

СИСТЕМИ І ПРОЦЕСИ УПРАВЛІННЯ

УДК 681.326:519.613

МОДЕЛІ ПРОЦЕСИНГУ ДЛЯ РОЗРІЗНЕННЯ КІБЕР-СОЦІАЛЬНИХ ЯВИЩ

ХАХАНОВА Г.В.

Пропонуються моделі і методи розрізнення соціальних явищ, що використовують матричні структури даних для завдання еталонних патернів. Описується модель social-процесу, яка містить компоненти: потік, функціональність, фрагмент, фрейм, параметр, значення, логіка, графіка, що знаходяться в логічних відносинах один з одним. Пропонується Pattern Recognition Social-Processor, який використовує хог-операції для розпізнавання соціально-значущих фреймів інформації.

Описується матрична автоматна модель моніторингу процесів і явищ для подальшої актуації соціальних дій в рамках створення компонентів соціального комп'ютингу. Видається ієрархічна модель структурних компонентів соціального комп'ютингу, які оформлені в відносини включення кожного елемента нижнього рівня в компонент, що покриває його.

Описується архітектура процесингу для визначення подібності фреймів на основі унітарного кодування даних і хог-порівняння для обчислення подібності-відмінності. Описується метрика для вимірювання подібності геометричних фігур в кіберпросторі, яка оперує нормованими параметрами: координата, орієнтація, суперпозиція, масштабування. Пропонується перетворення скалярною матриці подібності в бінарну шляхом примусового введення threshold або порогового значення параметрів, що дає можливість істотно спростити синтез автоматних моделей соціального процесингу.

Ключові слова – соціальний процес, розпізнавання патернів, фрагмент і фрейм процесу, унітарна кодування, схожість-відмінність, автоматні моделі соціального комп'ютингу.

1. Вступ

Великі дані (десятки джетгабайтів), які знаходяться в кіберпросторі, на 96 відсотків є незатребуваними, завдяки нестачі ефективних моделей і методів їх класифікації, кластеризації, розпізнавання і структуризації. Затребуваність таких технологій визначається запобіганням кримінальних актів, тероризму та антисоціальних дій, які все більш мають відображення в кіберпросторі. Поява продуктивних моделей і методів дозволить в режимі онлайн розпізнавати в інтернеті за ключовими словами наміри до деструктивних дій і запобігати їм. Цьому присвячено значну кількість

публікацій (13 644 – Springer, 95 052 – IEEE Xplore), серед яких є технології на які слід звернути увагу [1-12]. Тут основні ідеї пов'язані із застосуванням технік Machine Learning, глибоких нейронних мереж, а також з автоматним дискретним розпізнаванням патернів на основі хог-операції і синтезу еталонних соціальних таблиць істинності [13-18].

Мета – ідентифікація вхідного потоку X social-даних шляхом їх хог-моделювання на основі заданих патерн-зразків social-функціональностей Q , що дає можливість визначити час виконання закінченої соціальної функції, час відволікання на несуттєві операції, а також показати шляхи поліпшення якості сервісного обслуговування.

Завдання: 1) Парсеризація або конвертація вихідних даних в кубітно-матричну модель social-процесу. 2) Виконання хог-моделювання вхідного потоку X матричних даних щодо еталонних патернів Q . 3) Формування Y -матриці, як результату хог-взаємодії компонентів $Y=Q\oplus X$, а також якісних і кількісних оцінок їх відносин. 4) Створення software-модуля для формування таблиці інтегрального часу виконання social-функціональності, а також порційного часу виконання кожного фрагмента social-патерну по заданому параметру у вигляді ID патерн-зразка. 5) Визначення оптимального і ергономічного формату вихідних даних для візуалізації часу і кроків виконання реального social-патерну. 6) Тестування модуля на представницької вибірці патернів і вхідних потоків даних. 7) Дослідна експлуатація програмного додатка.

Аргументом для розробки нової програми є наявність ненульової похідною між базовим і новим варіантом. Базовий варіант визначається ручним проектуванням фактичних моделей social-процесів із застосуванням допоміжних сервісів, що дає можливість отримувати статистику ефективності роботи співробітників для подальшого управління персоналом. Новий варіант відрізняється від базового напівавтоматичним синтезом патернів, як двійкових еталонних моделей social-функціональностей, а також автоматичним аналізом-моделюванням вхідних даних, що дає можливість розпізнавати в ньому social-патерни Q , формувати функцію приналежності, статистику ефективності процесів і online управляти кадрами -інфраструктурою компанії або державної організації для отримання більшого прибутку.

2. Визначення моделі соціального процесу

1) Характеристичне рівняння моделювання social-процесу W для розпізнавання Q -функціональностей у вхідних даних R , з метою отримання social-реалізації Y з функцією приналежності, має вигляд: $Y=R\oplus Q$. Метрика для вимірювання фрагментів вхідних даних R , функціональностей Q і

вихідних даних Y має параметри: $Image=(Parameter, Value, Logic, Graphics)$. 2) $Image$ – модель компонента, що містить зображення, параметри, значення і правила: $S_i=\{G,P,V,R\}$, $S_i \in \{F,Q,I\}$ для освіти інформаційних структур. 3) $Social-process(or)$ – сукупність $social$ -функціональностей, представлених у вигляді послідовностей, які формують сервіси, з метою отримання прибутку: $Q=\{Q_1, Q_2, \dots, Q_i, \dots, Q_n\}$. 4) Фрейм - послідовність фрагментів, що задає виконання $social$ -функціональності (патерну), об'єднана логікою $T=(S_1, S_2, \dots, S_i, \dots, S_n)$, $\bigcap_{i=1}^n S_i = R_j$ заповнення полів форми на картині, яка містить $image$, $Parameters$ and $Logic$ (зображення, параметри і правила): $F=(S_1, S_2, \dots, S_i, \dots, S_m)$, $S_i=\{G,P,R\}$. 5) Потік даних - послідовність або безліч фреймів, які формують фрагмент алгоритму $social$ -функціональності. 6) Алгоритм - послідовно-паралельна комбінація потоків, що використовує логічні умови для виконання альтернативних гілок $social$ -функціональності. 7) Патерн Y - практична реалізація $social$ -функціональності у вигляді послідовності фрагментів, отриманих в результаті взаємодії потоку даних з фреймами $R \oplus Q = Y = (Y_1, Y_2, \dots, Y_i, \dots, Y_k)$, $\bigcap_{i=1}^m Y_i = D_j$, створюють потік. 8) $Process$ - структурована запис R вхідний послідовності фреймів в формі файлу, призначена для розпізнавання в ньому $social$ -функціональностей і інформаційного шуму. 9) Графіка - зображення фрейму, ідентифіковане моментом часу: $G=g(t)$. 10) $Parameters$ – сукупність атрибутів або змінних, характеризуючих всі компоненти фрейму вербально-чисельними значеннями: $P=(P_1, P_2, \dots, P_i, \dots, P_n)$. 11) $Logic$ – суперпозиція значень параметрів за допомогою логічних операцій and (or , not , xor) для об'єднання послідовності фреймів в алгоритм: $R=(R_1 \wedge R_2 \wedge \dots \wedge R_i \wedge \dots \wedge R_k)$. 12) $Pattern (Q)$ – структура фреймів T для еталонної реалізації $social$ -функціональності, об'єднана одним D : $Q=(T_1, T_2, \dots, T_i, \dots, T_m)$, $\bigcap_{i=1}^m T_i = D_j$. 13) Модель $social$ -процесу містить наступні компоненти: $W = \langle R, Q, Y, E, S, F, M, P, V, L, G \rangle$, що відповідно означає: $Process$, $Pattern$, Функціональність, Фрагмент, Фрейм, $Flow$, $Parameter$, $Value$, $Logic$, $Graphics$. Компоненти моделі знаходяться в наступних (xor - and $contain$ -) відносинах один з одним: $Y=R \oplus Q$, $\{R, Q, Y\} = \{E, S, F, M, P, V, L, G\}$. Реалізація двох видів відносин представлена на рис. 1.

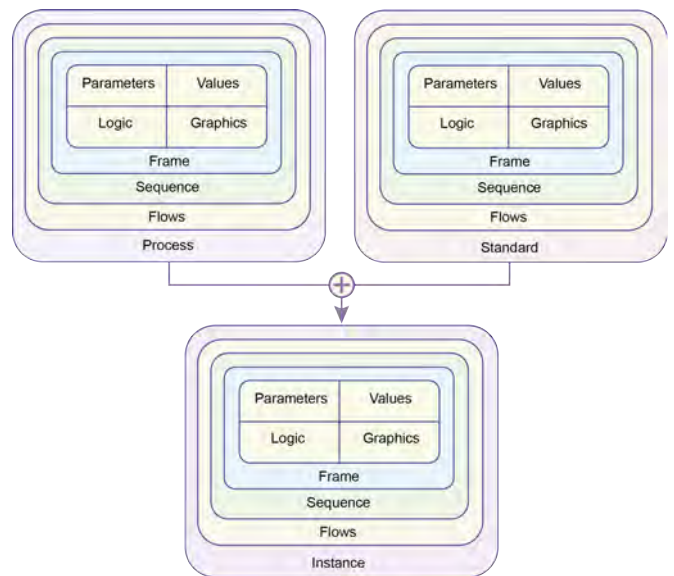


Рис. 1. Архітектура відносин в $social$ -процесі компанії

Структури і візуалізація вихідних даних формуються шляхом моделювання вхідного потоку даних, залежного від часу, на логічній матричній структурі патернів, інваріантних до часу: 1) Функція якості, характеризуюча всі види теоретико множинні відносини між ($Instance$, $Sequence$, $Frame$) і $Pattern Y(I, Q)$, $Y(T, Q)$, $Y(S, Q)$: $(Y_1, Y_2, Y_3, Y_4, Y_5)$. 2) Кількісна, приведена до інтервалу $(0,1)$, оцінка відносини як функція або ступінь відмінності між вхідними даними і $Pattern \mu(I, Q)$, $\mu(T, Q)$, $\mu(S, Q)$. 3) Час (t_1, t_2, t) початку і кінця виконання функціонального фрагмента, а також його тривалість. 4) Параметри і значення матриці Y , які відрізняються від патерну, максимально близького до функціонального фрагменту вхідного потоку. 5) Формат згаданих вихідних даних зводиться до таблиці, яка має наступну структуру колонок: $\langle ID, Y_i, \mu, t_1, t_2, t, Y \rangle$.

No	Q-ID	Y_i	μ	t_1	t_2	t	Y
1	loan	3	0,15	11.15	11.20	5	0100

3. Матричний $social$ -процесор

Візуалізація вихідних даних - результатів моделювання в процесі розпізнавання вхідного потоку даних $flow$ (рис. 2) має на меті створити ергономічне прочитання експертом або керівником макро-рисунок $social$ -процесу, його деталізацію, при необхідності, а також вироблення коригувальних актуаторних впливів, при істотних відхиленнях від $social$ -функціональності, виконуваних співробітником компанії.

