

УДК 656.212.5

Косолапов А.А.

МОДЕЛІ ДИСКРЕТНИХ СИСТЕМ РЕАЛЬНОГО МАСШТАБУ ЧАСУ КЕРУВАННЯ СОРТУВАЛЬНИМИ ГІРКАМИ

*Дніпропетровський національний університет залізничного транспорту
імені акад. В. Лазаряна*

У статті розглядаються моделі дискретних систем реального масштабу часу керування сортувальними гірками, способи описання дискретних подій, принципи декомпозиції і композиції дискретних систем керування технологічними процесами на сортувальних станціях.

Ключові слова: дискретні системи, реальний масштаб часу, події, процеси, композиція, декомпозиція.

1. Вступ

Поняття «система реального часу» (real-time system) описує цифрову систему, в якій дискретний обчислювальний процес повинен реагувати на внутрішні і зовнішні події, що відбуваються у випадкові моменти часу [1,2]. Опис таких процесів включає час як незалежну змінну, що приймається практично у всіх теоретичних підходах. У системах реального масштабу часу (СРМЧ) змінна часу набуває дискретних значень і представляється різними способами, наприклад лічильниками (таймерами), що не завжди відповідає умовам безперервного реального часу. До того ж проблема ускладнюється в розподілених інформаційних системах, де виникає необхідність синхронізації або встановлення відповідності систем відліку часу. Характерною особливістю СРМЧ є встановлення обмеження на тривалість системного часу на обробку певної події, що виникла в підсистемі i , і пов'язаної з виконанням певної дії (наприклад, дій з управління записом в зовнішню пам'ять, з видачі управляючого сигналу, тощо) у віддаленій підсистемі j .

У даному підрозділі пропонується абстрактна модель для систем реального масштабу часу, яка поєднує в собі дискретний характер розподіленого обчислювального процесу з безперервним характером часу. *СРМЧ* вивчаються вже тривалий час в області систем управління і побудови АСУТП. Моделі і методи вирішення завдань для таких систем, запропоновані різними дослідниками, засновані на диференціальних рівняннях, які є основним математичним апаратом в теорії управління з моменту її створення [2]. Зростання використання дискретних обчислювальних процесів привело до появи нових методів дослідження цього класу дуже важливих систем. Інтенсивно розвиваються тимчасова логіка, алгебричні системи (I/O автомати), алгебра процесів реального часу [2,3].

У цій роботі пропонуються абстрактні моделі дискретних процесів реального часу, які можна використовувати при розробці принципів декомпозиції/композиції технологічних процесів на сортувальних станціях і відповідних інформаційних систем управління, для розробки інформаційно-часових діаграм загального алгоритму функціонування розподілених систем управління на станціях.

2. Час як змінна в ДСРМЧ

Відправною крапкою для дослідження розглядатимемо загальні властивості *СРМЧ*. Це точка консенсусу у всіх наукових роботах, яка визначає:

- всі моделі повинні поважати принцип причинності;
- система повинна реагувати на події або стимул-реакції;
- система має дискретне подання і реалізацію (у вигляді цифрових пристроїв).

Для цих властивостей базовим є розуміння сутності часу.

Поняття час може бути формалізоване декількома способами залежно від того, що для нас є важливим в дослідженні. У класичній теорії автоматів воно моделюється безліччю натуральних чисел N , що виходить з визначення поняття автомат. Використовуючи таке подання, можна оцінити час роботи складної

програми, виражаючи його в кількості переходів. Час може бути заданий явно і описуватися як спеціальна подія, що нескінченно повторюється, під назвою *clock* (годинник) [2], або може бути явно представлено у визначенні поняття система як інший компонент системи [1]. Вважається, що час - природне розширення класичної теорії автоматів. Недолік такого визначення витікає з необхідності завдання спеціальних правил, які мають бути введені в цілях забезпечення розуміння часових подій. У даній роботі, як і у [3], вважатимемо, що час представляється безліччю позитивних чисел R^+ , і разом із звичайним бінарним відношенням $\leq$ описують модель часу. Легко бачити, що R^+ представляє повну частково впорядковану структуру.

Час є відносним по відношенню до деякої системи відліку, в якій він вимірюється, і, таким чином, кожна система має свій локальний (місцевий) час *local time* (*LT*). Для синхронізації роботи різних підсистем в розподіленій системі вводиться поняття системного (глобального) часу - *global time* (*GT*). Між будь-якими *LT* і *GT* існує ньютонівське часове відношення. Якщо τ є момент часу в *GT*, коли система *S* почала працювати, то цей момент часу в зв'язаних підсистемах відповідатиме t_g в *GT* і t_l в *LT*, які пов'язані вираженням $t_g = t_l + \tau$.

