

УДК 651.3:518.5

ТЕХНОЛОГІЯ ПРЕДСТАВЛЕННЯ МОДЕЛІ АДАПТИВНОЇ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ТА КОНТРОЛЮ ЗНАНЬ

П.І. Федорук

Прикарпатський національний університет ім. Василя Стефаника

e-mail: pavlo@pu.if.ua

1. Вступ

Використання сучасних інформаційних технологій у навчальному процесі дозволяє підвищити якість навчального процесу і підсилити освітні ефекти від застосування інноваційних педагогічних програм і методик, оскільки дає викладачам додаткові можливості для побудови індивідуальних освітніх траєкторій студентів. Застосування інформаційних технологій дозволяє реалізувати диференційований підхід до студентів з різним рівнем готовності до навчання. Інтерактивні навчальні програми, які базуються на гіпертекстовій структурі і мультимедіа, дають можливість організувати одночасне навчання студентів, які володіють різними здібностями і можливостями. Початок адаптивного навчання - час виникнення педагогічних праць Коменського, Песталоцци і Дістервега. А. Дістервег: "Викладай відповідно до природи... Вчи без прогалин... Починай викладання з того, на чому зупинився учень... Без знання того, на чому зупинився учень, неможливо добре навчити його". Сама ідея, яка існувала вже давно, в сьогоденній інтерпретації з використанням сучасного рівня інформаційних технологій може набути практичної реалізації і автоматизації. Тому створення моделі для побудови адаптивної системи навчання є важливою і актуальною задачею.

2. Адаптивна система дистанційного навчання

Традиційно навчальний курс розділяють на логічні частини – заняття. Проте, для реалізації адаптивних можливостей в системі дистанційного навчання заняття пропонується розділяти на менші частини – кванти знань. Квант – найменша неподільна смислова порція інформації (первісне поняття, ключове слово, аксіома, означення, тощо) [1, 2]. Кожен квант може використовуватися в різних заняттях.

В процесі навчання над квантами виконуються операції. Одна і та ж операція може виконуватись над різними квантами. Перед виконанням операції над квантом він мусить бути опрацьованим, тобто, вивченим, засвоєним студентом. Таким чином, навчальний курс може бути представлений як множина квантів, які треба засвоїти і множина операцій, які треба виконати над квантами для їх засвоєння. Кожному кванту відповідає декілька операцій і навпаки, одну і ту ж операцію можна виконувати над різними квантами.

Отже, адаптивність процесу навчання буде представлена в послідовності подання квантів на опрацювання, чергуванні виконання операцій над квантами та їх опрацюванні, присутності кванта в певному занятті.

Слід зазначити, що можливі два способи прив'язки певного кванта до певного заняття. В першому випадку заняття визначається набором квантів, які мають бути засвоєні. На основі присутності квантів в занятті визначається набір операцій, які мають бути виконані в цьому занятті. В другому випадку, заняття задається набором операцій, а кванти, які мають бути засвоєними вибираються за зв'язками з відповідними операціями.

При реалізації такого способу адаптивності потрібно створити модель керування та опрацювання потоком квантів. Для цього можна використати різні технології. Наприклад, ієрархічну модель дедуктивних запитів [3], мережі Петрі [4], потокові схеми Келлера [5]. Оцінивши з точки зору ефективності застосування дані технології для реалізації поставленої задачі запропоновано формальну модель з використанням елементів теорії графів [6] та теорії автоматів [7].

3. Графо-автоматна модель адаптивної системи дистанційного навчання

Формально схему управління та опрацювання потоком квантів можна зобразити у вигляді орієнтованого графа QSM (Quantum Stream Management), вершини та ребра якого являють собою структурні автомати різних конструкцій (в залежності від призначення вершини чи ребра).