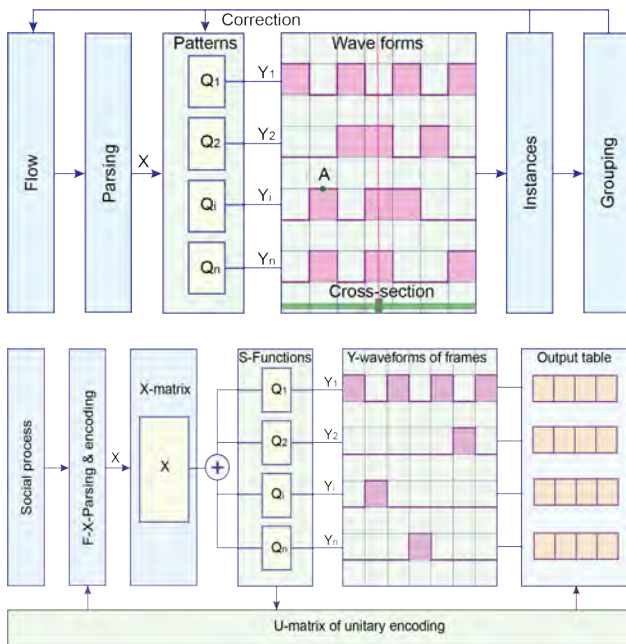


Рис. 2. Pattern Recognition Social-Processor

Архітектура social-процесора (Pattern Recognition Matrix-Social-Processor) містить наступні компоненти: Flow - це потік вхідних даних, що містить символно-графічну інформацію, яка подається на Parser-модуль, що конвертує дані в матричну форму, де рядки представлені значеннями змінних, які структуровано і метрично описують вхідні дані про фрейм, які далі здійснюють паралельне хог-взаємодія з усіма патерн-матрицями логічного процесора (Matrix processor), результатом якого є формування функцій приналежності вхідного фрейму до кожного патерну, що створює двовимірну двійкову (багатозначну) просторово-часову таблицю значень функцій приладдя (Y-waveforms), з яких формуються відповідні вейвформи, що ілюструють збіг вхідних потоків фреймів з social-патернами, що далі використовується для синтезу матриці Instances, яка представляє собою компактну запис кожної вейвформи, з якої виключені активності, що не мають відношення до виконання конкретної social-функціональності, що міститься в матриці Instances, яка далі обробляється в модулі оптимізації з метою отримання мінімальної матриці кроків виконання social-патерну шляхом групування фреймів за правилом наявності еквівалентних значень змінних в скріншотах, що дає можливість зменшити таблицю вихідних даних Y, а також використовувати її для корекції матриць патернів Q і вхідних даних X відповідно до рівнянням social-комп'ютингу $Q \oplus X \oplus Y = 0$, в разі фіксації помилкових дій операторів.

Найбільш оптимальною в просторово-часовій метрики є тимчасова діаграма social-процесу по вихідним змінним, число яких відповідає кількості social-патернів (виявлених) у вхідному потоці. Кожна змінна приймає значення $\{0,1\}$, які інтерпретуються низьким або високим потенціалом тимчасової діаграми. Як правило, social-процес, знятий з одного каналу даних, є послідовним. Це означає, що в кожен момент часу активною є лише одна вихідна змінна, відповідна патерну. Однак, при впровадженні робота в social-процес підприємства активними можуть бути всі social-патерни (вихідні змінні). Можна передбачити і третій стан вихідної змінної, що буде відповідати пасивного значенням social-патерну в режимі очікування.

Деталізація вихідних даних дає можливість показувати стану всіх атрибутів social-процесу: змінні, правила, фрейми в табличному вигляді при вказівці мишкою на точку (A) діаграми конкретної вихідної змінної. Можна зробити зріз даних (Cross section) по всім змінним social-процесу в заданий момент часу.

Процесор online-візуалізації призначений для розуміння структури social-процесів експертом і керівником, з метою вироблення коригувальних актуаторних впливів по оптимізації роботи персоналу. Структурні компоненти процесора: 1) Вхідний потік даних flow, який підлягає моделюванню з метою визначення приналежності фреймів в заданих social-патернах. 2) Матриці патернів складають зразки функціональностей social-процесу, щодо яких здійснюється порівняння фрагментів вхідного потоку даних. 3) Вихідні змінні, відповідні social-патернах, формують тимчасові діаграми, які в режимі online ідентифікують своїми одиничними значеннями стану функцій приладдя вхідних впливів до патернів. Ідеальне рішення пов'язане з такою тимчасовою діаграмою, яка показує інтервальну оцінку приналежності вхідного впливу до кожного патерну. При цьому рівень дискретності тимчасової діаграми вже визначається значністю з більш двох варіантів, що істотно підвищує адекватність моделі реальним social-процесам. Сукупність тимчасових діаграм на фреймі комп'ютера є ідеальною і найбільш ергономічною моделлю візуалізації процесу розпізнавання патернів, переваги якої дають можливість: 1) Спостерігати на часовому зрізі функції приналежності кадру з вхідного потоку даних flow до кожного social-патерну. 2) Бачити на кожній патернової Y_i-вейвформе кількості поруч розташованих фреймів з одиничним значенням, що підлягають гуртування в фрагмент. 3) Спостерігати в режимі online активність

виконання співробітниками кожної social-функціональності в масштабі робочого місця і / або компанії. 4) Перевіряти валідність моделей social-патернів шляхом перетинання одиничних значень на вейвформах для їх подальшого коректування. 5) Управляти по зворотньому зв'язку діями співробітників компанії і спостерігати якісні зміни на вейвформі з затримкою на реакцію оператора. 6) Здійснювати ручне групування фреймів шляхом вказівки зони об'єднання на вейвформі змінної. 7) Визначати ефективність виконання співробітником social-функціональності шляхом підрахунку кількості перемикачів на вейвформі змінної social-патерну. 8) Підраховувати інтегральний час роботи співробітника за робочий день шляхом підсумовування активностей по всіх вейвформах змінних. 9) Підраховувати час виконання social-функціональності шляхом підсумовування активностей на вейвформі змінної. 10) Визначати ефективність роботи співробітника(ів) на просторово-часовому форматі social-процесу шляхом підрахунку відношення площі, зайнятої одиничними значеннями, до загальної площі вейвформ:

$$E = \frac{\sum_{j=1,k}^{t=1,m} W_{tj}}{k \times m}$$

Вхідний потік даних flow в результаті його фільтрування (просторового розподілу) на патерн-процесорі диференціюються в реалізації social-функціональностей, якість яких необхідно оцінювати за часом їх виконання і кількості помилок дій, неявно задаються функціями приладдя. Валідність вейвформ пов'язана з існуванням лише однієї одиниці на вертикальному зрізі, що означає приналежність фрейма тільки одному патерну.

Природньо, що замість тимчасових діаграм можливо виводити послідовність чисельних значень функцій приладдя вхідних даних до social-патернів, що є менш читабельною формою інформації для експерта або керівника.

Чисельні значення функцій приналежності є актуаторними сигналами для формування instance-сегментів, які підлягають гуртуванню за правилами еквівалентності значень істотних змінних. Послідовність одиничних значень на тимчасовій діаграмі по одній змінній Y_i візуалізує можливість групування фреймів в один фрагмент.

Таким чином, в процесі проектування online social-аналітики створюються структури даних, які диференціюються на три типи: 1) Вербальні вхідні послідовні дані- social-поток. 2) Матричні паралельні моделі патернів, що формують логічний social-процесор. 3) Послідовні вихідні дані, відповідні кадрам, які отримані в результаті

групування за правилами еквівалентності значень істотних змінних.

Актуация або управління social-процесом є головним механізмом комп'ютерингу. Комп'ютер створюється для виконання певних дій при наявності апріорно заданих умов. Це означає, що таблиця істинності формує вхідні умови для функції виходів, яка може дорівнювати нулю або одиниці. Інакше, одиничний сигнал виходу логічного елемента включає лампочку, запускає двигун, звільняє працівника, присуджує премію. Тому створення соціального комп'ютерингу пов'язано з аналогічним детермінізмом закінченого циклу перевірки (моніторингу) умов і вироблення керуючого впливу при їх виконанні. При використанні апарату матриць значень змінних, які еквівалентні деяким умовам, необхідно їх доповнювати актуаторними сигналами функції виходів для управління механізмами соціальної інфраструктури з метою отримання якісних, матеріальних, тимчасових або моральних бонусів. Для мінімізації таблиці істинності можна формувати умови, які визначають одиничні значення функції виходів, що відповідає диз'юнктивній нормальній формі. Такий підхід є більш конструктивним при створенні матриці позитивних умов, які формують 1-значення вихідний актуаторної функції. Звідси випливає простий висновок - кожен патерн, як зразок social-функціональності, повинен мати актуаторний сигнал, спрямований на виконання конкретної дії роботом, людиною, керівником, соціальною групою, яке слід явно вказувати в патерн-матриці. Інше рішення пов'язане з формуванням матриці актуаторних впливів, в якій перелічені всі управлінські сигнали, відповідні впорядкованій послідовності патернів, що мають місце бути в компанії. Конкретний сигнал виконується і активує компонент інфраструктури, якщо сукупність вхідних впливів покриває відповідний патерн. Також можливим є варіант формування матриці актуаторних впливів, де показані сигнали управління інфраструктурою компанії, відповідні виконанню певних умов розглянутого патерну. В даному випадку мова йде о двох матрицях, де кожному вектор-рядку Q-матриці умов (патерну) ставиться у відповідність рядок A-матриці актуаторних дій.

M	Q	D(Q)	-	A	D(A)
1	Q1	D(Q1)	-	A1	D(A1)
2	Q2	D(Q2)	-	A2	D(A2)
3	Q2	D(Q3)	-	A3	D(A3)

Дана структура нагадує матрицю переходів-виходів алгоритму для синтезу цифрового пристрою управління обчислювальним процесом, яка відрізняється від двійкової по виходах таблиці

істинності багатозначністю завдання функцій виходів.

4. Ієрархічна модель social-процесів

Для створення social-комп'ютингу необхідно розуміти ієрархічну модель структурних компонентів, які можуть бути оформлені в наступні відносини включення:

1) Компанія - social-процеси. Як правило, компанія має кілька складових, а також підпорядкованих social-процесів, які призначені для створення продукції та сервісів з метою отримання прибутку. З позиції цифрового моніторингу керівнику (менеджеру) необхідно бачити online картину активності social-процесів на фреймі (рис. 3) з метою оперативного прийняття актуаторних впливів, спрямованих на мінімізацію простоїв.

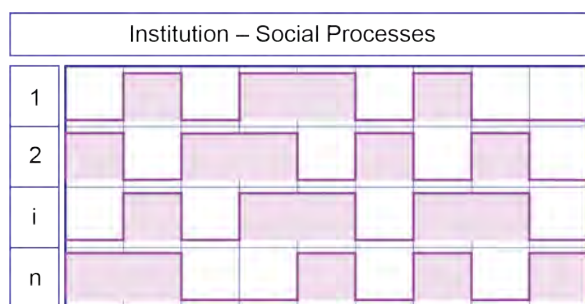


Рис. 3. Відношення «організація - процеси»

2) Social-процес - виконавці. Кожен social-процес забезпечується і виконується співробітниками, активність яких може бути асоційована з даними, знятими з інформаційного каналу (рис. 4). Це дає можливість інтегрально визначити активність кожного співробітника з метою оперативного прийняття актуаторних впливів, спрямованих на мінімізацію простоїв.

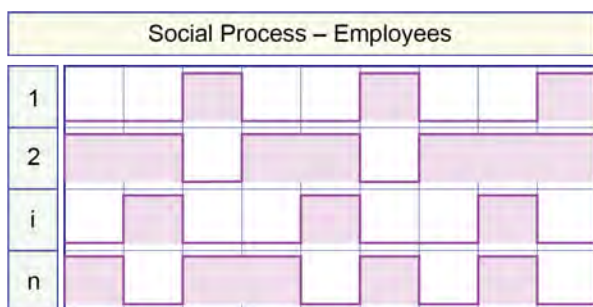


Рис. 4. Ставлення «процес - виконавці»

3) Виконавець - social-патерни. Кожен виконавець забезпечує і реалізує кілька social-функціональних або патернів, активність і стадії виконання яких можна виводити на монітор (рис. 5). Це дає можливість інтегрально визначити активність і спроможність співробітника для виконання

social-функціональності з метою оперативного прийняття актуаторних впливів, направлених на підвищення продуктивності.