Розглянемо систему *S*, що складається з 2-х підсистем P_1 і P_2 . Кожна підсистема має свій власний локальний час T_1 і T_2 відповідно. Передбачимо, що обоє підсистеми починають виконувати свої функції в деякий момент τ глобального часу *GT*. Зазвичай, між T_1 і T_2 існує ізоморфізм: $id: T_1 \rightarrow T_2$, який показує, що час в підсистемах вимірюється в тому ж порядку.

Системи з дискретними подіями характеризуються тим, що події з'являються в дискретні моменти часу і ці моменти можуть бути в будь-якому місці часової осі. R^+ є безперервною послідовністю чисел від нуля до ∞ і тому кожній події відповідатиме своя точка (момент) на осі часу.

Часові структури представляють не лише точки (моменти) часу на часовій осі. Важливою характеристикою є часові інтервали. Позначатимемо ча-

совий інтервал t_1, t_2 як $\sigma_{t_1}^{t_2}$. Залежності між t_1 і t_2 дозволяють виділити наступні спеціальні випадки, для яких введемо відповідні позначення.

Часовий інтервал, який має обмеження зверху $0, t$ позначатимемо σ^t . Аналогічно, інтервал, який має лише нижнє обмеження t, ∞ , позначатимемо σ_t . Одноточечні інтервали вигляду t, t позначатимемо $\overline{\sigma_t}$.

Розглянемо тепер множину Σ всіх часових інтервалів на R^+ . На цій множині визначимо часткову операцію «+», яку назвемо об'єднання інтервалів (interval concatenation), що виконується таким чином (див. рис. 1):

$$\sigma_{t_1}^{t_2} + \sigma_{t_2}^{t_3} = \sigma_{t_1}^{t_3}$$

$$\sigma_{t_1}^{t_2} + \sigma_{t_3}^{t_4} = \begin{cases} \sigma_{\min t_1, t_3}^{\max t_2, t_4}, & \text{якщо } t_1, t_2 \cap t_3, t_4 \neq \emptyset \\ \text{не визначено, якщо } t_1, t_2 \cap t_3, t_4 = \emptyset \end{cases}$$

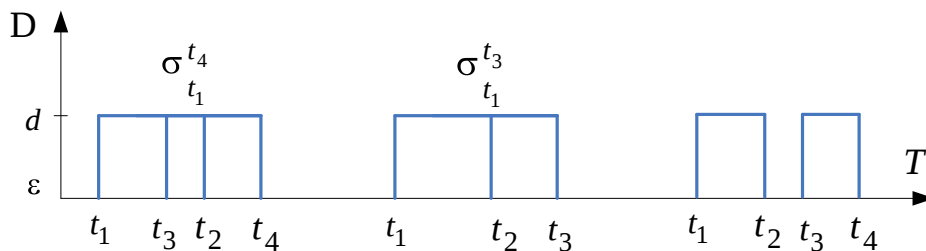


Рис.1. Об'єднання інтервалів, часові події

3. Події і принципи декомпозиції ДСРМЧ на сортувальних гірках

Грунтуючись на часі існування подій, їх класифікуватимемо на *дискретні* (discrete) і *тривалі* (duration) події (наприклад, управління стрілковим приводом і управління гальмуванням відчепів в сповільнювачі). У даній роботі дискретна подія в моделях буде базовим типом, а тривалу подію представимо як послідовність (рядок) дискретних подій. Введемо позначення: dt - дискретні події; it - інтервальні події.

Події визначатимемо двома часовими константами t_1 і t_2 , які задають нижню і верхню межі і визначають валідність (область існування, час життя) подій (див. рис. 1).

Подію можна описати трійкою e, t_1, t_2 . Перша компонента $e \in D$, де D - домен подій, $t_1, t_2 \in R^+$ - часові величини, що визначають час t_1 , коли подія стартувала, почала впливати на систему до тих пір, поки не настає момент часу t_2 завершення події. Події описуються у власній системі відліку. Аби бути точнішим, то якщо подія e, t_1, t_2 з'являється в момент t_d за локальним часом системи, то вона має бути оброблена за інтервал часу $t_d + t_1, t_d + t_2$, аби отримати позитивний ефект (наприклад, встигнути перевести стрілку за маршрутом скачування відчепа на гірці) екты олжно бути оброблено за інтевал часу омент часу завершення события.тся як безліч дискретних подій.). Цей же інтервал в GT має вигляд $\tau + t_d + t_1, \tau + t_d + t_2$.

Визначення 1.1 Множина часових інтервалів в домені D - це множина триплетів e, t_1, t_2 , де $t_1, t_2 \in R^+$, що володіє властивістю $t_1 \leq t_2$, а саме $E = \{ e, t_1, t_2 \mid e \in D, t_1 \in R^+, t_2 \in R^+, t_1 \leq t_2 \}$.