Структурним автоматом вважаємо мережу автоматів, вхід якої має n каналів ($X_1...X_n$), вихід має p каналів ($Y_1...Y_p$) [7]. Внутрішня пам'ять структурного автомата складається з k елементів ($Q_1...Q_k$). Кожному елементу пам'яті відповідає функція переходу $\varphi_1... \varphi_k$, а кожному вихідному каналу функція виходу ($\psi_1... \psi_p$). В роботі графу використовується поняття дискретного часу τ . В кожен момент часу τ на вхідні канали $X_i[\tau]$ структурного автомата поступають числові сигнали, які перетворюються функціями переходу φ_j і результат запам'ятовується у відповідному елементі пам'яті $Q_j[\tau] = \varphi_j(X_1[\tau], ..., X_n[\tau], Q_1[\tau - 1], ..., Q_k[\tau - 1])$, $j = 1..k$. Після переобчислень всіх елементів пам'яті з допомогою функцій виходів обчислюються числові значення, які будуть встановлені на вихідних каналах структурного автомата в даний момент дискретного часу τ : $Y_i[\tau] = \psi_i[\tau](X_1[\tau], ..., X_n[\tau], Q_1[\tau], ..., Q_k[\tau])$, $i = 1..p$.

До структурного автомата будемо застосовувати два види операцій – запис вектора сигналу на вхід автомата та зчитування значення з конкретного виходу автомата.

Нехай AVT - структурний автомат. Визначимо функції:

$G_{AVT}(Y_j)$ - повертає значення з j -го виходу автомата AVT в даний момент дискретного часу τ ;

$S_{AVT}(x_1...x_n)$ - подає на входи ($X_1...X_n$) автомата AVT відповідні значення ($x_1...x_n$).

Для цієї функції будемо застосовувати скорочений запис – $S_{AVT}(X_i = x)$ - подає на вхід X_i автомата AVT значення x , а на інші входи не впливає.

При розгляді елементів графа QSM як автоматів виникає необхідність опрацювати окремий вхідний чи вихідний канал автомата. Для таких ситуацій введемо позначення $[N].K$, де N - структурний автомат, K - вхідний чи вихідний канал.

Граф QSM складається із структурних автоматів, які взаємодіють з допомогою вхідних та вихідних каналів. Передача числового сигналу від автомата A до автомата B відбувається за два моменти дискретного часу. В перший момент на вихідному каналі $[A].Y$ встановлюється значення в результаті роботи відповідної вихідної функції. В другий момент часу це значення переноситься на канал $[B].X$. Зв'язок виходу $[A].Y$ і входу $[B].X$ будемо позначати $[A].Y \rightarrow [B].X$.

Розглянемо структуру графу QSM.

$QSM = \langle \text{Nodes}, \text{Edges}, \text{IO} \rangle$,

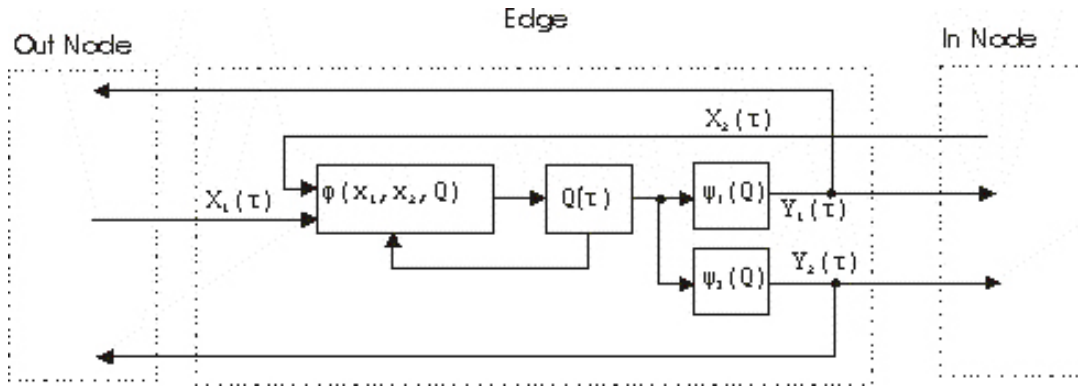
де Nodes - множина вершин графа QSM, Edges – множина ребер графа QSM, IO – множина портів зовнішнього обміну даними з зовнішнім середовищем.

Опишемо множину Edges . $\text{Edges} = Ed_c \cup Ed_d$,

де Ed_c - множина орієнтованих управляючих ребер, Ed_d - множина орієнтованих ребер даних.

Орієнтоване ребро направлене від вихідної вершини до вхідної. Управляючі ребра з множини Ed_c можуть перебувати в двох станах – активний і неактивний. Ребро переходить в активний стан після відповідної команди від вихідної вершини. В неактивний стан ребро переводиться командою вхідної вершини. Введемо поняття часу t знаходження ребра в активному стані як ще одну характеристику управляючого ребра. Для активного ребра $ed \in Ed_c$ час t дорівнює кількості інтервалів дискретного часу τ з моменту активації ed . Для неактивного ребра ця характеристика дорівнює 0.