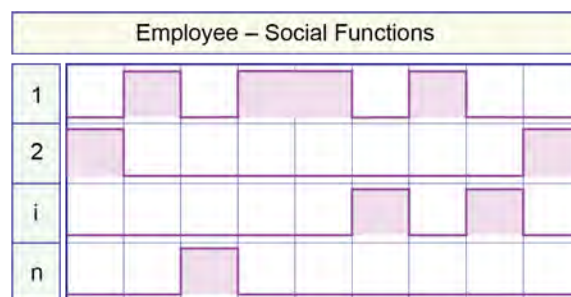


Рис. 5. Ставлення «виконавець - патерни»

4) Social-патерн - параметри (змінні). Кожен параметр своїми багатозначними значеннями формує social-функціональність або патерн, активність якого виводиться на монітор (рис. 6). Це дає можливість інтегрально визначити активність кожної змінної в часі для виконання social-функціональності з метою оперативного прийняття актуаторних впливів, направлених на підвищення продуктивності виконавця при реалізації патерна (рис. 7,8,9). Крім того, waveforms social-патерну дають можливість істотного зменшення інформації шляхом групування фреймів на основі фіксації однакових значень змінних.

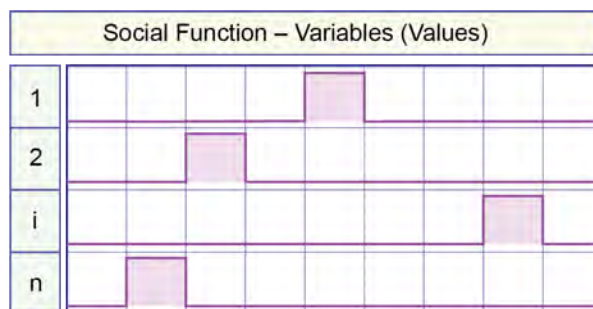


Рис. 6. Відношення «фрейм - унітарна кодування параметрів»

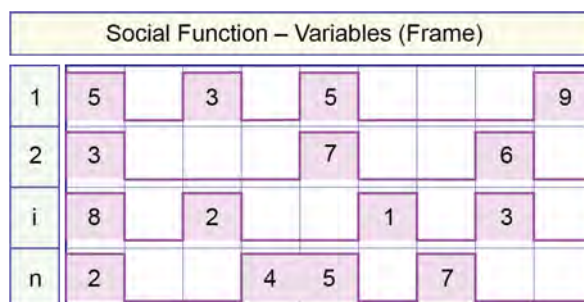


Рис. 7. Відношення «патерн - чисельні коди значень параметрів в фреймах»

Social Function – Variables (Frame)						
1		3	3	3	3	2 2
2		7	7	7	7	4 4
i						1 1
n						5 5

Рис. 8. Відношення «патерн - однакові параметри фреймів»

Social Function – Frames (Grouped)						
1		3		2		
2		7		4		
i				1		
n				5		
		4		2		2

Рис. 9. Відношення «патерн - згруповані фрейми»

Програмний модуль РАТТ розпізнавання патерну у вхідному потоці, довжиною р фреймів, формує на виході дві таблиці. Перша з них є результатом для відношення «патерн - потік»:

flow		pattern		$\mu(R,P)$	
No	UUID	ΔT	Name	GUID	
1	abc	5	Excel	a12	0
2	ert	7	Application	D34	0

Друга таблиця є додатковою і формує всі інші відношення для потоку, включаючи нефункціональний «сміття» і інші патерни:

flow		pattern		$\mu(R,P)$	
N	UUID	ΔT	Name	GUID	
3	fgh	2	Error	A56	0,75
4	hjt	6	User Name	g386	0,25

Об'єднання двох таблиць формує повну картину розпізнавання фреймів потоку. При цьому функція відмінності $\mu(R,P) = 0$, якщо фрейм належить патерну. В іншому випадку, коли $\mu(R,P) > 0$, робиться висновок про неналежність аналізованого фрейма заданому патерну. Параметр ΔT – позначає час виконання поточного фрейма потоку. Ефективність алгоритму розпізнавання патернів в потоці. Для підрахунку продуктивності вводяться

наступні позначення, що мають відображення в структурах даних, представлених на рис. 10. Параметри матриці патерну: n - число змінних, m - кількість значень в кожній змінній, що формує n * m стовпців матриці, які є шаблоном для запису n рядків або кроків патерну, що формують закінчену функціональність.

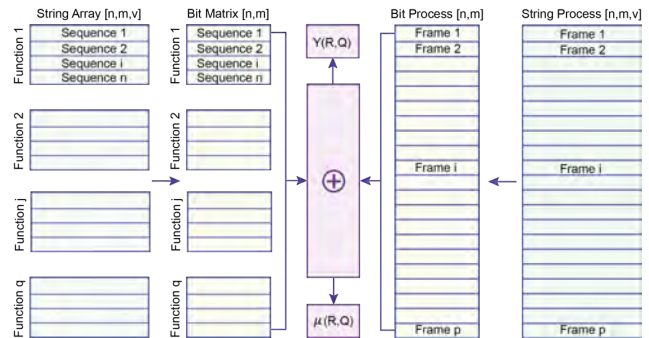


Рис. 10. Структури даних РАТТ-додатки

Обчислювальна складність розпізнавання патерну в базовому варіанті визначається наступною формулою:

$$Q(\text{basic}) = (n * m * k * q) \times (n * m * p),$$

яка використовує символічні структури даних і мультиплікативний алгоритм порівняння патерн-матриць і потоку, багаторазово повторюваний для фіксації збігу або відношення включення.

Обчислювальна складність нового алгоритму розпізнавання патернів в потоці включає надмірність, пов'язану з адитивними оцінками конвертації $(n * m * k * q + n * m * p)$ символічних значень змінних патернів і потоку в двійкові унітарні бітові коди, а також паралельне виконання хог-процедури над рядками матриць для фіксації п'яти варіантів відносин між черговим фреймом і патерном $k * q * p$. Таким чином, загальна оцінка обчислювальної складності має вигляд:

$$Q(\text{new}) = (n * m * k * q + n * m * p) + (k * q * p),$$

Виграш у швидкодії нового алгоритму виходить за рахунок разового перетворення даних із символів в біти, що дає можливість істотно зменшити обчислювальну складність процедури хог-відносин в бітовому форматі даних:

$$Q(\text{basic}) / Q(\text{new}) = [(nmkq) \times (nmp)] / (kqp) = nm * nm = (nm)**2.$$

Якщо не враховувати разові тимчасові витрати на унітарне кодування патернів і потоку, то продуктивність нового алгоритму розпізнавання патернів в потоці, в порівнянні з базовим зростає квадратично від розмірності сукупних значень даних одного фрейма.

Крім того, якщо число двійково-кодованих патернів більше одиниці, $q > 1$, тоді за рахунок їх багаторазового використання продуктивність хог-розпізнавання фреймів потоку збільшується в $q-1$ раз в порівнянні з базовим алгоритмом, де всі патерни схильні до аналізу в повному форматі символічних даних на кожному фреймі потоку. Таким чином, інтегрально, продуктивність програмного продукту РАТТ для розпізнавання патернів в потоці в порівнянні з базовим варіантом, збільшується в

$$Q(\text{basic})/Q(\text{new}) = (q-1) \times (nm)^{**2} \text{ раз,}$$

де q - число актуальних патернів, n - загальна кількість змінних для ідентифікації фрейма, m - інтегральне число значень всіх змінних. У сукупності добутку $m * n$ являє собою сукупний обсяг даних, представлених в фреймі потоку або патерну. В абсолютних оцінках швидкодію унітарного матричного процесора розпізнавання патернів в потоці на 2-4 порядки вище, ніж в базовому алгоритмі символічного аналізу. Отримані оцінки продуктивності необхідні для доказу можливості онлайн реалізації систем моніторингу та управління social-процесами.

Щоб піти від тривимірних масивів при аналізі взаємодії патернів з потоком, матриця значень параметрів скріншота або кроку розпускається в один двійковий вектор унітарних кодів, розмірністю $(n * m)$. Це дає можливість сформувати матрицю більш високого рівня, де сукупність кроків патерна формує повну функціональність, а кожен крок, який має свій початок, має формат рядка, над якою паралельно виконуються логічні операції.

Два вчених з Princeton University, [David Wentzlaff](#) и [Adi Fuchs](#), провели дослідження щодо продуктивності програмних і апаратних продуктів. Висновки, які вони доповіли на представницькому форумі [IEEE International Symposium on High-Performance Computer Architecture](#) наступні: 1) розробка спеціалізованої апаратури ASIC не вирішує проблем істотного підвищення швидкодії в порівнянні з програмними алгоритмами; 2) Найбільш істотними є архітектури, орієнтовані на мінімальне число логічних операцій, що застосовуються до пам'яті, де розміщуються великі дані; 3) Наступна ефективна за значимістю технологія ґрунтується на використанні вже раніше оброблених даних, записаних в пам'яті, для подальших обчислень, як альтернатива повторного час-витратного перерахунку. Таким чином, програмний модуль розпізнавання патернів виявився повністю відповідним двом технологіям обчислень, згаданими авторами, як найбільш ефективними в сучасній кіберкультурі комп'ютерингу [19-20].

Практичні висновки 1) Програмний модуль РАТТ призначений для розпізнавання патернів у вхідному потоці social-процесу компанії. 2) Швидкодія паралельного алгоритму обробки двійкових матриць, реалізованого в програмному продукті в 100 - 10000 разів вище, ніж у базового варіанту, що використовує символічні (текстові значення змінних) дані для розпізнавання патернів. 3) Використання програмного продукту дає можливість: визначати загальний час виконання аналізу взаємодії фреймів потоку даних з матрицями патернів, визначати якісну і кількісну оцінки приналежності фреймів до заданих патернів; визначати інтегральний час виконання патерну або функціональності співробітником компанії; визначати загальний час виконання нефункціональних обов'язків співробітником; визначати ефективність роботи співробітника за робочий день. 4) Подальші роботи, пов'язані з програмуванням, будуть спрямовані на: автоматичну побудову довічних патерн-матриць; формування зручного інтерфейсу візуалізації social-процесу в режимі онлайн; побудову візуальних точних (нестатистичних) діаграм ефективності роботи співробітників і компанії.

5. Архітектури розпізнавання патернів і фігур

Однією з основних проблем є розпізнавання компонентів фрейма з метою їх логічного інтегрування в семантичну структуру social-патерну або потоку для їх подальшого аналізу за допомогою програми РАТТ, що використовує матричні бітові структури даних.

Структура процесу представлена наступним рис. 11, який показує перетворення компонентів фреймів в регулярні матричні структури даних потоку або патерна.

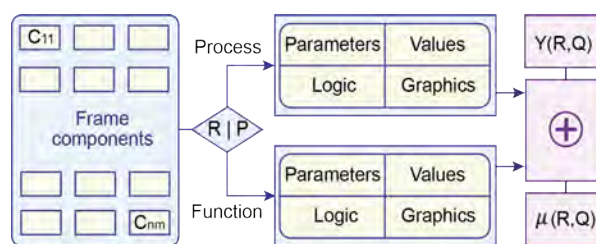


Рис. 11. Процес розпізнавання компонентів фрейму

Завдання: Розробити і закодувати модуль визначення подібності або подоби між двома інформаційними фреймами. Мета - мінімізувати обсяг даних (паспорт або ідентифікатор), декодувати з фреймів, для подальшого аналізу потоку recording R щодо еталонного образу E, що дає можливість отримати функції приладдя $m(R_i, E)$ і структурний вектор відмінності $Y(R_i, E)$ двох фреймів.