Динаміка системи визначається взаємодією з довкіллям через свої входи і виходи. Хоча в реальних системах множина подій E складається з трьох підмножин (піддоменів): E^w - власне множина вхідних подій із зовнішнього середовища (зовнішнього світу, world); E^{sys} - множина внутрішніх, системних подій (наприклад, сигнали переривань по параметрах, що характеризують стан системи); E^{timer} - множина подій, що ініціюються за станом таймера (контрольно-діагностичні процедури, оновлення баз даних і ін.) [1,2]. Вочевидь, що $E = E^w \cup E^{sys} \cup E^{timer}$. Особливу увагу необхідно приділяти E^w , подіям, що формуються в зовнішньому середовищі. Ця множина є динамічною і такою, що складно структурується, для технологічних процесів на сортувальних станціях і, особливо, на сортувальних гірках. У останньому випадку $E_{Горка}^w \subset E^w$ може бути представлено у вигляді динамічного списку подій, що формується i -м відчепом, що скачується по гірці за m -м маршрутом. Кількість списків в технологічному циклі буде не менше числа відчепів в потязі, що розформову-

ється. У кожному списку будуть три види (підмножини) подій, що динамічно змінюються:

- $(e_{im}, t_1, t_2, \text{mod})$ прогнозовані події для відчепа i , що скачується за маршрутом m ;
- $(e_{im}, t_1, t_2, \text{now})$ поточні події для того ж відчепа;
- $(e_{im}, t_1, t_2, \text{past})$ події, що мали місце з відчепом i , що скачується за маршрутом m , до моменту часу t_1 .

Списки *mod* формуються перед розпуском чергового потягу в результаті моделювання з метою досягнення мінімального часу розпуску. Списки *past* фіксують результати розпуску з метою коректування програми моделювання.

Динамічний характер цих списків обумовлений нештатними ситуаціями на гірці, пов'язаними, наприклад, з нагонами, неперевердженням стрілок і створенням чужаків, і тому подібне. У цих випадках події *mod* і *now* повинні коректуватися в реальному часі, що вимагає відповідних обчислювальних ресурсів для непорушення граничного часу t_2 . Розглянутий підхід до побудови списків подій пов'язаний з принципами декомпозиції/композиції розподіленої ДСРМЧ і принципами управління технологічним процесом розпуску потягів на гірці [4].

В даному випадку говоритимемо про **об'єктно-топологічний принцип декомпозиції**, коли кожен список формується для об'єкту управління, яким є відчеп, що скачується за певним маршрутом відповідно до топології сортувальної гірки. Така організація подій передбачає наявність свого локального часу для кожного об'єкту (відчепа) зі своєю точкою відліку, тобто відчеп «живе» при скачуванні в своїй системі часових координат (LTO_i) , але в рамках *GT*.

Другий спосіб організації $E_{\text{Горка}}^w \subset E^w$ відповідає принципу, який назовемо **принципом технологічної декомпозиції**, коли події пов'язані з технологічними ділянками певного типу, тобто функціонального призначення з точки зору управління процесом розформування потягів на гірці. Наприклад,

стрілочні ділянки, ділянки гальмівних позицій, контролю розчепа і ін. На кожній ділянці, для реалізації певної функції f , встановлений датчик j (один або декілька), що формує відповідну подію $(e^{\bar{f}}, t_1, t_2)$.

Третій принцип декомпозиції можна розглядати як комбінацію перших двох принципів - **принцип функціонально-топологічної декомпозиції** (ділянки групуються за топологією гірки). При цьому зберігається схема формування подій другого принципу декомпозиції - $(e^{\bar{f}}, t_1, t_2)$.

Четвертий принцип декомпозиції - **подійний** - полягає в формуванні класів подій (e^s, t_1, t_2) , що потребують однотипної обробки на апаратно-програмно-алгоритмічному рівні. Тоді клас подій s містить k подій $E_{\text{горка}}^e = \{e_{s1}, e_{s2}, \dots, e_{sk}\}$ і для всіх класів S маємо $E_{\text{горка}}^w = \{E_{\text{горка}}^e \mid \forall s \in S\}$

Вочевидь, що способи декомпозиції/композиції технологічного процесу і керуючої ДСРМЧ взаємозв'язані між собою як з точки зору часових змінних, так і списків подій, що формуються.

Коли система отримує подію, цей момент стає частиною цієї події. Як приклад, розглянемо множину E у вище даному визначенні. Множина можливих рядків подій, які можуть згенерувати зовнішніми процесами (довкіллям), є підмножиною $E \times T^*$ з очевидною умовою, що якщо $i = \dots e_\alpha, t_\alpha \quad e_\beta, t_\beta \dots$, то $t_\alpha < t_\beta$. Визначимо $\omega_{t_1}^{t_2}$ як вхідний рядок, який стартує у момент часу t_1 і завершується в момент t_2 . Якщо t_2 - кінцеве, то і рядок подій також кінцевий.

