, як процес, що є первинним по відношенню до явищ,

які він пояснює, породжує, обслуговує і використовує, можна зробити кілька істотних практично орієнтованих висновків. Замість універсального цільового критерію ефективності: час-гроші-якість, вводиться вимір процесу або явища в метриці двох взаємодіючих пар: простір-час, матерія-енергія. Будь-який процес спрямований на зміну явища в метриці параметрів простір-час, матерія-енергія для досягнення мети. Процес реалізується, реально чи віртуально, в архітектурі або моделі комп'ютингу, який визначається вісьмома взаємодію-складовими компонентами: мета, відносини, візуалізація, управління, виконання, ресурси, моніторинг актуалізація. Будь-який соціальний процес може і повинен бути реалізований у форматі комп'ютингу, де головною відмінністю від всієї передісторії людства є цифровий моніторинг і актуаторное human-free online управління. Киберсоціальний комп'ютинг має сенс лише у випадку прямої і безпосередньої взаємодії кожного громадянина з будь-якими сервісами моніторингу і керування, що усувають армію чиновників. Видавництво Springer анонсувало пропозиція авторам з написання серії книг в напрямку Computational Social Sciences. Такий інтерес видавництва підігрівается цифровим ринком, який вимагає більш адекватної поведінки політичної еліти держав по відношенню до своїх народів, а також корекції політичних рішень, який цілком можлива в рамках реалізації Cyber Social Computing [32].

Література:

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. U. Reinsalu, J. Raik, R. Ubar and P. Ellervee, "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, pp. 164-170, 2011.
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. C. Zhu, VCM Leung, L. Shu and ECH Ngai, "Green Internet of Things for Smart World," in IEEE Access, vol. 3, pp. 2151-2162, 2015.
11. A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities," in IEEE IoT Journal, vol. 1, no. 1, pp. 22-32, Feb. 2014.
12. Frahm J. Securing the Internet of Things: A Proposed Framework / J. Frahm // 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) .- vol. 1.- Berlin, Heidelberg: Springer International Publishing.- 2017.
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). - vol. 2.- Berlin, Heidelberg: Springer International Publishing.- 2017.
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. Volume 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 Conferences on Big Data and Cloud Computing, Social Computing and Networking, 2016. Pages: 274 – 283.
22. Maria R. Lee, Tsung Teng Chen. Understanding Social Computing Research. IT Professional. 2013. Volume: 15, Issue: 6 Pages: 56 – 62.
23. Jerry Higg, Varadraj Gurupur, Murat Tanik. A Transformative Software Development Framework: Reflecting the paradigm shift in social computing. 2011 Proceedings of IEEE Southeastcon. 2011 Pages: 339 – 344.
24. 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. Year: 2018, Volume: 5, Issue: 3. P. 134-144.
25. Christopher M. Bishop. Pattern Recognition and Machine Learning, 2006. Springer. 738 p.
26. Susan Stepney, Steen Rasmussen, Martyn Amos. Computational Matter. Springer. 2018. 336 p.
27. Zhuge Hai. Cyber-Physical-Social Intelligence. On Human-Machine-Nature Symbiosis. Springer. 2020. 336 p.
28. Elias G. Carayannis. David F. J. Campbell. Marios P. Efthymiopoulos. Handbook of Cyber-Development, Cyber-Democracy, and Cyber-Defense. 2018. 1089 p.
29. Vladimir Hahanov; Svetlana Chumachenko; Eugenia Litvinova; Anastasia Hahanova. Cyber-physical social monitoring and governance for the state structures. IEEE 9th International Conference on Dependable Systems, Services and Technologies. 2018. P.123-127.
30. Vladimir Hahanov; Svetlana Chumachenko; Eugenia Litvinova; Abdullayev Vugar Hacimahmud; Anastasia Hahanova; Tetiana Soklakova. Cyber Social Computing. 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/marked+items?SGWID=0-1741513-93-0-0>
- Надійшла до редколегії 23.12.2019
- Рецензент: д-р техн. наук, проф. Дрозд О.В.
- Хаханова Ганна Володимірівна**, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: комп'ютерна та кіберсоціальна інженерія. Адреса: Україна, 61166, Харків, 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, email: anna.hahanova@nure.ua