Рішення завдання виконується шляхом програмування алгоритму розпізнавання поточного фрейму R щодо еталона E, що має квадратичну обчислювальну складність. Алгоритм містить три обчислювальні процедури, які використовують архітектуру, представлену на рис. 12.

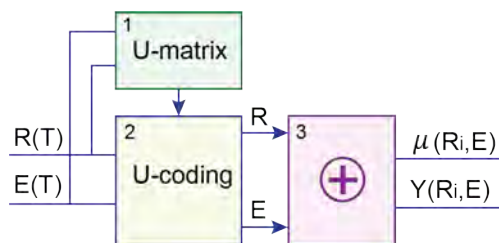


Рис. 12. Архітектура процесінгу для визначення подібності фреймів

1) Препроцесінг-фаза, коли виконується аналіз усіх фігур в кадрі з текстами з метою унітарного кодування значень. Потім виконується аналіз чергового фрейму для пошуку таких фігур і значень, які відсутні в раніше оброблених фреймах. В результаті виходить U-вектор текстових значень змінних або фігур для унітарної кодування всіх фреймів.

2) Процесінг унітарного кодування для отримання двійкових образів фреймів в універсальному форматі U-вектора унікальних текстових значень. В результаті виходить двійкова матриця унітарного кодування всіх фреймів і еталонного образу, що підлягають аналізу по визначенню ступеня подібності або приналежності кожного фрейму до вибраного еталону.

3) Заключний хог-аналіз двійкової матриці фреймів щодо обраного двійкового еталон-фрейма з метою виставляння кожному рядку матриці фреймів функції приналежності до еталону, визначеної в інтервалі (0,1). Крім того, функціональний модуль, реалізований програмно, дає можливість отримувати структурну оцінку функції приналежності, що дозволяє коригувати вихідні вектори або фрейми, в разі необхідності. Подоба двох фігур. Метрика для визначення подібності двох фігур використовує обчислення відстані між двома точками в декартовому просторі шляхом застосування такої формули:

$$L = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}.$$

При цьому, для отримання наведеної оцінки відмінності прямокутників на перше місце ставиться відношення проєкцій відстані на осі абсцис і ординат в декартовій системі координат до скалярної оцінки даного відстані:

$$\mu(P) = \frac{1}{2} \left[\frac{X_2 - X_1}{L} + \frac{Y_2 - Y_1}{L} \right],$$

де в чисельнику фігурують катети, наведені до загальної гіпотенузи, яка є відстанню між двома однойменними точками-координатами двох різних прямокутників.

З огляду, що в прямокутнику число реперних точок, які формують фігуру, всього дві, то для визначення подібності двох прямокутників, необхідно обчислити дві відстані між двома відповідними точками-координатами (**top-left** and **bottom-right**), які формують знаменник нормованих відносин для отримання інтервальної оцінки подібності двох фігур в двовимірному просторі:

$$L^t = \sqrt{(X_2^t - X_1^t)^2 + (Y_2^t - Y_1^t)^2},$$

$$L^b = \sqrt{(X_2^b - X_1^b)^2 + (Y_2^b - Y_1^b)^2}.$$

Таким чином інтегрально, сума наведених відмінностей між однойменними координатами (**top-left** and **bottom-right**) двох прямокутників на сусідніх у часі фреймах формує нормовану в інтервалі (0,1) оцінку **подібності** фігур в декартовому просторі:

$$\mu(R) = 1 - \frac{1}{2} \left\{ \left[\frac{X_2^t - X_1^t}{L^t} + \frac{Y_2^t - Y_1^t}{L^t} \right] + \frac{1}{2} \left[\frac{X_2^b - X_1^b}{L^b} + \frac{Y_2^b - Y_1^b}{L^b} \right] \right\}$$

Інша інтерпретація оцінки **подібності** має такий вигляд:

$$\mu(D) = 1 - \frac{1}{4} \left[\frac{X_2^t - X_1^t}{L^t} + \frac{Y_2^t - Y_1^t}{L^t} + \frac{X_2^b - X_1^b}{L^b} + \frac{Y_2^b - Y_1^b}{L^b} \right].$$

Тут береться модуль різниці між однойменними координатами X, Y двох прямокутників. На практиці це означає, що оцінка формується шляхом знаходження середнього значення суми відносин катетів до гіпотенузи трикутника, побудованої на двох однойменних точках, які підлягають порівнянню. При цьому число таких прямокутних трикутників дорівнює кількості реперних (істотних) точок в постатях, які підлягають порівнянню. Наприклад, для прямокутника достатньо двох таких точок, для трикутника - трьох, для неправильного чотирикутника - чотирьох, рис. 13.

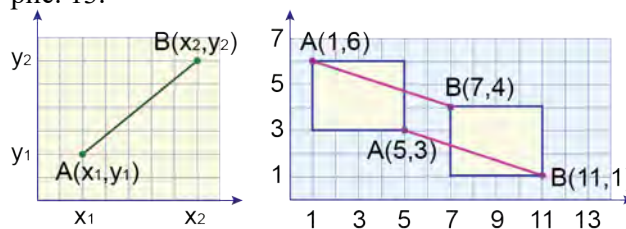


Рис. 13. Обчислення відстані між об'єктами

Гідність наведеної універсальної формули-метрики визначення середнього значення нормованої відстані полягає в її використанні для обчислення скалярної оцінки (у відсотках!) подібності двох будь-яких об'єктів, для яких задані реперні точки. Другим завданням є створення метрики розпізнавання схожих прямокутників з текстовими блоками. В даному випадку відсутня прив'язка геометричної фігури до позиції в фреймі. Важливими компонентами при цьому може виступати площа прямокутника, що відповідає за розмір, і його діагональ, яка характеризує структуру фігури по довжині або висоті. Однак більш простою метрикою для вимірювання прямокутника є ширина і висота, щодо яких всі інші характеристики фігури є похідними. Використання даної пари дає можливість порівнювати подобу двох геометричних фігур навіть в змінених масштабах з метою визначення скалярної нормованої оцінки в інтервалі від 0 до 1. Наступна формула задає метрику для порівняння двох фігур:

$$Q=(1/2)[X1/X2 + Y1/Y2].$$

Другий варіант оцінки має вигляд:

$$Q=(1/2)[(X1-X2)/L + (Y1-Y2)/L],$$

де $L = (g1^{**2}+g2^{**2})^{**}(1/2)$.

Сутність метрики подібності для двох прямокутників полягає у визначенні загальної площі шляхом перетину при їх накладанні один на одного при суміщенні лівими верхніми кутами або точками.

```
def intersection(self, box):
    x1 = max(self.x1, box.x1)
    x2 = min(self.x2, box.x2)
    y1 = max(self.y1, box.y1)
    y2 = min(self.y2, box.y2)
    if x1 < x2 and y1 < y2:
        return Box(x1, y1, x2, y2)
    return Box(0,0,0,0)
```

```
def union(self, box):
    x1 = min(self.x1, box.x1)
    x2 = max(self.x2, box.x2)
    y1 = min(self.y1, box.y1)
    y2 = max(self.y2, box.y2)
    return Box(x1, y1, x2, y2)
```

```
def iou(self, box):
    return float(self.intersection(box).area) / (float(self.union(box).area) + 0.0001)
```

Обчислення метрики подібності фреймів має наступні пункти: 1) Вставити в додаток функціональний Java-модуль для визначення оцінки

подібності текстових фрагментів з подальшою імплементацією її в інтегральну метрику подібності фреймів. 2) Створити формулу-метрику для нормованого оцінювання подібності двох будь-яких фреймів, включаючи нерівне число геометричних компонентів, фігур і текстів. 3) Формула-метрика повинна також враховувати: систематичне і локальне зміщення фігур по осях абсцис і ординат; систематичне і локальне розтягування фігур і текстів на фреймі по ширині і висоті. 4) Визначення взаємозалежної структури реперних точок фігур фрейму для формування системної оцінки подібності фреймів, як цілісних образів. Нижче представлений рис. 14 для чотирьох можливих взаємодій прямокутників, інваріантних до позиції в фреймі, а також формули, які дають можливість визначати подібність прямокутників r and R на сусідніх фреймах. Перша і друга формули ідентичні за функціональними можливостями і відносяться до лівого верхнього найскладнішого перетину.

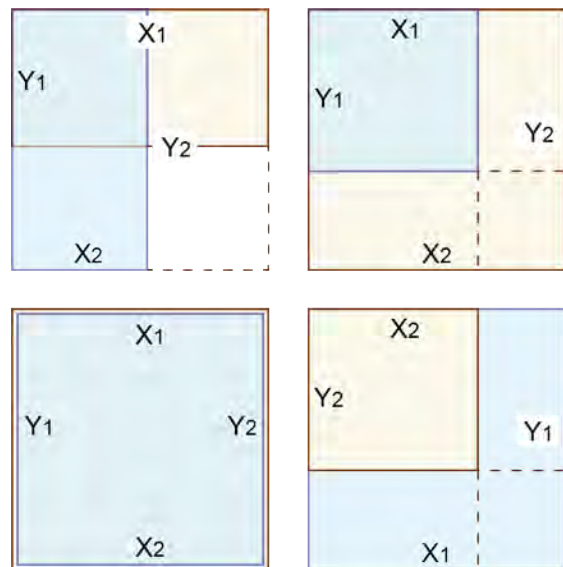


Рис. 14. Варіанти взаємодії фігур

Таким чином, інтегральна метрика для вимірювання двох геометричних фігур на сусідніх скріншотах оперує трьома нормованими параметрами, які в сукупності зі значеннями текстових параметрів здатні визначити скалярну оцінку ідентичності ($Q=1$) або подібності прямокутників в $(0,1)$ -інтервалі:

1) **Координата.** На основі існуючих координат (X_i, Y_i) двох реперних точок двох фігур визначається *близькість або схожість* між ними у вигляді скалярної $(0,1)$ -норми.

$$\mu(D) = 1 - \frac{1}{4} \left[\frac{X_2^t - X_1^t}{L^t} + \frac{Y_2^t - Y_1^t}{L^t} + \frac{X_2^b - X_1^b}{L^b} + \frac{Y_2^b - Y_1^b}{L^b} \right]$$

2) **Орієнтація.** Положення на площині прямокутників в декартовій системі координат має дві орієнтації: прямокутник лежить або стоїть, що істотно впливає на подібність фігур, яке враховує орієнтацію фігур в двовимірному просторі у вигляді скалярної (0,1)-норми *близькості* по орієнтації, де X_i, Y_i - довжини сторін двох прямокутників. Далі наведені три форми метрики оцінювання подібності двох прямокутників, які оперують відносинами сторін прямокутників, що дає можливість визначити наскільки близько дані фігури відповідають один одному:

$$\mu(S) = \min\left(\frac{X_1}{Y_1}, \frac{X_2}{Y_2}\right) / \max\left(\frac{X_1}{Y_1}, \frac{X_2}{Y_2}\right)$$

$$\mu(S) = \frac{\min\left(\frac{X_1}{Y_1}, \frac{X_2}{Y_2}\right)}{\max\left(\frac{X_1}{Y_1}, \frac{X_2}{Y_2}\right)}$$

$$\mu(S) = \frac{\min[(X_1 / Y_1), (X_2 / Y_2)]}{\max[(X_1 / Y_1), (X_2 / Y_2)]}$$

Крім того, наведені вище формули інваріантні до масштабування двох фігур, що дає можливість оцінювати схожість дрібних і великих прямокутників шляхом ділення відносин довжин сторін прямокутників. Якщо пропорції дотримуються, то частка від ділення відносин формує одиничну (максимальну) ступінь подібності розглянутих фігур.