Події, як вже наголошувалося, описуються у власній системі відліку. Таким чином, якщо подія $e_{t_1}^{t_2}$ з'являється в момент t_d за місцевим часом підсистеми, то воно має бути оброблене в інтервалі $t_d + t_1, t_d + t_2$ для досягнення ефекту. Такий же інтервал в GT має вигляд $\tau + t_d + t_1, \tau + t_d + t_2$. Звичайно, можна дати складнішу інтерпретацію подій, що відбуваються в часі, з «хорошою» поведінкою в інтервалі і «поганою» поведінкою поза ним. Якщо часо-

вий інтервал $0, \infty$, то тоді подія не прив'язана до часу. Пересічення інтервалів може спростити опис подій. Хай $e_{d_{t_1}}^{t_2}$ і $e_{d_{t_3}}^{t_4}$ - дві тимчасові події для деякого символу d з області D . Якщо справедлива операція $\sigma_{t_1}^{t_2} + \sigma_{t_2}^{t_3}$, то $e_{d_{\min(t_1, t_3)}}^{\max(t_4, t_2)}$ представляє обоє події (рис. 2).

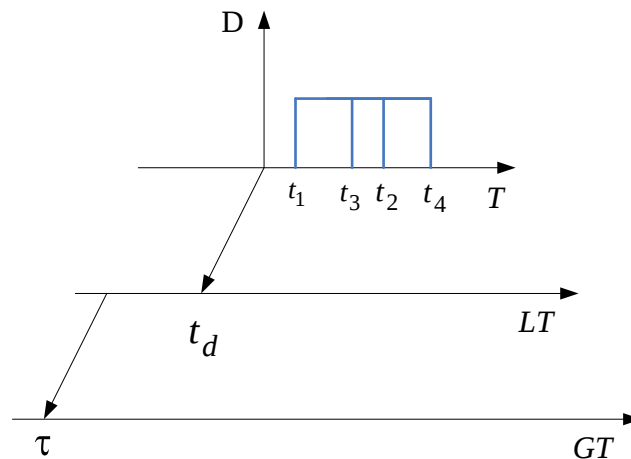


Рис.2. Відношення між часовими посиланнями

Множина допустимих вхідних функцій визначена на підмножині T і в піддомені E . Множина всіх подій, які з'являються на будь-якому кінцевому інтервалі часу, кінцева, тобто має кінцеву потужність. Це вірно, тому що в режимі реального часу система витрачає кінцевий час на обробку будь-якої події. У разі, коли безконечне число подій можуть з'являтися на кінцевому інтервалі часу, система не зможе їх обробити за обмежений час. Така поведінка вважається неприйнятною, тому накладається приведена вище умова. Якщо $\omega_{t_0}^{t_1}$ є вхідною функцією, заданою на інтервалі часу, з кількістю часових точок n , в яких визначена ω , то система оброблятиме всі входи за період часу $n * \mu$. Ω складається з функцій ω і визначається таким чином: $\Omega = \{\omega : T \rightarrow E \mid \omega = e_k(t_{e_k})\}$. Таким чином, будь-яка функція має бути визначена на лічильній множині моментів часу. Якщо $T_d \in T$ - є множина моментів часу, коли визначена функція ω , то існує морфізм $f : T_d \rightarrow N$. Якщо $Card(T_d) = \infty$, то f є ізоморфізмом.

Дана умова гарантує, що так звана страта Zeno [2] на кінцевому інтервалі не з'явиться. Існує і інший тип прояву Zeno. Це відбувається, коли ряд подій, які з'являються на будь-якому кінцевому інтервалі, перевищує кількість місць у вхідній черзі і на обробці. Це пов'язано з обмеженим числом місць в черзі на обробку, з низькою швидкістю обробки подій, з високою інтенсивністю надходження подій в систему [6]. Якщо K - максимальне число подій, які можуть бути буферизовані в системі, то число подій, які можуть з'явитися в інтервалі $\sigma_{t_1}^{t_2}$ повинно бути $N \leq K + |t_2 - t_1|/\mu$. Якщо кількість подій, що з'являються в період часу більше, ніж $K + |t_2 - t_1|/\mu$, то деякі події втрачатимуться. На кінцевому інтервалі часу це означає, що інтенсивність появи подій не повинна перевищувати інтенсивність їх обробки $1/\mu$.

Використовувані вхідні функції або функції обробки вхідних подій, можуть розглядатися як слова деякої мови L_E . Мова визначається на кінцевому алфавіті $E \times T$, де E є набором подій (часових або не часових) і T - це множина часу. Слово в мові w визначається таким чином: $\forall \omega \in \Omega, \forall t \in T$, $w = \omega \ t_0 \ \omega \ t_1 \ ... \omega(t_n)$. Мова (L_E) складається з множини всіх слів w таких, що $w = e_0 \ t_0 \ e_1 \ t_1 \ ...e_n \ t_n$ і $t_0 < t_1 < ... < t_n$, де $e_0, e_1, ..., e_n \in E$ і $t_0, t_1, ..., t_n \in T$. Таким чином, L_E є підмножиною $E \times T^*$.