Як структурний автомат елемент Ed_c має таку схему (Рис.1):


 Рис. 1. Схема елемента Ed_c

Тут функції φ, ψ_1, ψ_2 на основі вхідних даних X_1, X_2 та проміжного стану Q в кожен момент дискретного часу τ визначають вихідні сигнали Y_1 та Y_2 :

$$Q(\tau) = \varphi(x_1(\tau), x_2(\tau), Q(\tau - 1)) = \begin{cases} 0 & x_1(\tau) = 0 \vee x_1(\tau) = 0 \wedge x_2(\tau) = 0 \wedge Q(\tau - 1) = 0 \\ Q(\tau - 1) + 1, & x_1(\tau) = 1 \wedge x_2(\tau) = 1 \vee x_1(\tau) = 0 \wedge x_2(\tau) = 0 \wedge Q(\tau - 1) > 0 \end{cases};$$

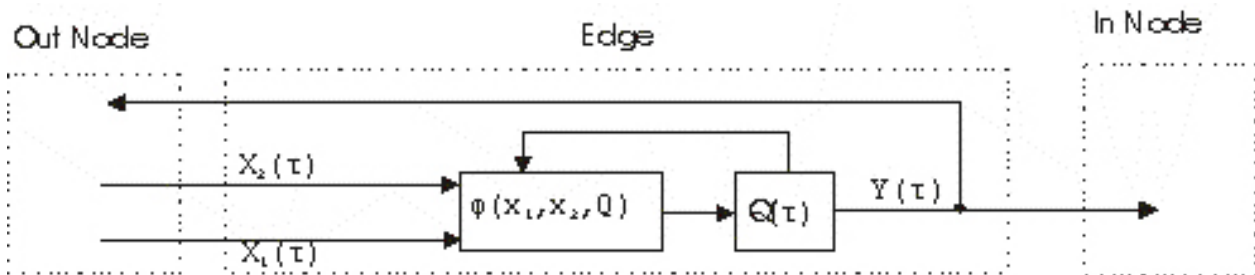
$$Y_1(\tau) = \psi_1(Q(\tau)) = Q(\tau);$$

$$Y_2(\tau) = \psi_2(Q(\tau)) = \begin{cases} 1, & Q(\tau) > 0 \\ 0, & Q(\tau) = 0 \end{cases};$$

З наведених формул видно, що вхід X_1 призначений для активації ребра вихідною вершиною, а вхід X_2 — деактивації вихідною вершиною. Вихід Y_1 своїм значенням має час активності. Вихід Y_2 визначає активність ребра.

Елементи множини Ed_d виконують функцію елементів пам'яті в процесі роботи графа QSM. Характеристикою ребер-елементів Ed_d є вага. Це числова величина, яка встановлюється вихідною вершиною і утримується ребром до наступних змін, незалежно від стану вихідної вершини.

Для опису елементів множини Ed_d як структурних автоматів наведемо таку схему (Рис.2):


 Рис. 2. Схема елементів множини Ed_d

Вихід цього автомата $Y(\tau)$ визначається його станом $Y(\tau) = Q(\tau)$. Стан $Q(\tau)$ в кожен момент часу τ переобчислюється як результат функції φ :

$$Q(\tau) = \varphi(x_1(\tau), x_2(\tau), Q(\tau - 1)) = \begin{cases} Q(\tau - 1), & x_2(\tau) = 0 \\ x_1(\tau), & x_2(\tau) = 1 \end{cases}$$

На вхід подаються два сигнали: X_1 — нова вага ребра, ненульове значення каналу X_2 означає необхідність встановити нове значення ваги ребра, тобто X_1 . Якщо X_2 рівне 0, ваги не міняються.