3) **Суперпозиція.** Визначається шляхом накладення площ двох фігур, коли їх ліві верхні координати поєднуються один з одним, що дає можливість визначити якість перетину площ або близькості двох фігур у вигляді скалярної (0,1)-норми, де X_i, Y_i - довжини сторін фігури.

$$k = \begin{cases} 0 \Leftrightarrow (X_1, Y_1) \geq (X_2, Y_2) \vee (X_1, Y_1) \leq (X_2, Y_2); \\ 1 \Leftrightarrow (X_1 > X_2) \wedge (Y_1 < Y_2) \vee (X_1 < X_2) \wedge (Y_1 > Y_2); \end{cases}$$

$$\mu(R) = \frac{\min(X_1, X_2) \min(Y_1, Y_2)}{\max(X_1, X_2) \max(Y_1, Y_2) - k [|X_1 - X_2| |Y_1 - Y_2|]}$$

Якщо точка з максимальними координатами по $X = X_1$ і $Y = Y_2$ не належить прямокутникам, то такі фігури взаємодіють за правилом непорожнього перетину. Останні формули обслуговують всі варіанти відносин збігу щодо запропонованих фігур стріншота, включаючи непорожній перетин між двома прямокутниками і **масштабовані** подібності з відношенням включення. Масштабована подібність геометричних прямокутників визначається накладенням двох фігур один на одного, коли одна фігура повністю міститься в іншій і виконується пропорція співвідношення сторін, що в сукупності визначає подібність прямокутників.

Середнє значення трьох наведених норм близькості формує оцінку ідентичності ($Q = 1$), подібності або схожості двох фігур, що надає експерту-аналітику аргументи для розпізнавання і класифікації фреймів:

$$Q = \frac{1}{3} [\mu(D) + \mu(S) + \mu(R)],$$

$$Q = \frac{1}{5} [\mu(D) + \mu(S) + \mu(R) + \mu(T) + \mu(C)].$$

Далі необхідно визначити метрику порівняння двох фреймів, яка повинна враховувати оцінки подібності текстових фрагментів, включаючи нерівне число геометричних компонентів, фігур і текстів, систематичне і / або локальне зміщення, розтягування і / або стиснення фігур по осях абсцис і ординат.

1) Для цього спочатку визначається кількість компонентів-фігур n на зразковому фреймі, а також їх число m на поточному фреймі, маючи на увазі пари координат прямокутників. 2) Для побудови двійкового вектора структурної організації фігур на зразок-фреймі всі прямокутники нумеруються в порядку: зліва-направо, зверху-вниз. Далі всі фігури на аналізованому фреймі нумеруються вже по метриці еталона. 3) Можливі варіанти взаємодії компонентів на фреймах, після вимірювання подібності всіх текстів і фігур, визначаються формулами метричного оцінювання.

Перший вираз визначає функцію приналежності або ступінь подібності між парою прямокутників в фреймах, яка залежить від:

$$Q_i = \frac{1}{5} [\mu_i(D) + \mu_i(S) + \mu_i(R) + \mu_i(T) + \mu_i(C)].$$

координат, які формують відстань між фігурами; становища фігур в двовимірному просторі; співвідношення загальних площ при їх суперпозиції; текстуальної подібності або відстані між прямокутниками, отриманої за допомогою Java-функції; кодової відстані між двійковими векторами, які формують структурну оцінку присутності фігур на фреймах, де нульове значення координати свідчить про відсутність прямокутника на вже згаданому фреймі. Кодова відстань між структурними векторами визначається за допомогою рівняння:

$$\mu_i(C) = \frac{1}{m} \sum_{i=1}^n C_i = 1 \leftarrow C_e \oplus C_i = \mu_i(C).$$

Другий вираз задає відносини між компонентами (тексти-фігури) еталонного і аналізованого фрейма, виражені за допомогою апарату теорії множин, які формують відносини приналежності (дві перших формули), рівності, порожнього і непорожнього перетину:

$$\begin{aligned}
R_e \subseteq R_i &\leftarrow R_e \cap R_i = R_e; \\
R_i \subseteq R_e &\leftarrow R_e \cap R_i = R_i; \\
R_e = R_i &\leftarrow R_e \cap R_i = R_e = R_i; \\
R_e \not\subseteq R_i &\leftarrow R_e \cap R_i = \emptyset; \\
\exists R_i \subset R_e &\leftarrow R_e \cap R_i \neq \emptyset.
\end{aligned}$$

Інтегральний критерій якості взаємодії між фреймами в цілях визначення оцінки подібності сумарно інтегрує функції приладдя в метриці текстів-фігур:

$$\mu(R_e, R_i) = \frac{1}{n} \sum_{i=1}^n Q_i.$$

Розгорнутий вираз для обчислення ступеня подібності скріншотів являє собою середнє значення оцінок для n функцій, які формують ступінь близькості всіх компонентів на парі фреймів:

$$\mu(R_e, R_i) = \frac{1}{n} \sum_{i=1}^n \frac{1}{5} [\mu_i(D) + \mu_i(S) + \mu_i(R) + \mu_i(T) + \mu_i(C)].$$

Сумарно, метрика для визначення подібності фреймів на основі використання п'яти функцій приладдя для еталонного і поточного фрейму має вигляд:

$$L^t = \sqrt{(X_2^t - X_1^t)^2 + (Y_2^t - Y_1^t)^2},$$

$$L^b = \sqrt{(X_2^b - X_1^b)^2 + (Y_2^b - Y_1^b)^2}.$$

$$\mu(D) = 1 - \frac{1}{4} \left[\frac{X_2^t - X_1^t}{L^t} + \frac{Y_2^t - Y_1^t}{L^t} + \frac{X_2^b - X_1^b}{L^b} + \frac{Y_2^b - Y_1^b}{L^b} \right]$$

$$\mu(S) = \frac{\min\left(\frac{X_1}{Y_1}, \frac{X_2}{Y_2}\right)}{\max\left(\frac{X_1}{Y_1}, \frac{X_2}{Y_2}\right)}$$

$$k = \begin{cases} 0 & \Leftrightarrow (X_1, Y_1) \geq (X_2, Y_2) \vee (X_1, Y_1) \leq (X_2, Y_2); \\ 1 & \Leftrightarrow (X_1 > X_2) \wedge (Y_1 < Y_2) \vee (X_1 < X_2) \wedge (Y_1 > Y_2); \end{cases}$$

$$\mu(R) = \frac{\min(X_1, X_2) \min(Y_1, Y_2)}{\max(X_1, X_2) \max(Y_1, Y_2) - k [|X_1 - X_2| |Y_1 - Y_2|]}.$$

$$\mu_i(C) = \frac{1}{m} \sum_{i=1}^n C_i = 1 \leftarrow C_e \oplus C_i = \mu_i(C)$$

$$Q_i = \frac{1}{5} [\mu_i(D) + \mu_i(S) + \mu_i(R) + \mu_i(T) + \mu_i(C)]$$

$$\mu(R_e, R_i) = \frac{1}{n} \sum_{i=1}^n \frac{1}{5} [\mu_i(D) + \mu_i(S) + \mu_i(R) + \mu_i(T) + \mu_i(C)]$$

Якщо кількість прямокутників відмінна на фреймах, то метрика працює за сценарієм включення в якості підмножини фігур. Тут обслуговуються дві відносини включення, а також порожнього або непорожнього перетину прямокутників. Звичайно, що відношення рівності в даному випадку не може з'явитися. Результат виходить шляхом підрахунку одиничних значень, виділених на за-

гальне число координат-фігур в векторі структурної взаємодії фігур і текстів двох фреймів. Однак в загальному випадку, структурна оцінка обслуговує всі п'ять видів взаємодії компонентів на фреймах, включаючи рівне число текстів-фігур. Для визначення взаємодії двох фреймів будується матриця оцінок подібності, розмірністю pxm, де p - число текст-фігур (рядків) на першому фреймі, m - (стовпців) на другому. Таким чином, матриця подібності формує сильне відношення між компонентами двох фреймів, що дорівнює декартовому добутку двох множин. На рис. 15 представлені дводольні: сильний граф, а також два графа, які формують можливі відносини між компонентами фреймів, де кожен з них може бути шаблоном.

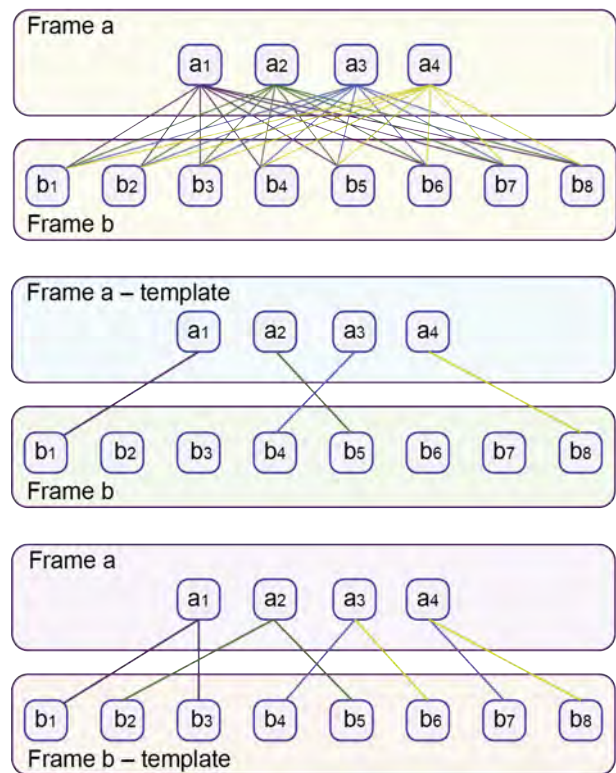


Рис. 15. Графові моделі відношення компонентів на двох фреймах

Визначення подібності двох фреймів зводиться до побудови або пошуку взаємно однозначної відповідності на повному дводольному графі відносини двох множин компонентів, розташованих на двох фреймах:

$\mu(a,b)$	1	2	3	4	5	6	7	8	$\max(j)$
1	1,0	0,7	0,8	0,9	0,3	0,2	0,1	0,2	1(1,0)
2	0,5	0,1	0,4	0,6	0,7	0,9	0,2	0,3	6(0,9)
3	0,1	0,4	0,2	0,9	0,5	0,6	0,5	0,2	4(0,9)
4	0,3	0,4	0,5	0,1	0,7	0,2	0,4	0,3	5(0,7)
$\max(i)$	1(1,0)	1(0,7)	1(0,8)	1(0,9)	2(0,7)	2(0,9)	3(0,5)	2(0,3)	$\mu(a,b) = 0,875$ $\mu(b,a) = 0,725$

Матриця подібності є достовірною моделлю повних відносин між компонентами (фігури-тексти) master-slave фреймів для пошуку і класифікації фреймів. З даної матриці можна витягти дві оцінки подібності фреймів, залежно від порядку відносини: ($a \subseteq b$, $a \supseteq b$). Координата матриці визначена скалярною оцінкою, що представляє середнє значення чотирьох (п'яти) параметрів, кожен з яких нормований в інтервалі від 0 до 1.

$$\mu_{ij} = \frac{1}{5} [\mu_{ij}(D) + \mu_{ij}(S) + \mu_{ij}(R) + \mu_{ij}(T) + \mu_{ij}(C)],$$

$$\mu_{ij} = \frac{1}{5} [\mu_{ij}(D) + \mu_{ij}(S) + \mu_{ij}(R) + \mu_{ij}(T)].$$

Оцінка подібності двох фреймів обчислюється формулою, яка визначає середнє значення максимумів, знайдених для кожної з n рядків матриці. Дана оцінка формує функцію приналежності $a \subseteq b$, яка не має властивості симетричності:

$$\mu(a, b) = \frac{1}{n} \sum_{j=1, m}^n \max \mu_{ij}.$$

Тому оцінка подібності двох фреймів при зміні порядку щодо включення $b \subseteq a$ обчислюється подібною формулою, але яка визначає, як правило, інше середнє значення максимумів, знайдених для кожного з m стовпців:

$$\mu(b, a) = \frac{1}{m} \sum_{j=1, n}^m \max \mu_{ij}.$$

З п'яти можливих теоретико-множинних взаємодій між двома множинами два з них, які формують відношення включення ($a \subseteq b$, $a \supseteq b$), не є симетричними. Тому в наведеній вище таблиці максимальна оцінка подібності за координатами рядків $a \subseteq b$ дорівнює 0,875, а за координатами стовпців $b \subseteq a = 0,725$. Це означає, що ступінь подібності істотно залежить від співвідношення числа компонентів в зразковому і вже згаданому фреймі: чим менше компонентів в зразку, тим вище може бути оцінка подібності, завдяки більшій вибірці, що надається рядками матриці, що має m стовпців і n рядків, де $m > n$. Важливо відзначити, що визначається тільки одна фігура на вже згаданому фреймі, подібна фігурі на зразковому фреймі, що формує взаємно-однозначну відповідність між прямокутниками і текстами на двох фреймах. Дана фігура вибирається за критерієм максимальної оцінки подібності, отриманої в результаті порівняння еталонного прямокутника з усіма аналогами на вже згаданому фреймі.