4. Основні визначення дискретних систем реального масштабу часу

Моделі систем, які ми називаємо дискретними, мають лічильну множину станів. Тривалість переходу із стану в стан залежить від типу події, що обробляє система. Таким чином, інтервал (час) переходу задається функцією $\mu: D \rightarrow T$. Це означає, що на будь-якому кінцевому інтервалі часу система робить лічильне число кроків різної тривалості.

Визначення 1.2 Дискретна Система Реального Масштабу Часу (ДСРМЧ) є вісімка $A = \langle T, Q, E, Y, \phi, \beta, \mu, \hat{\mu} \rangle$, де

- $T = R^+$ множина значень часу;

- $E = E^w \cup E^{sys} \cup E^{timer}$ множина подій і $I = E \times T$ - вхідний алфавіт;
- Q множина символів, відповідних станам;
- $Y = Y_j | j \in J$ індексована множина вихідних величин з $Y_j \in D \times T \times T$;
- $\mu : tran \ e \rightarrow T$ - функція переходів;
- $\hat{\mu}$ - граничне значення функції переходів $\mu \leq \hat{\mu}$;
- ϕ - функція: $\phi : T \times Q \times I \rightarrow Q$, функція наступного стану; якщо система

знаходиться в стані q у момент t , то в стані q_1 система буде у момент часу $t + \mu \ tran \ e_{t_1}^{t_2}$, де $e_{t_1}^{t_2} \in E$ є подія, t_d є момент появи події і $t \geq t_d$ - момент, коли завершується обробка події;

- β - множина загальних функцій $\beta = \{\beta_j | \beta_j : Q \rightarrow Y_j\}$, які називають вихідними функціями.

Вище приведене визначення не виділяє вихідні і фінальні стани. Сформулюємо нове визначення, що розділяє ці стани.

Визначення 1.3 *ifДСРМЧ* - це *ДСРМЧ*, в якій об'єднані множини вхідних $q_0 \in Q$ і кінцевих $F \in Q$ станів.

Клас реактивних систем (*reactive systems*) впливає з приведеного вище визначення, якщо множина F є порожньою. Такі системи називатимемо *iДСРМЧ*.

Робота *ДСРМЧ* зі входами-виходами визначається рядками триплетів e, t, y , де e - подія, що з'являється відповідно до принципів декомпозиції системи, t - момент, коли відбувається подія, y - вихід системи. Всі запуски системи розглядаються в LT і, отже, стартують в момент $t = 0$. Для реактивних систем кількість запусків - нескінченна, а для *ДСРМЧ* - кінцева. Розглянемо *ДСРМВ* A , позначимо Ω_q - множина всіх можливих запусків A , коли система знаходиться в стані q . Для *iДСРМЧ*, коли вхідний стан відомий, цю множину позначимо Ω . Два стани q_1 і q_2 будуть еквівалентними, якщо $\Omega_{q_1} = \Omega_{q_2}$.

Оскільки *iДСРМЧ* стартують завжди з одного стану, q_0 , називатимемо дві системи *iДСРМЧ* A і B еквівалентними, якщо $\Omega_A = \Omega_B$.

5. Композиція і декомпозиція структур *iДСРМЧ*

У даному пункті розглянемо можливість комбінації простих паралельних і послідовних структур, структур із зворотним зв'язком з метою отримання складніших рішень, що відповідають сформульованим вище принципам декомпозиції. У наступних визначеннях встановимо обмеження з метою встановлення морфізму між двома *iДСРМЧ*.

Визначення 1.4 Нехай $M_1 = T, E_1, Q_1, Y_1, \delta_1, \beta_1, q_{01}$ і

$M_2 = T, E_2, Q_2, Y_2, \delta_2, \beta_2, q_{02}$ описують дві *iДСРМЧ*. Морфізм між M_1 і M_2 є множина функцій $\{tm: T \rightarrow T, h: Q_1 \rightarrow Q_2, ev: E_1 \rightarrow E_2\} \cup \{g_j: Y_{1j} \rightarrow Y_{2j} | \forall j \in J\}$,

що задовольняють наступним умовам $f \tau_1 = \tau_2, f \mu_1 = \mu_2, f \hat{\mu}_1 = \hat{\mu}_2,$

$h q_{01} = q_{02}, h \delta_1 t, q_1, e_1, t_d = \delta_2 tm t, h q_1, ev e_1, tm t_d$ і

$\forall j \in J, g_j \beta_{1j} q_1 = \beta_{2j} h q_1$.