Для роботи вище описаного графа потрібні механізми контакту з зовнішнім середовищем — Z. Як рішення цієї проблеми використовується ІО — множина портів обміну управляючими

сигналами та даними із Z . Ці порти використовуються для запуску алгоритму описаного в графі, отримання сигналів про закінчення, для введення даних та виведення результатів. Порти, які належать множині ІО подібно до ребер (множина Edges) розділені на дві підмножини – управляючі порти та порти даних, є структурними автоматами таких самих конструкцій і мають ті ж характеристики. Елементи ІО одним кінцем з'єднані з вершиною, яка належить Nodes, іншим кінцем з'єднані із Z . Порти також є орієнтованими, тобто мають початок і кінець. Множина ІО складається з чотирьох підмножин – $IO = IO_{d,in} \cup IO_{d,out} \cup IO_{c,in} \cup IO_{c,out}$, де

$IO_{c,in}$ - множина управляючих портів, які мають вхідну вершину в множині Nodes, активуються ззовні;

$IO_{c,out}$ - множина управляючих портів, які мають вихідну вершину в множині Nodes, деактивуються (переходять в неактивний стан) ззовні;

$IO_{d,in}$ - множина портів даних, які мають вхідну вершину в множині Nodes, вага встановлюється ззовні;

$IO_{d,out}$ - множина портів даних, які мають вихідну вершину в множині Nodes.

Основне призначення елементів ІО є запуск алгоритму керування потоком квантів та повідомлення про завершення алгоритму. Також елементи ІО застосовуються для контакту QSM з механізмом відображення квантів та процесу виконання операцій (інтерфейс студента). Крім того, порти (елементи ІО) дають можливість отримувати різноманітну статистичну інформацію про процес проходження курсу. Наприклад, рівень засвоєння конкретного кванта чи заняття, загальний рівень засвоєння курсу, відсоток опрацьованого матеріалу, тощо.

Таким чином можна сказати що ІО – це механізм обміну інформацією між ядром навчальної системи (QSM) та інтерфейсом студента.

Кожне ребро графа QSM можна описати так:

$$set : sNode \xrightarrow{mdf} eNode,$$

де set - множина, до якої належить ребро ($Ed_c, Ed_d, IO_{c,in}, IO_{c,out}, IO_{d,in}, IO_{d,out}$),

$sNode$ - вихідна вершина, $eNode$ - вхідна вершина, mdf – модифікатор, використовується у випадках коли між двома вершинами існує більше ніж одне ребро одного і того ж типу.

Вершини графа QSM також являють собою структурні автомати. Оскільки вершини за своїм логічним призначенням поділяються на значно більше груп ніж ребра, то і конструкція відповідних автоматів є різною. Проте, для опису вершин графа QSM введемо деякі загальні поняття.

Вершини графа QSM об'єднані в множині Nodes. Вершини характеризуються множинами P_{in} та P_{out} . P_{in} - множина вхідних ребер з Edges та вхідних портів з $IO_{d,in} \cup IO_{c,in}$. P_{out} - множина вихідних ребер з Edges та вихідних портів з $IO_{d,out} \cup IO_{c,out}$. В свою чергу кожна з цих двох множин складається з двох підмножин. $P_{in} = PC_{in} \cup PD_{in}$, $P_{out} = PC_{out} \cup PD_{out}$. PC_{in} - вхідні управляючі ребра (порти), PC_{out} - вихідні управляючі ребра(порти), PD_{in} - вхідні ребра(порти) даних, PD_{out} - вихідні ребра(порти) даних.

Вхідним ребром для вершини з Nodes називається таке ребро, для якого ця вершина є вхідною. Аналогічно для вихідного ребра.

Кожна вершина $nd \in Nodes$ являє собою скінчений комбінаційний автомат, тобто автомат, який не має внутрішньої пам'яті і вихідні сигнали в кожен момент дискретного часу встановлюються тільки на основі вхідних сигналів. В наших позначеннях – вершина $nd \in Nodes$ кожен момент дискретного часу τ використовує вихідні канали елементів множини P_{in} як вхідні дані і застосувавши функції виходів подає числові значення на свої вихідні канали, тобто на вхідні канали елементів множини P_{out} . Відносно вершини ребра мають скорочене позначення:

$$set : \xrightarrow{mdf} Node \in P_{in}, set : Node \xrightarrow{mdf} \in P_{out}.$$

Тут *Node* - вершина на іншому кінці графа.