Асиметрія відношення між фреймами, що задається матрицею подібності

$$a \subseteq b \neq b \subseteq a \approx \mu(a, b) \neq \mu(b, a) \approx \mu(i, j) \neq \mu(j, i),$$

виражається пошуком максимумів по рядках або стовпцях, які, в загальному випадку, не рівні один одному

$$\max_{i=1, n} \mu_{ij} \neq \max_{j=1, m} \mu_{ij}.$$

При аналізі матриці подібності двох фреймів, один з яких – master або зразковий, а другий – slave, який є поточним аналізованим фреймом з потоку великих даних, можливі випадки: 1) Одному компоненту на master-фреймі відповідає кілька подібних об'єктів на slave-фреймі, які мають однакові оцінки (ін'єктивна відповідність). 2) Одному компоненту на slave-фреймі відповідає кілька подібних об'єктів на master-фреймі, які мають однакові оцінки (функціональна відповідність). 3) Одному компоненту на master-фреймі відповідає тільки один подібний об'єкт на slave-фреймі (взаємно-однозначна відповідність). 4) Компоненту на master-фреймі не знайдений подібний об'єкт на slave-фреймі. 5) Компоненту на slave-фреймі не знайдений подібний об'єкт на master-фреймі. 6) Компоненти на двох фреймах взаємодіють за схемою, що включає всі згадані вище види відповідностей.

Таким чином, інтегральний критерій подібності фреймів повинен враховувати всі види згаданих відповідностей, які об'єднуються в такій формулі:

$$\mu(a, b) = \frac{1}{n} \sum_{j=1, m}^n \max \mu_{ij}, \quad \mu(b, a) = \frac{1}{m} \sum_{j=1, n}^m \max \mu_{ij}.$$

Наступна таблиця ілюструє всі можливі види відносин між компонентами скріншотів, які також показані в формі дводольних графів на рис. 16.

$\mu(a, b)$	1	2	3	4	5	6	7	8	$\max(j)$
1	1,0	0,7	0,8	0,9	0,3	0,2	0,1	0,2	$\mu(1,1) = 1,0$
2	0,5	0,1	0,4	0,6	0,7	0,9	0,2	0,3	$\mu(2,6) = 0,9$
3	0,1	0,4	0,2	0,9	0,5	0,6	0,5	0,2	$\mu(3,4) = 0,9$
4	0,3	0,4	0,5	0,1	0,7	0,2	0,4	0,3	$\mu(4,5) = 0,7$
$\max(i)$	$\mu(1,1) = 1,0$	$\mu(2,1) = 0,7$	$\mu(3,1) = 0,8$	$\mu(4, \{1,3\}) = 0,9$	$\mu(5, \{2,4\}) = 0,7$	$\mu(6,2) = 0,9$	$\mu(7,3) = 0,5$	$\mu(8, \{2,4\}) = 0,3$	$\mu(a, b) = 0,875$ $\mu(b, a) = 0,725$

Проблема аналізу матриці подібності полягає в неоднозначності типу відповідності від більшої множини компонентів до меншої, яка пов'язана, в даному випадку, з відношенням між великим числом стовпців і меншою кількістю рядків. Даний факт зафіксований в нижньому рядку таблиці. Щоб усунути таку невизначеність необхідно побудувати зустрічну відповідність, з метою перетину двох отриманих множин, яка прибирає всі компоненти, які не є загальними для двох відповідностей.

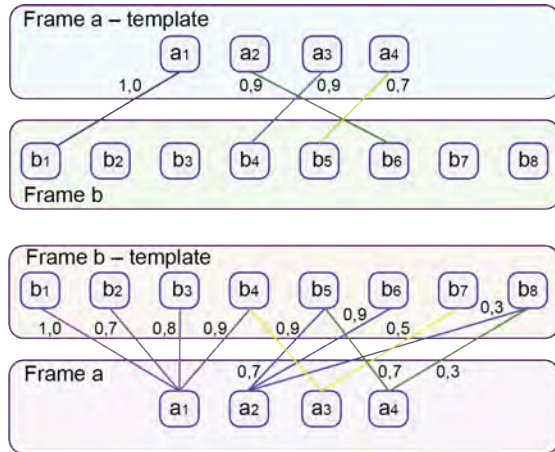


Рис. 16. Дводольні графи взаємодії компонентів фреймів

Структура програмного додатку для аналізу великих даних з метою синтезу матриці подібності та подальшого визначення скалярної оцінки, як функції приналежності або подібності двох фреймів, що диференціюються на master and slave, представлена на рис. 17.

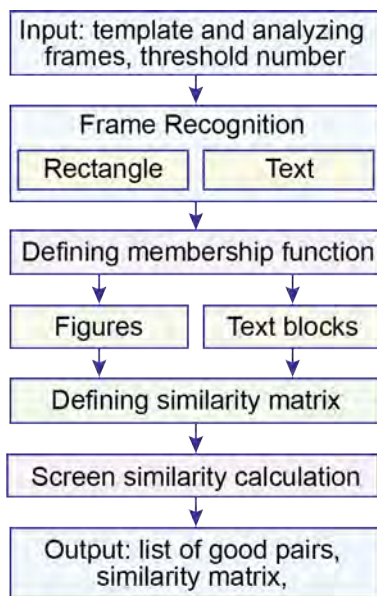


Рис. 17. Структура моделей програмного додатку для визначення подібності між фреймами

6. Метрика подібності пари фреймів

Формула для метричного оцінювання подібності геометричних фігур - прямокутників (з можливими текстами) включає чотири компоненти подібності - функції приладдя двох фреймів (по координаті, орієнтації, площі і тексту):

$$\mu_{ij}^r = \frac{1}{4} [\mu_{ij}(D_r) + \mu_{ij}(S_r) + \mu_{ij}(R_r) + \mu_{ij}(T_r)].$$

Дана формула використовується для побудови матриці подібності, кожна координата якої вклю-

чає інтегральний критерій для кожної пари прямокутників на двох аналізованих скріншотах. Аналіз матриці подібності фреймів за такими формулами:

$$\mu^r(a, b) = \frac{1}{n} \sum_{i=1}^n \max_{j=1, m} \mu_{ij}^r, \quad \mu^r(b, a) = \frac{1}{m} \sum_{j=1}^m \max_{i=1, n} \mu_{ij}^r.$$

дає можливість сформувані інтегральні оцінки подібності двох фреймів в метриці геометричних фігур - прямокутників. Оскільки одній фігурі на еталонному (master) фреймі може відповідати кілька компонентів з однаковими оцінками на вже згадуваному (slave) фреймі то перша формула створює необхідні, а друга формула - достатні умови подібності двох фігур на фреймі, яка уточнює, шляхом зустрічного вибору максимальної оцінки відповідності slave-master, справжню пару подібних компонентів, якщо відношення master-slave мало кілька однакових максимальних оцінок. Якщо компоненти на двох фреймах мають максимальні оцінки в двох відносинах (master-slave і навпаки), то фігури подібні один одному за умовою необхідності і достатності. Інша інтерпретація: необхідність і достатність подібності фреймів визначається перетином фігур, отриманих шляхом їх master-slave і slave-master аналізу. Наприклад, для останньої таблиці, з метою визначення подібності фреймів, виконується теоретико-множинна операція перетину:

$$\begin{aligned} MS \cap SM &= \\ &= (1-1), (2-6), (3-4), (4-5) \cap SM = (1-1), (2-1), \\ &(3-1), (4-1,3), (5-1,4), (6-2), (7-3), (8-2,4) = \\ &= (1-1), (2-6), (3-4), (4-5). \end{aligned}$$

Такий перетин особливо необхідний, коли master-slave відповідність визначена на різних потужностях множин, де $n > m$, наприклад, $|a| > |b|$ або $|MS| > |SM| = 8 > 4$. В цьому випадку, перша відповідність може мати неоднозначності по фігурах меншої множини, які мають бути довизначені шляхом побудови зворотньої відповідності. З метою мінімізації витрат часу можна попередньо визначати потужності множин $|MS|$ і $|SM|$ ($|a|$ і $|b|$), а потім синтезувати тільки одну відповідність від меншої множини до більшої:

$$\mu^r = \begin{cases} \mu^r(a, b) = \frac{1}{n} \sum_{i=1}^n \max_{j=1, m} \mu_{ij}^r \leftarrow n < m; \\ \mu^r(b, a) = \frac{1}{m} \sum_{j=1}^m \max_{i=1, n} \mu_{ij}^r \leftarrow n < m. \end{cases}$$

$$\mu^r = \begin{cases} \mu^r(a, b) = \frac{1}{n} \sum_{i=1}^n \max_{j=1, m} \mu_{ij}^r \leftarrow |a| < |b|; \\ \mu^r(b, a) = \frac{1}{m} \sum_{j=1}^m \max_{i=1, n} \mu_{ij}^r \leftarrow |b| < |a|. \end{cases}$$

Дана формула обчислює середнє значення між максимумами метричної подібності всіх компонентів фреймів. Якщо число фігур на master-фреймі менше або дорівнює, ніж на slave-фреймі, ніяких додаткових дій не потрібно робити. Якщо число фігур на master-фреймі більше, ніж на slave-фреймі, то інтегральна оцінка за фактом формується середнім значенням максимумів, яке може бути менше за рахунок меншої вибірки значень, також за рахунок більшого знаменника - кількості master-компонентів. Таким чином, якщо розглядати два фрейми, то оцінка, отримана з боку master-фрейма з великою кількістю фігур, завжди буде об'єктивно меншою, ніж з боку master-фрейма з меншим числом компонентів. Інакше, математично тут фігурують два види відповідностей, які не симетричні і не рівні один одному по визначенню.