Використання трьох типів композицій - послідовної, паралельної і із зворотним зв'язком, дозволяє отримати з простих структур більш складні. Не дивлячись на велику кількість робіт в області теорії систем, теорія композиції в області побудови формальних моделей представлена слабо [3]. Більшість авторів концентрувалися на фундаментальному описі систем і паралельної композиції [5]. Паралельна композиція забезпечує досить точний опис поведінки для не взаємодіючих систем, що функціонують одночасно (паралельно). Для систем, які взаємодіють одна з одною, необхідні складніші конструкції. Деякі визначення паралельної композиції викладені в [2]. Наступні класичні роботи з теорії систем [7] і теорії автоматів [2] вже містять визначення послідовної, паралельної композиції і композиції із зворотним зв'язком.

5.1 Послідовна композиція

У послідовній конфігурації (рис. 3) два автомати зв'язано так, що виходи одного автомата пов'язані зі входами іншого автомата. Звичайно, з'єднання може бути реалізоване тоді і лише тоді, коли множина $Y_{12} \subset E_{21}$. Без цього з'єднання з метою сумісності, друга система не

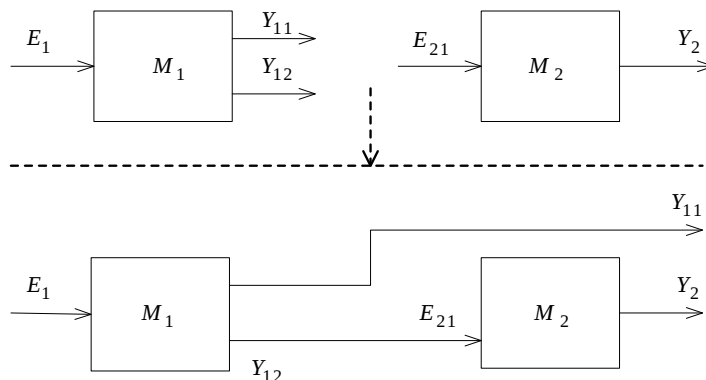


Рис. 3. Послідовна композиція *іДСРМЧ*

зможє зробити жодного переходу. У складеній системі всі компоненти стартують в один і той же час τ і, таким чином, множина T є загальною для обох підсистем. Крім того, вихід з першої системи має бути подією, що формує події в другій підсистемі. Без цього, на виході генеруватиметься незчисленна множина подій.

Множини допустимих входів, є, як правило, різними в різних *іДСРМЧ* через різні часи переходів і довжини вхідних черг. Для того, щоб отримати в результаті працездатну систему, необхідно ввести обмеження на допустимі входи для запуску композитної системи. Допустимі запуски мають відповідні часи переходів. Аби отримати їх для композитної системи, необхідно порівняти часи переходів для її частин. Якщо $\mu_1 \geq \mu_2$, то дії, породжені першою системою з'являтимуться з інтервалами μ_1 , а друга система синхронно оброблятиме їх в процесі появи. Отже, множина прийнятних входів для M_1 є множиною прийнятних входів для композитної системи. Якщо $\mu_1 < \mu_2$, то множина допустимих входів для композитної системи має бути зменшена так, щоб швидкість прибуття подій була менша, ніж $1/\mu_2$. Тому швидкість прибуття подій в

композитній системі повинна перевищувати $1/\max(\mu_1, \mu_2)$. Ця властивість випливає з особливостей побудови «тандемних» (послідовних) систем обробки подій, що в теорії масового обслуговування пов'язано з теоремами Бйорке і Джексона [6].

Надалі розглядатимемо лише випадки, коли $\mu_1 > \mu_2$. Тоді час переходу композитної системи становитиме $\mu = \mu_1 + \mu_2$. Це рівняння стверджує, що виходи змінюються лише після того, як вхідна подія пройде систему.

Для *іДСРМЧ* існує умова, яка може обмежувати довжину (розмір) послідовної композитної системи. Цю умову можна записати таким чином:

$$(\forall e_{t_1}^{t_2} \in E) \sum_{i=1}^L \mu_i \leq \hat{\mu}_{\text{кр}}^e.$$

Визначення 1.5 Нехай $M_1 = (T, E_1, Q_1, Y_1, \delta_1, \beta_1, q_{01})$ і

$M_2 = (T, E_2, Q_2, Y_2, \delta_2, \beta_2, q_{02})$ - дві *іДСРМЧ*, де $Y_1 = (Y_{11}, Y_{12})$, $\beta_1 = (\beta_{11}, \beta_{12})$.

