

МОДЕЛЮВАННЯ ПЛАНУ ПОВЕДІНКИ ІНТЕЛЕКТУАЛЬНОГО АГЕНТА НА ОСНОВІ МЕРЕЖ ПЕТРІ ТА ОНТОЛОГІЧНОГО ПІДХОДУ

© Литвин В.В., 2009

Розглянуто побудову плану діяльності інтелектуального агента. Для моделювання поведінки інтелектуального агента використано мережі Петрі та онтологічний підхід. Показано, що розроблену модель поведінки можна звести до задачі асинхронного динамічного програмування.

Ключові слова – інтелектуальний агент, мережі Петрі, онтологічний підхід.

Article considers intelligent agent functioning plan development. For intelligent agent behavior modeling Petri nets and ontology method were used. It was shown that developed behavior model could be transformed into asynchronous dynamic programming task.

Keywords – intelligent agent, Petri nets, ontology method.

Постановка проблеми у загальному вигляді

Моделювання плану поведінки інтелектуального агента, мета якого полягає у досягненні кінцевого стану, є актуальною задачею сьогодення, оскільки саме від правильно побудованих моделей залежать кількісні показники, отримані під час досягнення цілі (затрачені ресурси, витрачений час для досягнення цілі, оцінка досягнутого стану тощо). З літератури відомо, що для моделювання такого плану знаходження оптимального рішення використовують стохастичні або детерміновані мережі з вершинами типу I/АБО і методи пошуку вглиб, вишир з різноманітними евристичними функціями [1]. Однак ці моделі можна застосувати для задач, в яких стани чітко задаються множиною фактів, які реалізуються однією з формальних логік або набором продукційних правил та не враховують витрати ресурсів. Очевидно, що у складних прикладних областях опис станів та вибір альтернатив для відповідних переходів між станами вимагає зовсім інших підходів, а отже, і моделей, які ґрунтуються на онтологіях задач та онтології предметної області [2].

З огляду на постановку задачі – досягнення цільового стану інтелектуальним агентом – для її розв’язування запропоновано використовувати мережі Петрі [3] для моделювання шляхів (процесів) досягнення цільового стану, байєсівські мережі [4] для моделювання імовірнісних оцінок переходів між станами, онтологію предметної області для опису станів та обчислення необхідно затрачуваних ресурсів для переходів між станами.

Формування цілей

Розробити модель функціонування інтелектуального агента, поведінка якого полягає у досягненні деякого цільового стану з мінімальною витратою ресурсів.

Основні припущення та поняття

Для досягнення цільового стану інтелектуальний агент (ІА) насамперед повинен змоделювати план досягнення цього стану зі всіма можливими альтернативами. Процес планування ґрунтується на декомпозиції. Задача планування ZP містить три складові: множину станів St , множину дій F , множину цілей Z ; тобто

$$ZP = \langle St, F, Z \rangle.$$

Свою чергою, дія складається із трьох частин: ім’я дії і список параметрів, передумова та результат. А сам план визначається як кортеж з чотирьох елементів – \langle Множина дій, Множина

обмежень впорядкування, Множина причинних зв'язків, Множина відкритих передумов» [5]. Для урахування декомпозиції, і/або залежностей між станами та переходами, відображення альтернатив досягнення цільових станів пропонуємо використовувати мережі Петрі. Приклад такої мережі з цільовим станом $St(z)$ наведено на рис. 1.