Обробка текстфілдів на master-slave (MS) фреймах здійснюється аналогічним чином (master-фрейм один, а slave-ів може бути будь-яке кінцеве число). При цьому враховуються три параметра: координати текстів, кількість рядків і власне розпізнаний текст:

$$\mu_{ij}^t = \frac{1}{3} [\mu_{ij}(D_t) + \mu_{ij}(S_t) + \mu_{ij}(T_t)].$$

Далі виконується підрахунок схожості всіх текстів на MS-фреймах шляхом побудови матриці подібності з подальшим винесенням середньої оцінки:

$$\mu^t(a,b) = \frac{1}{n} \sum_{i=1}^n \max_{j=1,m} \mu_{ij}^t, \quad \mu^t(b,a) = \frac{1}{m} \sum_{j=1}^m \max_{i=1,n} \mu_{ij}^t.$$

Обробка інших фігур на master-slave (MS) фреймах здійснюється подібно процедурі аналізу прямокутників. При цьому враховуються три параметра на фреймах: координати фігури, їх взаємодія по площах і статус: «активні-пасивні»:

$$\mu_{ij}^c = \frac{1}{3} [\mu_{ij}(D_c) + \mu_{ij}(R_c) + \mu_{ij}(S_a)].$$

Далі виконується підрахунок схожості всіх фігур на MS-фреймах шляхом побудови матриці подібності з подальшим винесенням середньої оцінки:

$$\mu^c(a,b) = \frac{1}{n} \sum_{i=1}^n \max_{j=1,m} \mu_{ij}^c, \quad \mu^c(b,a) = \frac{1}{m} \sum_{j=1}^m \max_{i=1,n} \mu_{ij}^c.$$

Після роздільного виконання аналізу трьох класів компонентів на MS- фреймах: прямокутників, текстів і фігур обчислюється інтегральна оцінка подібності MS-фреймів шляхом знаходження середнього значення від суми оцінок подібності прямокутників, текстів і фігур:

$$\mu^\Sigma = \frac{1}{3} [\mu^r + \mu^t + \mu^c].$$

З урахуванням наперед заданої верхньої межі подібності фреймів, наприклад, рівній 0,95, робиться висновок про подібність поточного або аналізованого slave-фрейма першому master-кадру.

Алгоритм обчислення метрики для визначення подібності текстів використовує відстань (Levenshtein Distance) між словами, пропозиціями і текстами. Таку відстань враховує мінімальну кількість букв, необхідних для трансформування текстового фрагмента X в Y. При цьому виконуються елементарні процедури: вставка, видалення або заміна символу в рядку.

Таким чином, можна зробити наступні висновки: 1) Нерівнозначності множин компонентів на двох фреймах призводить до асиметрії відносин в переважній більшості випадків вимірювання, за винятком випадку, коли число компонентів збігається на двох фреймах: вони можуть перетинатися - мати загальну частину; не перетинатися - не мати нічого спільного один з одним; бути ідентичними або збігатися - повністю перетинатися по всіх елементах, які є однаковими. 2) Варіант, коли потужність безлічі компонентів master-фрейма менше, ніж на slave-фреймі, означає - для кожного елемента першого фрейму існує більше варіантів порівняння, ніж навпаки, коли потужність компонентів master більше, ніж на slave. Природно, даний факт є логічним доказом того, що оцінка відповідності, при $m < s$ буде практично завжди більшою, ніж для випадку $m > s$. 3) Більш того, обидві обчислювані оцінки в сукупності формують критерій якості відносин між двома фреймами, а саме: якщо $q(m, s) > q(s, m)$, то m is contained in s , в іншому випадку, якщо $q(m, s) < q(s, m)$, то s is contained in m . Це дуже важливо і універсально працює для визначення подібності будь-яких об'єктів. Для обчислення системної оцінки подібності між фреймами необхідно отримати середню оцінку двох (прямого і зворотного) відповідностей. Отриманий критерій може служити основою для класифікації фреймів за рівнями подібності, що може бути зведено до вирішення відомої задачі побудови класів еквівалентності, яка має квадратичну обчислювальну складність. 4) Третя взаємодія пов'язана, в загальному випадку, з наявністю завжди однакового непорожнього перетину між нерівнозначними множинами, який (перетин) формує нерівнозначні частки від двох, нерівних один одному множин, що беруть участь у вимірі подібності. Це означає, що оцінки, які ми отримуємо при непорожньому перетині: $q(m, s)$ і $q(s, m)$, в загальному випадку також будуть неоднаковими, оскільки $q(m,s)/m$ не дорівнює $q(s,m)/s!$ Окремий випадок

можливий, коли $m=s$, то $q(m,s)/m = q(s,m)/s$, що описано в пункті 1. 5) Таким чином, алгоритм визначення подібності між двома об'єктами зводиться до метриці визначення прямого і зворотного відношення включення за варіантом 2, яку слід вважати універсальною для ідентифікації будь-якого типу відповідності. 6) Залишається відкритим питання ідентифікації якості взаємодії фреймів, оскільки нерівнозначності прямого і зворотного оцінок не є характерною тільки для відносини включення. Хотілося б диференціювати між собою відносини включення і непорожнього перетину між двома множинами шляхом аналізу двох оцінок. Тут не обійтися без більш детального аналізу двох множин компонентів master і slave. Завдання-драфт формулюється так: на основі відомого двудольного графа (відповідності), отриманого з матриці подібності, визначити дві множини в універсальній метриці, закодувати елементи множин в унітарному коді, а далі виконати паралельні логічні операції над двома вектор-множинами з метою обчислення одного з п'яти типів відносин між множинами.

Перетворення скалярної матриці подібності в бінарну дає можливість істотно спростити обчислювальний процес для пошуку подібних патернів: 1) Скалярну матрицю подібності можна трансформувати в бінарну шляхом примусового введення threshold або граничного (порогового) значення, коли необхідно говорити про подібність компонентів фрейма. При цьому координата матриці ідентифікується одиницею, якщо, наприклад, $m(ij) \geq 0,7$, то $m(ij) = 1$ і якщо $m(ij) < 0,7$, то $m(ij) = 0$. 2) Отримана бінарна матриця подібності є зручною формою для отримання структурної якісної взаємодії двох фреймів. Краще рішення - діагональна одинична матриця, яка ідентифікує взаємно однозначну відповідність або структурний рівність двох фреймів. 3) Можливе структурне рішення по бінарній матриці подібності, коли одному компоненту на фреймі поставлено у відповідність кілька фігур на суміжному фреймі - кілька одиничних координат в рядку. Інший варіант, коли одному компоненту на суміжному фреймі поставлено у відповідність кілька фігур на першому фреймі - кілька одиничних координат в стовпці. Можливі більш складні структурні організації одиничних значень бінарної матриці подібності. 4) Бінарна матриця подібності, як форма опису структурної взаємодії, може бути використана для синтезу структури - декартова квадрата подібності послідовності або безлічі фреймів. У такій, завжди квадратичній матриці, будуть фігурувати

скалярні інтегральні оцінки подібності, які можуть бути перетворені в бінарну матрицю M, де поодинокі координати будуть показувати класи еквівалентності або групи подібних між собою фреймів.

M	a	b	c	d	e	f	x	y	z
a	1	0	0	1	0	0	1	0	0
b	0	1	0	0	0	0	0	1	0
c	0	0	1	0	0	1	0	0	1
d	1	0	0	1	0	0	1	0	0
e	0	0	0	0	1	0	0	0	0
f	0	0	1	0	0	1	0	0	1
x	1	0	0	1	0	0	1	0	0
y	0	1	0	0	0	0	0	1	0
z	0	0	1	0	0	1	0	0	1

З метою візуалізації істотних координат матриці далі використовується позначення нульової координати за допомогою точки (.):

M	a	b	c	d	e	f	x	y	z
a	1	.	.	1	.	.	1	.	.
b	.	1	1	.
c	.	.	1	.	.	1	.	.	1
d	1	.	.	1	.	.	1	.	.
e	1
f	.	.	1	.	.	1	.	.	1
x	1	.	.	1	.	.	1	.	.
y	.	1	1	.
z	.	.	1	.	.	1	.	.	1

Аналіз такої матриці на основі визначення безлічі однакових стовпців дає можливість визначити класи еквівалентних об'єктів, які далі можна записати у формі підмножин або у вигляді сукупності рядків або довічних векторів, 1-координати яких ідентифікують компоненти, що входять в кожен клас. Кількість таких векторів дорівнює числу класів еквівалентності, при цьому перетин будь-якої пари векторів між собою дорівнює порожній безлічі:

M	a	b	c	d	e	f	x	y	z
1	1	.	.	1	.	.	1	.	.
2	.	1	1	.
3	.	.	1	.	.	1	.	.	1
4	1

Координати векторів можна перегрупувати шляхом концентрації одиниць для отримання більш структурованої таблиці, що дає можливість візуалізації еквівалентності сусідніх компонентів або об'єктів:

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

Інша форма еквівалентування безлічі об'єктів в системі пов'язана з перетворенням бінарної вихідної матриці подібності в матрицю діагональної симетрії, де фігурально і структурно представлено взаємодію всіх класів еквівалентності в межах матриці декартового квадрата множини компонентів:

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

Наслідком викладеного матеріалу є факти: 1) Поєднання розпізнаних фігур і текстів на скріншоті дає можливість уникнути помилкової подібності фреймів і істотно підвищити якість автоматичного розпізнавання додатків за рахунок розширення метрики фігур (структура, площа, орієнтація, координати), яка доповнена параметром подібності текстів на основі вимірювання відстані по Левенштайну. 2) Інтегральна оцінка подібності скріншотів дає можливість автоматично класифікувати фрейми по додаткам і будувати мережевий графік роботи оператора в часі і просторі, який дозволяє компактно записувати (мінімізувати) серії скріншотів однієї програми в горизонтальних відрізках мережевого графіка, де похідні за оцінками подібності близькі до нуля. 3) Інтегральні оцінки подібності фактично дають можливість ідентифікувати (розпізнавати) 4 види взаємодії між компонентами скріншотів: 1) Рівність компонентів скріншотів $a = b$, коли оцінка подібності дорівнює $Q = 1$. 2) Включення фреймів $a \in b$ або $b \in a$, коли оцінка подібності знаходиться в інтервалі значень $0,5 \leq Q < 1$. При необхідності, тут можна створити процедури поділу даних видів взаємодії. 3) Непорожнє перетин фреймів $a \cap b \neq \emptyset$, коли оцінка подібності знаходиться в інтервалі значень $0,1 \leq Q < 0,5$. 4) Пу-

стий перетин фреймів $a \cap b = \emptyset$, коли оцінка подібності знаходиться в інтервалі значень $0 \leq Q < 0,1$. Даний вид взаємодії скріншотів дає підстави вважати перехід оператора до іншого додатку, що формує вертикальні дуги мережевого графіка.

Недоліком зонної метрики визначення подібності скріншотів є ручна попередня обробка скріншотів з метою визначення надлишкових і віртуальних зонних шаблонів, що специфікують закономірності додатків, які використовуються в компанії. Переваги зонної метрики визначаються урахуванням закономірностей у формуванні шаблонів додатків на основі розуміння макро-топологічної структури, що дає можливість зменшити рівень деталізації компонентів скріншота без зменшення точності і адекватності оцінок подібності фреймів.

7. Висновок

Запропоновано моделі та методи розрізнення соціальних об'єктів, що використовують матричні структури даних для завдання еталонних патернів, які забезпечують паралельне виконання логічних операцій для підвищення продуктивності програмних додатків. Описана модель social-процесу, яка містить ієрархічні компоненти, що знаходяться в теоретикомножинних відношеннях включення один з одним. Запропоновано Pattern Recognition Social-Processor, який використовує хог-операції для розпізнавання соціально-значущих фреймів інформації. Описана матрична автоматна модель моніторингу процесів і явищ для подальшої актуалізації соціальних дій в рамках створення механізмів соціального комп'ютингу. Описана архітектура процесингу для визначення подібності фреймів на основі унітарного кодування даних і хог-порівняння для обчислення подібності-відмінності. Описана метрика для вимірювання подібності геометричних фігур в кіберпросторі, яка оперує нормованими параметрами: координата, орієнтація, суперпозиція, масштабування. Запропоновано процедуру перетворення скалярною матриці подібності в бінарну шляхом примусового введення threshold або порогового значення параметрів, що дає можливість істотно спростити синтез автоматних моделей соціального процесингу. Наукова новизна:

1) Запропоновано алгоритмічну, детерміновану і повторювану структуру логічної моделі функціональної взаємодії компонентів інформаційного фрейму, яка дає можливість перейти від невпорядкованої матричної структури даних до впорядкованої векторної форми, яка в кілька разів підвищує продуктивність обчислювальних процесів для визначення подібності фреймів.