В графі QSM є множина ребер, які мають одну і ту ж вершину в ролі як вхідної, так і вихідної (петля). Наприклад, $Ed_c: Quantum_k \rightarrow Quantum_k$ - використовується для збереження загального рівня засвоєння кванта в результаті виконання різних операцій. Відносно вершин такі ребра позначаються – $set \xrightarrow{mdf}$.

Загальну схему вершини графа QSM як структурного автомата можна зобразити так (Рис.3):

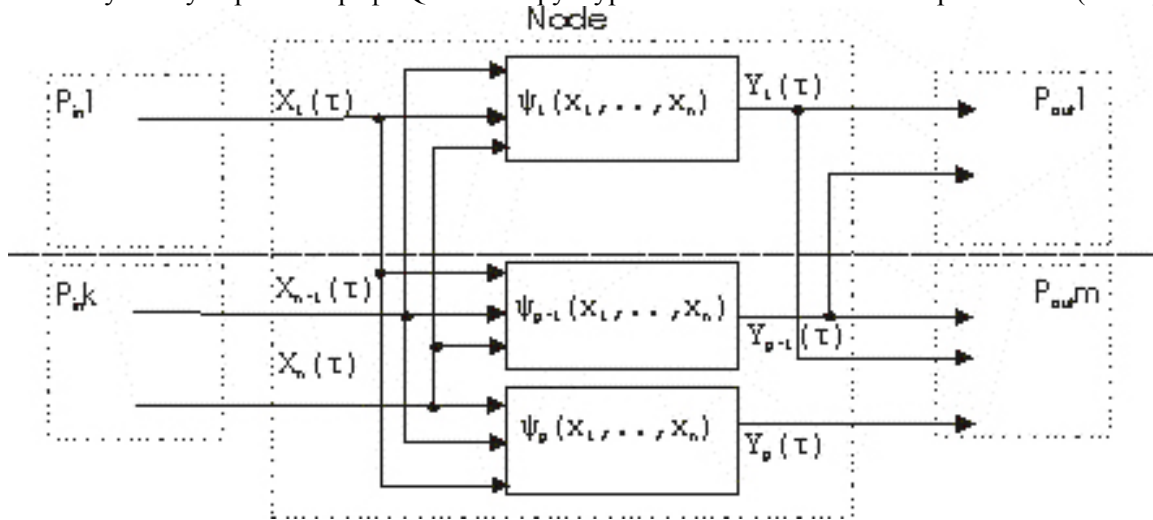


Рис.3. Загальна схема вершини графа QSM

Кількість вхідних та вихідних каналів, вирази функцій виходів залежать від призначення вершини. Більшість аналітичних виразів для функцій $\psi_1, \dots, \psi_p$ є складеними функціями (значення залежить від виконання логічних умов).

6. Висновок

Описана модель представлена за допомогою графа QSM вирішує задачу оптимального управління потоком квантів знань в адаптивній системі дистанційного навчання та контролю знань в процесі вивчення курсу. До переваг описаної моделі можна віднести:

- збереження логічної моделі навчального курсу в структурі графу QSM;
- інтерфейсна частина навчальної системи не залежить від конкретного курсу і є периферійним модулем ядра курсу, описаного графом QSM;
- забезпечення оптимального чергування між процесами виконання операцій та засвоєння нових квантів знань;
- вибір найбільш прийнятної послідовності видачі квантів знань для засвоєння.

Описана модель управління потоком квантів дозволяє реалізувати в системі дистанційного навчання і контролю знань значні можливості адаптації до особливостей сприйняття і рівнів знань тих, хто навчається.

Література:

1. Федорук П.І. Технологія розробки навчального модуля в адаптивній системі дистанційного навчання та контролю знань // Математичні машини і системи. – 2005р. – №3. – С.155-165.
2. Сирожа И.Б. Квантовые модели и методы инженерии знаний в задачах искусственного интеллекта // Искусственный интеллект. Научно-технический журнал. – Донецк, 2002. -№3. – С.161-171.
3. Аднан Али. Иерархическая формальная модель дедуктивных запросов // Математичні машини і системи. – 1999. -№2. – С. 70-77.
4. Котов В.Е. Сети Петри. – М.: Наука, 1984.
5. Вальковский В. Распараллеливание алгоритмов и программ. Структурный подход.- М.:Наука,1989.
6. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978.
7. Пономарев В.Ф. Конечные автоматы. Учебное пособие. – Калининград: КГТУ, 1999.