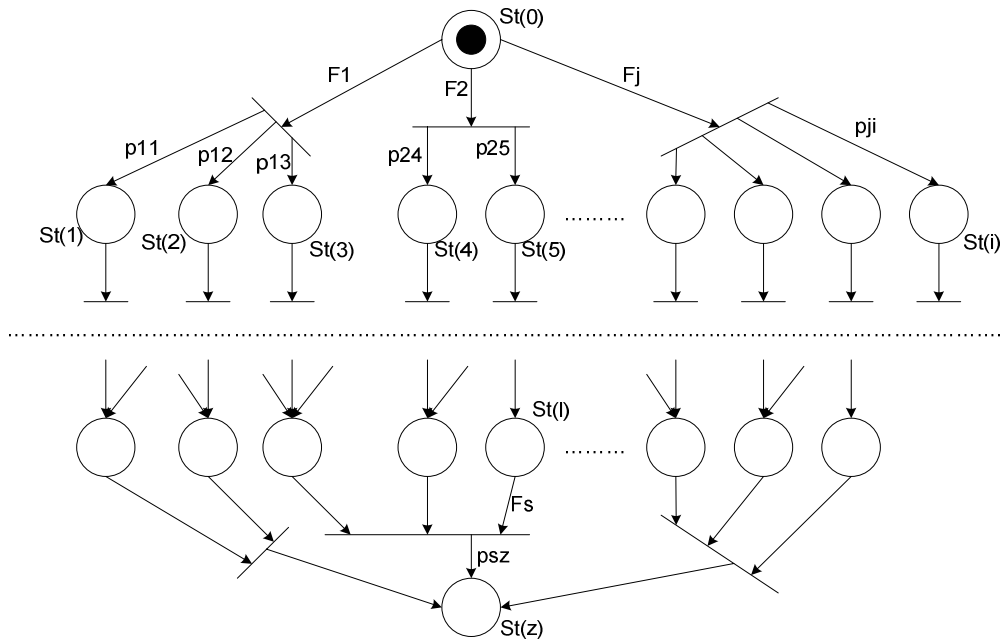


Рис. 1. Приклад мережі Петрі моделювання поведінки інтелектуального агента

На початку фішка мережі перебуває у стані $St(0)$. Тоді очевидно, що суть функціонування ІА полягає у переміщенні фішки в кінцевий стан $St(z)$ з мінімальною витратою ресурсів (ресурси можуть задаватися у формі грошового еквівалента, людино-годин, часу тощо). Щоб почати розв'язувати цю задачу, спочатку треба довести, що мережа Петрі побудована так, що кінцевого стану $St(z)$ досягти насправді можна. Відомо, що для такого доведення використовуються планувальники – програми, які шукають розв'язок або доводять неіснування розв'язку. Ця робота оминає розгляд такого доведення. Апостерорно вважаємо, що такий перехід існує і не один, бо інакше задача немає змісту.

Стан $St(i)$ характеризується базою даних та базою знань (сховищем даних), поданих у вигляді множини фактів з відповідними ймовірнісними оцінками.

Дія F_{ij} подається у вигляді відображення зі стану $St(i)$ в стан $St(j_i)$ з відповідною ймовірністю p_{ji} , тобто: $St(i) \rightarrow St(j_i)$ з ймовірністю p_{ji} .

Оцінка станів

Для вибору необхідних дій користувач повинен вміти оцінювати стани – як ті, в яких він вже перебував, так і бажані. Нехай $q(St(i))$ – оцінка стану $St(i)$. Для оцінки станів, в яких вже перебував раціональний агент, використовуватимемо онтологію предметної області O . Вважатимемо, що онтологія однозначно визначає ідеальний опис будь-якого стану $O(St(i))$, який за деякими значеннями відрізняється від реального опису стану, в який потрапив раціональний агент. Тоді оцінку відкритого стану визначимо як обернену пропорційну величину до відстані між ідеальним описом та реальним описом станів:

$$q(St(i)) = \text{const} / d(O(St(i)), St(i)),$$

де const – коефіцієнт пропорційності, $d(O(St(i)), St(i))$ – відстань між ідеальним описом та реальним описом стану $St(i)$. У цій роботі пропонується метод для обчислення відстані. Розглянемо його детальніше. Спочатку введемо формальне означення онтології. Онтологію будемо визначати як п'ятірку

$$O = \langle X, R, F, W, L \rangle,$$

де \mathbf{X} – скінченна множина концептів (понять, термінів) предметної області, яку задає онтологія \mathbf{O} ; \mathbf{R} – скінченна множина відношення між концептами (поняттями, термінами) заданої предметної області; \mathbf{F} – скінченна множина функцій інтерпретації (аксіоматизація), заданих на концептах або відношеннях онтології \mathbf{O} . \mathbf{W} – важливість понять \mathbf{X} , \mathbf{L} – важливість відношень \mathbf{R} .

Пропонуємо визначати відстань між станом та його онтологією як відстань між їх центрами ваг. Центром ваг може бути одне поняття, два і більше; однак якщо їх три і більше, пропонуємо вибирати перші три. Ця кількість визначена на основі опитувань експертів різних ПО і вважається ними оптимальною. У такому випадку ми маємо три центри ваг ідеального стану і три центри ваг поточного стану. Якщо рахувати всі можливі відстані між ними, то їх буде 9. Тоді вибираємо три найменші з них та їх додаємо. Отримана у такий спосіб сума й буде відстанню між прецедентом та поточною ситуацією. Очевидно, що визначена так відстань залежатиме від того, як ми визначимо відстань між двома суміжними вершинами. Для цього пропонуємо визначати відстані між вершинами, що з'єднані зв'язком як

$$d_{ij} = \frac{Q}{L_{ij}(W_i + W_j)}$$

де W_i та W_j – коефіцієнти важливості вершин X_i та X_j відповідно; L_{ij} – коефіцієнт важливості зв'язку між вершинами; Q – константа, яка залежить від конкретної онтології. Прийmemo, що $L_{ii} = \infty$, тоді $d_{ii} = 0$.