2) Розроблено структурно-функціональний метод визначення подібності фреймів, який має суттєві переваги у швидкодії перед аналогами з пошуку відмінностей між структурними компонентами, завдяки використанню тільки двох паралельних операцій (ог, хог) над двійковими даними матриць суміжностей.

3) Представлений метод визначення подібності фреймів, який дає можливість обчислювати нормовані оцінки відмінності об'єктів на основі бінарних матриць суміжностей графів компонентів, а також структурно-метричні оцінки шляхом використання відстаней між функціонально взаємопов'язаними геометричними фігурами на об'єктах.

4) Запропоновано матрицю структурної ідентичності об'єкта, яка представляє собою шаблон-модель для подальшого створення аналогів: матриці площ геометричних фігур; матриці орієнтації фігури в двовимірному просторі; матриці текстових компонентів; матриці даних. В результаті виходить структурно-функціональна модель фрейму, де в єдиному матричному форматі представлена як структура його компонентів, так і всі його функціональності. Створені матриці по метриці, що містять 5 перелічених параметрів, слід використовувати для отримання двох похідних матриць по кожному параметру, які використовуються для обчислення п'яти локальних оцінок за такою формулою:

$$Q(m,s) = 1 - \frac{\sum_{i=1,n} \sum_{j=1,n} |M_{ij}^m - M_{ij}^s|}{\sum_{i=1,n} \max(M_{ij}^m, M_{ij}^s)}$$

Далі для перелічених вище локальних оцінок обчислюється їх середнє значення $Q=1/5(Q1+Q2+Q3+Q4+Q5)$, яке остаточно формує критерій подібності об'єктів.

5) Подальші дослідження слід спрямувати на розширення спектра параметрів для більш адекватного опису компонентів об'єктів на основі векторної форми з метою істотного підвищення швидкості і якості розпізнавання образів, пов'язаних з інформаційними фреймами в кіберпросторі. При цьому необхідно мати на увазі аксіому: два упорядкованих безлічі чисел формують нормований критерій їх s-подібності (D-відмінності) шляхом ділення суми мінімумів (модулів різниць) цих чисел на суму максимумів дійсних чисел. Приклад обчислення подібності між двома векторами

N =	1	2	3	4	5
M =	11	15	12	17	18
S =	19	14	13	18	16
D =	8/19 =	1/15 =	1/13 =	1/18 =	2/18 =

0,16	0,42	0,06	0,07	0,05	0,1
s=0,84	11/19 =	14/15 =	12/13 =	17/18 =	16/18 =
	0,58	0,94	0,93	0,95	0,9

за формулою відносини між сумою різниць і сумою максимальних значень представлений в наступному вигляді:

$$D = \{[(19-11) + (15-14) + (13-12) + (18-17) + (18-16)] / (19+15+13+18)\} = 1 - 13/83 = 1 - 0,16 = 0,84;$$

$$s = \{[11 + 14 + 12 + 17 + 16] / (19+15+13+18)\} = 70/83 = 0,84.$$

Подальші дослідження будуть спрямовані на формування алгоритмічної повторюваності послідовної обробки фігур і текстів з метою отримання ідентичних моделей об'єктів при довільному масштабуванні останніх. Крім того, необхідно навчитися розпізнавати фрагменти додатку, представленого окремим об'єктом, який є складовою частиною іншого, з більш складною конфігурацією компонентів. Іншими словами, як визначити - чи є заданий фрагмент складовою частиною існуючої складної структури.

Література: 1. *Ulisses Braga-Neto*. Fundamentals of Pattern Recognition and Machine Learning. Springer, 2020. 300 p. 2. *Теория управления*. Дополнительные главы. Учебное пособие. Под редакцией Д.А. Новикова. М.: ЛЕНАРД, 2019. 552с. 3. *Vento M., Percannella G., Colantonio S., Giorgi D., Matuszewski B.J., Kerdegari H., Razaak M*. Computer Analysis of Images and Patterns. Proceedings CAIP 2019 International Workshops. 2019. 129p. 4. *Marinai Simone, Fujisawa Hiromichi*. Machine Learning in Document Analysis and Recognition. Springer. 2008. 434p. 5. *Maji Pradipta, Paul Sushmita*. Scalable Pattern Recognition Algorithms. Applications in Computational Biology and Bioinformatics. Springer. 2014. 304 p. 6. *Fratavchan V. and Fratavchan T*. One pattern recognition method for complex geometric clusters configuration // 2018 International Conference on Development and Application Systems (DAS), Suceava, 2018. P. 200-203. 7. *Sasidharan A. and Palakkeel P*. Performance of Pattern Recognition Algorithms in Identifying Banking Networks // 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018. P. 1463-1467. 8. *Gao Y., Chen Y., Wang J., Tang M. and Lu H*. Dense Chained Attention Network for Scene Text Recognition // 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018. P. 679-683. 9. *Taukute I., Trzcinski T. and Bielski A*. I Know How You Feel: Emotion Recognition with Facial Landmarks // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Salt Lake City, UT, 2018. P. 1959-19592. 10. *Panhwar M.A., Memon K.A., Abro A., Zhongliang D., Khuhro S.A. and Memon S*. Signboard Detection and Text Recognition Using Artificial Neural Networks // 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China. 2019. P. 16-19. 11. *Zhao H., Ye N. and Wang R*. A Survey on Automatic

Emotion Recognition Using Audio Big Data and Deep Learning Architectures // 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), Omaha, NE, 2018. P. 139-142. **12.** Liu M., Xie Z., Huang Y., Jin L. and W Zhou. Distilling GRU with Data Augmentation for Unconstrained Handwritten Text Recognition // 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY. 2018. P. 56-61. **13.** Qadir F. and Peer M.A. Complex Pattern Formation Using Cellular Automata // 2013 International Conference on Machine Intelligence and Research Advancement, Katra, 2013. P. 545-548. **14.** Vidal E., Thollard F., C. de la Higuera, F. Casacuberta and Carrasco R. C. Probabilistic finite-state machines - part II // in IEEE Transactions on Pattern Analysis and Machine Intelligence. July 2005. Vol. 27, no. 7. P. 1026-1039. **15.** Zare M., Jampour M., Arezoomand A. S. and Sabouri M. Handwritten Recognition based on Hand Gesture Recognition using Deterministic Finite Automata and Fuzzy Logic // 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA). Tehran, Iran. 2019. P. 93-99. **16.** Cui Q., El-Arroudi K. and Joos G. An effective feature extraction method in pattern recognition based high impedance fault detection // 2017 19th International Conference on Intelligent System Application to Power Systems (ISAP). San Antonio, TX. 2017. P. 1-6. **7.** Ahangaran M. and Beigy H. Cellular learning automata with external input and its applications in pattern recognition // 2009 Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control. Famagusta. 2009. P. 1-4. **18.** Hahanov V. Cyber Physical Computing for IoT-driven Services. Springer. New York. 2018. 279 p. **19.** https://spectrum.ieee.org/nanoclast/semiconductors/processors/the-accelerator-wall-a-new-problem-for-a-post-moores-law-world?utm_source=circuitsandsensors&utm_campaign=circuitsandsensors-02-19-19&utm_medium=email **20.** Susan Stepney, Steen Rasmussen, Martyn Amos. Computational Matter. Natural Computing Series. Switzerland, Springer, 2018. 336 p.

Transliterated bibliography:

1. *Ulisses Braga-Neto*. Fundamentals of Pattern Recognition and Machine Learning. Springer, 2020. 300 p.
2. *Teorija upravljenija. Dopolnitel'nye glavy. Uchebnoe posobie Pod redakciej D.A. Novikova. M.: LENARD.* 2019. 552s.
3. *Vento M., Percannella G., Colantonio S., Giorgi D., Matuszewski B.J., Kerdegari H., Razaak M.* Computer Analysis of Images and Patterns. Proceedings CAIP 2019 International Workshops. 2019. 129p.
4. *Marinai Simone, Fujisawa Hiromichi.* Machine Learning in Document Analysis and Recognition. Springer. 2008. 434p.
5. *Maji Pradipta, Paul Sushmita.* Scalable Pattern Recognition Algorithms. Applications in Computational Biology and Bioinformatics. Springer. 2014. 304p.

6. *V. Fratavchan and T. Fratavchan,* "One pattern recognition method for complex geometric clusters configuration," 2018 International Conference on Development and Application Systems (DAS), Suceava, 2018, pp. 200-203.
7. *A. Sasidharan and P. Palakkeel,* "Performance of Pattern Recognition Algorithms in Identifying Banking Networks," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018, pp. 1463-1467.
8. *Y. Gao, Y. Chen, J. Wang, M. Tang and H. Lu,* "Dense Chained Attention Network for Scene Text Recognition," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018, pp. 679-683.
9. *I. Tautkute, T. Trzcinski and A. Bielski,* "I Know How You Feel: Emotion Recognition with Facial Landmarks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, 2018, pp. 1959-19592.
10. *M. A. Panhwar, K. A. Memon, A. Abro, D. Zhongliang, S. A. Khuhro and S. Memon,* "Signboard Detection and Text Recognition Using Artificial Neural Networks," 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 2019, pp. 16-19.
11. *H. Zhao, N. Ye and R. Wang,* "A Survey on Automatic Emotion Recognition Using Audio Big Data and Deep Learning Architectures," 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), Omaha, NE, 2018, pp. 139-142.
12. *M. Liu, Z. Xie, Y. Huang, L. Jin and W. Zhou,* "Distilling GRU with Data Augmentation for Unconstrained Handwritten Text Recognition," 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, 2018, pp. 56-61.
13. *F. Qadir and M. A. Peer,* "Complex Pattern Formation Using Cellular Automata," 2013 International Conference on Machine Intelligence and Research Advancement, Katra. 2013. P. 545-548.
14. *E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta and R. C. Carrasco,* "Probabilistic finite-state machines - part II," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 7, pp. 1026-1039, July 2005.
15. *M. Zare, M. Jampour, A. S. Arezoomand and M. Sabouri,* "Handwritten Recognition based on Hand Gesture Recognition using Deterministic Finite Automata and Fuzzy Logic," 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), Tehran, Iran, 2019, pp. 93-99.
16. *Q. Cui, K. El-Arroudi and G. Joos,* "An effective feature extraction method in pattern recognition based high impedance fault detection," 2017 19th International Conference on Intelligent System Application to Power Systems (ISAP), San Antonio, TX, 2017, pp. 1-6.
17. *M. Ahangaran and H. Beigy,* "Cellular learning automata with external input and its applications in pattern recognition," 2009 Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, Famagusta, 2009, pp. 1-4.

18. *Hahanov V.* Cyber Physical Computing for IoT-driven Services. Springer. New York. 2018. 279 p.

19. https://spectrum.ieee.org/nanoclast/semiconductors/processors/the-accelerator-wall-a-new-problem-for-a-post-moores-law-world?utm_source=circuitsand-sensors&utm_campaign=circuitsand_sensors-02-19-19&utm_medium=email

20. *Susan Stepney, Steen Rasmussen, Martyn Amos.* Computational Matter. Natural Computing Series. Switzerland, Springer, 2018. 336 p.

Надійшла до редколегії 03.12.2019

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

Хаханова Ганна Володимірівна, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: стиснення та відновлення двійкової інформації. 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, e-mail: anna.hahanova@nure.ua