Послідовну композицію M_1 і M_2 позначимо як $(M_1 \Rightarrow M_2)$. Вона описується як

$$M = M_1 \Rightarrow M_2 = (T, E, Q, Y, \delta, \beta, q_0),$$

де

- $Q = Q_1 \times Q_2$ множина станів;
- $E = E_1$ множина подій і $I = E_1 \times T$ вхідний алфавіт;
- $Y = (Y_{11}, Y_2)$ множина виходів;
- $\delta: T \times Q \times I \rightarrow Q$ функція наступного стану;
- $\delta(t, (q_{01}, q_{02}), (e_1, t_d)) = (\delta_1(t, q_{01}, (\beta_{11}(q_{01}), t_d)), \delta_2(t, q_{02}, (\beta_2(q_{02}), t_d)))$;
- $\beta = \beta_{11}, \beta_2$ множина вихідних функцій;
- $q_0 = q_{01}, q_{02}$ вхідний стан.

Розглянемо випадок, коли декілька входів, застосовуються до тих же систем. Є дві можливі точки зору на ці види структуризації входів. Перша передбачає, що події мають бути синхронізовані [3].

Для пояснення цього розглянемо дві *іДСРМЧ* M_1 і M_2 , що працюють паралельно. Нехай E_1 - множина можливих подій для M_1 і E_2 - для M_2 . Тоді, множина подій для результуючої системи буде

$$E' = \{(e_1, \varepsilon) | e_1 \in E_1\} \cup \{(\varepsilon, e_2) | e_2 \in E_2\} \cup \{(e_1, e_2) | e_1 \in E_1, e_2 \in E_2\}.$$

Множина подій для паралельної композиції означає, що події можуть з'явитися в один і той же момент часу, тобто синхронно, і можуть з'явитися асинхронно. Для приведення цих ситуацій до одного вигляду введемо порожню подію ε для узагальненого синхронного представлення композиції. Такий підхід розглядає всі події в черзі до одного блоку, як такі, що поступають і обробляються синхронно. Якщо $I_j = E_j \times T$ - це вхідний алфавіт для одного входу, тоді, оскільки $T \times T \times \dots \times T \cong T$, алфавітом для композитних входів буде $E_1 \times E_2 \dots E_n \times T$.

5.2 Паралельна композиція

Другий підхід передбачає, що входи обробляються незалежно. Проте, така точка зору суперечить нашому припущенню, що система - монолітна.

У паралельній конфігурації множина подій E в результуючій машині є підмножиною декартова добутку множин складових подій об'єднаних машин $E \subset E_1 \times E_2$. Як і в послідовній композиції, в даному випадку обоє підсистеми мають свою систему відліку часу і свій початковий момент часу (час старту). Це гарантує, що елементи декартова добутку відповідають подіям машини, що скомпонована.

Визначення 1.6 Нехай $M_1 = (T, E_1, Q_1, Y_1, \delta_1, \beta_1, q_{01})$ і $M_2 = (T, E_2, Q_2, Y_2, \delta_2, \beta_2, q_{02})$ - дві *іДСРМЧ* (рис. 4) з $E_1 = (E_{11} \times E_{12})$ і $E_2 = (E_{21} \times E_{22})$. Паралельну композицію M_1 і M_2 позначимо $(M_1 \parallel M_2)$. Вона представляє систему $M = M_1 \parallel M_2 = (T, E, Q, Y, \delta, \beta, q_0)$,

де

- T множина значень часу;
- $Q = Q_1 \times Q_2$ - множина станів і $q_0 = (q_{01}, q_{02})$;

- $E = (E_{11} \times Z \times E_{22})$ з $Z \subseteq E_{12} \cap E_{21}$ - множина подій і

$I = E_{11} \times Z \times E_{22} \times T$ - вхідний алфавіт;

- $Y = Y_1, Y_2$ множина вихідних значень;

- $\delta: T \times Q \times I \rightarrow Q$ функція наступного стану, визначувана як

$\delta(t, ((q_1, q_2), (e_1, z, e_2, t_d))) = (\delta_1(t, q_1, (e_1, z, t_d)), \delta_2(t, q_2, (e_2, z, t_d)))$;

- $\beta = \beta_1, \beta_2$ множина вихідних функцій.

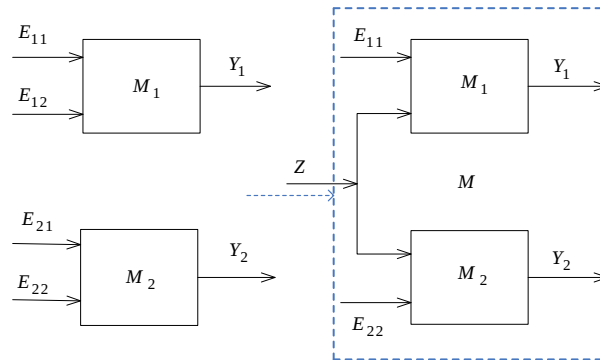


Рис.4. Паралельна композиція іДСРМЧ

Час переходів в композитній системі рівний $\max(\mu_1, \mu_2)$ і повинен задовольняти умові $\max(\mu_1, \mu_2) \leq \min(\hat{\mu}_1, \hat{\mu}_2)$.