Далі знаходимо центри ваг концептуального графа. Це перші три вершини, для яких середня відстань \bar{d}_i є найменшою:

$$\bar{d}_i = \min \bar{d}_i$$

Середню відстань \bar{d}_i для вершини X_i обчислюємо згідно з формулою:

$$\bar{d}_i = \frac{\sum_{j=1, j \neq i}^n d_{ij}^*}{n-1}$$

де n – кількість вершин графа, d_{ij}^* – найкоротший шлях між вершинами X_i та X_j , який обчислюється за допомогою відомих алгоритмів, наприклад Форда, Дейкстри, Флойда–Уоршалла [5].

Далі згідно з концептуальним графом, що задає онтологію, шукаємо відстань від неї до стану. Якщо поняття поточного стану не входять в концептуальний граф, то онтологію цього стану доповнюємо онтологією всього ІА. Зазначимо, що запропонована відстань задовольняє три аксіоми метрики.

Справді, згідно з визначенням відстані, автоматично виконуються дві перші аксіоми:

$$d(X_i, X_i) = 0;$$

$$d(X_i, X_j) = d(X_j, X_i).$$

Нехай R_{ij}^* – шлях між вершинами X_i та X_j , який відповідає відстані між ними. Тоді $d_{ij} = d_{ik} + d_{kj}$, якщо вершина X_k лежить на шляху R_{ij}^* і $d_{ij} < d_{ik} + d_{kj}$, якщо вершина X_k не лежить на шляху R_{ij}^* . А це означає, що виконується третя аксіома метрики.

Оцінка дій

Для оцінки стану, в який ІА ще не потрапляв, використовуватимемо евристичні функції. Як відомо, загального визначення евристик не існує, а таке визначення евристичних функцій строго прив'язується до предметної області. Тому оцінка майбутніх станів є складною задачею. У наших дослідженнях для вибору дій ІА ми спиратимемось на раціональність агента як прагнення мінімізувати витрати ресурсів для досягнення кінцевого стану. Тому вважатимемо, що дія \mathbf{F}_{ij}

однозначно визначається витратами ресурсів R_{ijk} (ціна переходу зі стану в стан), де $k=1,2,\dots, n_i$. n_i – кількість альтернатив a_k для здійснення переходу F_{ij} .

Розглянемо функціонування ІА на основі оцінки витрат ресурсів. Інформація про альтернативи та необхідний для цього ресурс ϵ у літературних джерелах T .

Кортеж

$$\langle d(O(St(j_i)), St(j_i)), St(j_i), R_{ijk}, o(St(j_i)) \rangle$$

визначає, чи робити нам перехід F_{ij} , використовуючи альтернативу a_k .

Для обчислення необхідних ресурсів R_{ijk} використовуємо онтології літературних джерел $O(T)$ та онтології станів $O(St(i))$. А саме:

$$\langle O(T), O(St(i)) \rangle \rightarrow R_{ijk},$$

$$\langle O(F_{ij}(St(i))), O(St(j_i)) \rangle \rightarrow d(O(St_k(j_i)), St(j_i)).$$

Мета досягнута, якщо

$$d(O(St(z)), St(z)) < v,$$

де v порогова величина, яка залежить від предметної області та ІА. Враховуючи вигляд мережі переходів станів, а саме кон'юнкції дій, можемо стверджувати, що стан $St(j)$ досягнений, якщо $St(j) = F_{j_1j} \wedge F_{j_2j} \wedge \dots \wedge F_{j_mj}$.

Затрати ресурсів для досягнення стану $St(j)$:

$$R(j) = \sum_{i=1}^m R_{j,ik},$$

якщо була використана альтернатива a_k .

Позначимо g – функцію для визначення часових параметрів переходу зі стану в стан. Час для переходу зі стану i в стан j запишемо як:

$$t_{ij} = g(R_{ijk}, R'_{ij}),$$

де R'_{ij} – додаткові ресурси, які необхідні витратити для зменшення часу переходу.

Реалізація моделі

Для реалізації запропонованої моделі мережі вибрано реляційну модель (див. рис. 2).

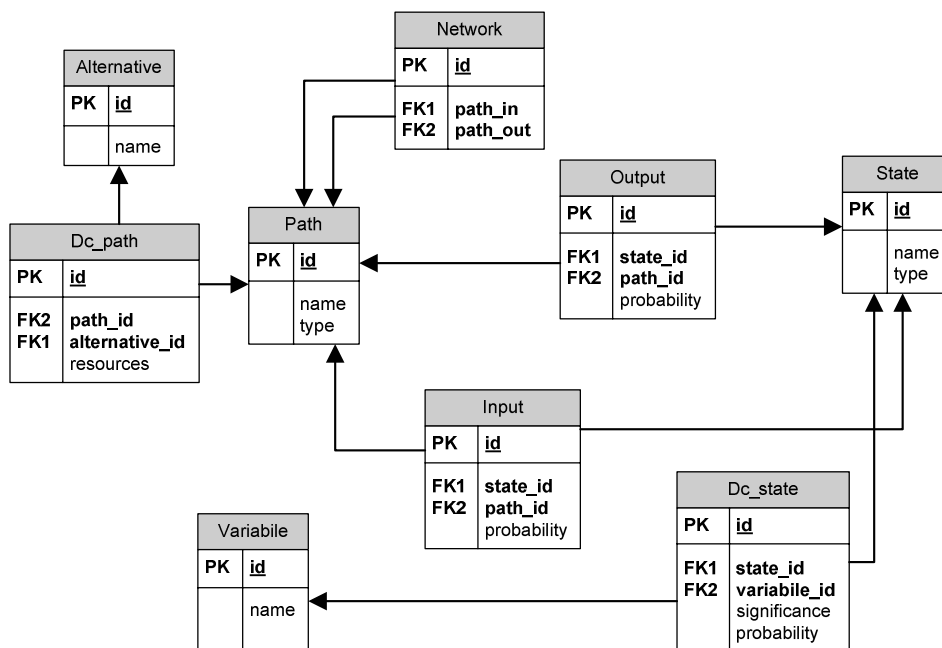


Рис. 2. Реляційна модель плану поведінки інтелектуального агента

Коротко опишемо призначення таблиць:

Alternative – множина альтернатив, які використовуються для вибору дії.

Variable – множина змінних, якими описується стани (входять у онтологію ПО).

Path – шлях, який об'єднує будь-які два стани (ділиться на дві частини: вхід та вихід).

State – множина станів мережі.

Network – мережа, яка задається послідовністю шляхів.

Dc_path – опис шляху, що прив'язує альтернативи з відповідними витратами ресурсів.

Dc_state – опис стану, який визначає значення відповідних змінних з ймовірностями.

Output – множина шляхів, які виходять з деякого стану та ймовірності переходу.

Input – множина шляхів, які входять в деякий стан та ймовірності переходу.

Контрольний приклад

Як приклад розглянемо таку мережу Петрі, яка моделює план досягнення цільового стану **St(z)**.

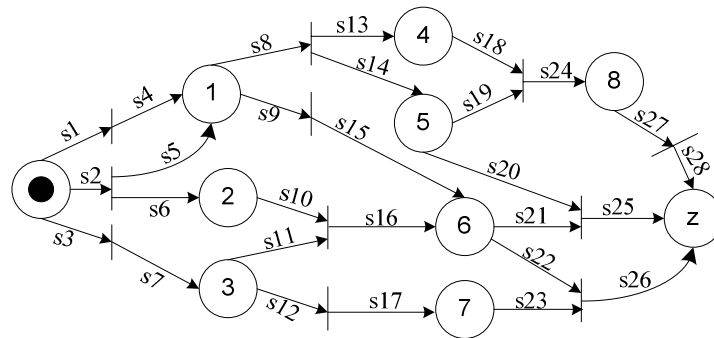


Рис 3. Приклад мережі Петрі моделювання плану діяльності інтелектуального агента

Шляхи **s1, s2, ...** є вихідними, вони зберігаються у таблиці **Output**, а **s4, s5, ...** – вхідними і вони зберігаються у таблиці **Input**. Діям **F_i** надаватимемо такі самі значення індексів, як вихідним шляхам. Так, те, що ІА здійснив дію **F₁₂**, означає, що він пішов по шляху **s12**, якого достатньо, щоб потрапити у 7-й стан.