5.3 Композиція зі зворотнім зв'язком

Введення зворотних зв'язків в систему, коли виходи системи з'єднуються з її входами, утворює іншу систему. Системи із зворотним зв'язком можуть виконати декілька переходів по одній події на вході, оскільки їх вихід на вході дає нову подію, яка повинна оброблятися. іДСРМЧ, за **визначенням 1.2**, виконують лише один перехід при появі події. В цьому випадку ми визначимо, що перехід в системі із зворотним зв'язком матиме місце лише тоді, коли система досягне стійкого стану, і знаходитиметься в очікуванні вхідної події. Вочевидь, що система може зробити декілька внутрішніх переходів, аби увійти до такого стану. Множина допустимих функцій входу для системи із зворотним зв'язком - це лише підмножина множини функцій входу у первинній системі. У

результаті, зворотний зв'язок призводить до збільшення часу обробки і робить нестійким виконання умови $\max(\mu) \leq \min(\hat{\mu})$.

Визначення 1.7 Нехай $M_1 = T, E_1, Q_1, Y_1, \delta_1, \beta_1, F_1, q_{01}$ - це *іДСРМЧ* (рис. 5) з $E_1 = E_{11} \times E_{12}$ і $Y_1 = Y_{11}, Y_{12}$. Зворотний зв'язок в M_1 утворює систему

$$M = (T, E, Q, Y, \delta, \beta, F, q_0)$$

де

- T множина значень часу;
- $Q = Q_1$ множина станів з $q_0 = q_{01}$;
- $E = E_{11}$ множина подій і $I = E \times T$ - вхідний алфавіт;
- $Y = Y_{11}$ множина виходів;
- $\beta = \beta_1$ множина вихідних функцій;
- $\delta: T \times Q \times I \rightarrow Q$ функція наступного стану:

$$q_1(t + n * \mu) = \delta(t, q, (e_{11}, t_d)) = \delta_1(\dots(\delta_1(t, q, (e_{11}, t_d))))).$$

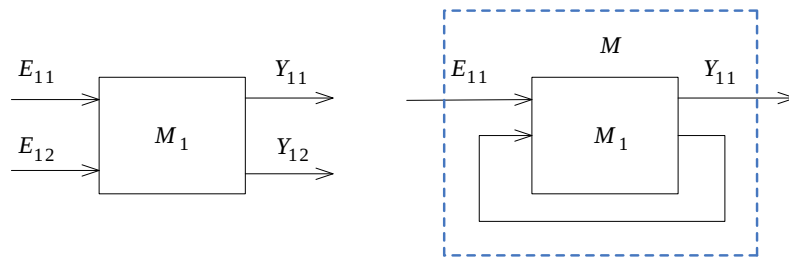


Рис. 5. Композиція *іДСРМЧ* із зворотним зв'язком

Завдяки своїй природі, деякі системи із зворотним зв'язком можуть мати безконечний сумарний час переходів, тому що система виконуватиме безконечне число внутрішніх переходів. Це означає, що клас всіх *іДСРМЧ* не вичерпується операцією введення зворотного зв'язку. Застосовувати її необхідно з крайньою обережністю.

Література:

1. Дреус Ю.Г. Системы реального времени: технические и программные средства: Учебное пособие. М.: МИФИ, 2010. – 320 с.
2. REAL-TIME SYSTEMS Modeling, Design and Applications. AMAST Series in Computing — Vol. 8 World scientific. 2007.
3. D. Ionescu. Designing supervisor for real-time systems. In T. Rus & C. Rat-tray, editors, Theories and experiences for real-time system development, pages 103-128. World Scientific Publishing Company, 1994.
4. Косолапов А.А., Шафит Е.М. Принципы декомпозиции и интеграции автоматизированных систем управления сортировочными станциями // Автоматизированные информационно-управляющие вычислительные системы на сортировочных станциях железнодорожного транспорта. - Межвуз. сб. научн. трудов. - Днепропетровск: изд. ДИИТа, 1993. - Вып. № 289/18. - С. 5-17.
5. J. F. Sogaard-Anderson, N. Lynch & B. Lampson. Correctness of communication protocols. Technical report TR-589, MIT Computer Science Laboratory, 1993.
6. Косолапов А.А. Аналітичні моделі масового обслуговування в задачах аналізу і проектування комп'ютерних систем і мереж (на залізничному транспорті) : навчальний посібник / А. А. Косолапов. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна , 2012. – 154 с.
7. Месарович М., Мако Д., Такахара И. Теория иерархических многоуровневых систем.-М.: Мир, 1972.