Таблиця 1

Випадки трансформації мережі Петрі, що моделює діяльність ІА у звичайний граф

№ з/п	Випадок	Мережа Петрі	Граф
1	Один вихід – один вхід		
2	Один вихід – більше від одного входу (для прикладу два входи)		
3	Більше від одного виходу (для прикладу два виходи)		

Проаналізувавши мережу, ми бачимо, що у кінцевий стан **z** ми можемо потрапити із 8-го стану, або одночасно, виконавши правильні дії, з 5-го та 6-го, або одночасно з 6-го і 7-го.

Якщо завдяки онтології ПО та літературним джерелам ми оцінили затрати ресурсів **R_{ij}** для переходу із стану **i** в стан **j**, то мережу на рис. 3 можна трансформувати у граф і задачу вибору шляху розглядати як задачу асинхронного динамічного програмування [6]. Для такої трансформації розглянемо три можливі випадки щодо переходів у мережі Петрі (див. табл. 1).

Тобто якщо є більше ніж один вихід, то дуги, що збираються у вершині, з'єднуємо, щоб вказати, що ми маємо справу з кон'юнкцією.

Зауважимо, що пара станів i та j можуть з'єднуватись не лише однією дугою, тобто індексування ресурсів у вигляді двійки індексів ij є неправильним. Тому для індексації затрат ресурсів використовуватимемо одинарний індекс.

Згідно з описаними перетвореннями отримаємо граф, наведений на рис. 4.

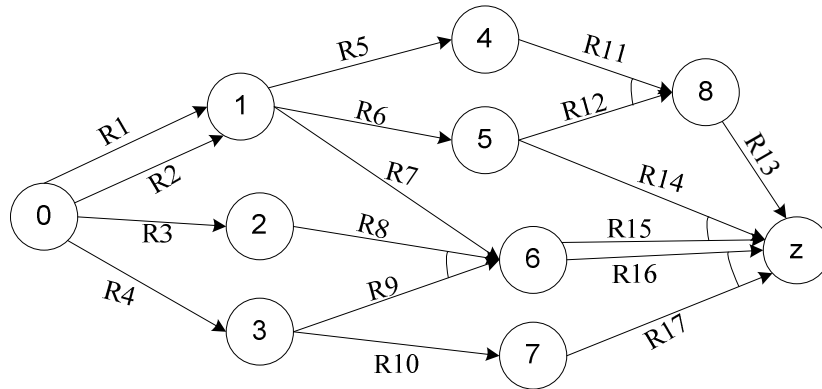


Рис. 4. Приклад графу для задачі асинхронного динамічного програмування

Використовуючи методи, придатні для розв'язування задач асинхронного динамічного програмування, знаходимо розв'язок у вигляді шляху переходу з початкового у кінцевий стан з мінімальними затратами.

Питання побудови онтології ПО (станів) та літературних джерел, щоб оцінити витрати ресурсів, у цій роботі не розглядаються.

Висновки

Розроблено модель планування поведінки інтелектуального агента на основі його раціональної діяльності. Раціональність полягає у прагненні мінімізувати затрати ресурсів для досягнення цільового стану. Побудова моделі ґрунтується на мережах Петрі та використовує онтологію предметної області для оцінки витрат ресурсів та релевантності станів. Показано, що задачу мінімізації витрат ресурсів можна звести до задачі асинхронного динамічного програмування.

1. Нильсон Н. Принципы искусственного интеллекта / Н. Нильсон. – М.: Мир, 1985. – 373 с.
2. Даревич Р.Р. Метод автоматического назначения информационной ваги понятий в онтологии базы знаний / Р.Р. Даревич, Д.Г. Досин, В.В. Литвин // *Відбір та обробка інформації*. – 2005. – Вип. 22(98). – С.105–111.
3. Логика рассуждений и ее моделирование / Под ред. Д. А. Поспелова. – М.: Науч. совет по комплекс. пробл. “Кибернетика” АН СССР, 1983. – 180 с.
4. Рассел С. Искусственный интеллект / С. Рассел, П. Норвиг. – М., С. – П., К.: Вильямс, 2006. – 1408 с.
5. Даревич Р.Р. Оцінка подібності текстових документів на основі визначення інформаційної ваги елементів бази знань / Р.Р. Даревич, Д.Г. Досин, В.В. Литвин, З.Т. Назарчук // *Искусственный интеллект*. – Донецк. – № 3. – 2006. – С. 500–509.
6. Bertsekas D.P. *Dynamic Programming and Optimal Control* / D.P. Bertsekas. – Athena Scientific, 2000. – Vols. 2. – 306 p.