

УДК 004.422.832

ПЕРЕТВОРЕННЯ ДІАГРАМИ СТАНІВ UML У ГІБРИДНУ РЕСУРСНУ МЕРЕЖУ

Павленко А.О.

к.т.н., доцент каф. СПЗ Пригожев О.С.

Одеський національний політехнічний університет, УКРАЇНА

АНОТАЦІЯ. В тезисах описано конвертер діаграми станів *UML* у гібридну ресурсну мережу для системи *CHITS*. Спочатку описаний формат, що використовувався у цій системі для збереження діаграм станів. Потім описаний конвертер. Зроблені висновки щодо доцільності розробки конвертора.

Вступ. Актуальною задачею розробки програмного забезпечення є подальша автоматизація процесів розробки програмного забезпечення, зокрема тестування. Наразі перспективним є застосування систем, що базуються на математичних моделях програм[1]. Тому актуальною є конвертація існуючих форматів та специфікацій у ці моделі.

Мета роботи. Мета роботи полягає у розробці алгоритма конвертера діаграми станів *UML* у гібридну ресурсну мережу, для розширення функціональності модуля конвертації діаграм станів *UML* у вихідний код.

Опис формату. На діаграмі станів *UML* можуть бути присутні такі елементи та їхні атрибути:

1. Стан, з можливими атрибутами: назва (1), дія при вході в стан *entry/* (0..1), дія при виході зі стану *exit/* (0..1), діяльність стану, що зазначається після ключового слова *do/* (0..*), внутрішній перехід (0..).

2. Зовнішній перехід, з можливими атрибутами: подія, що спричинила перехід (0..*), сторожова умова (0..1), дія, що виконується при переході (0..1).

3. Псевдостан, з можливим атрибутом назва (0..1).

Далі показані заємов'язки між елементами та їхніми атрибутами (рисунок 1).

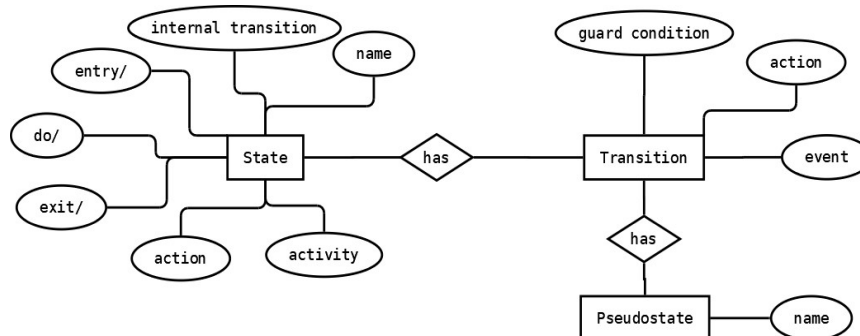


Рис. 1 – Модель сутність-зв'язок для діаграми станів *UML*

Оскільки за формою запису дія при вході в стан, дія при виході зі стану, діяльності стану та внутрішні переходи не відрізняються, то логічно було б представити їх як однаковий атрибут. При цьому важливо зазначити, що це є лише уніфікованою формою запису, яка не змінює семантики цих атрибутів.

Також, псевдостан можна зобразити як стан, і представити їх як один і той самий елемент з різним набором атрибутів.

Так як зовнішній перехід поєднує один стан/псевдостан з іншим, то зовнішній перехід можна внести у якості атрибуту в стан, із якого він виходить.

Діаграма станів описана у вигляді *xml*-файлу. Основним тегом цього *xml* є *<objects>* - це кореневий елемент документа. *<mxCell>* - елемент, що зберігає елемент діаграми. *<state>* - елемент, що описує стан. *<name>* - елемент, що зберігає назву елемента. *<internalTransition>* - елемент, що описує внутрішній перехід, дію при вході в стан, дію при виході зі стану або діяльність після ключового слова *do/*. *<transition>* - елемент, що описує перехід між станами. *<target>* - елемент, що зберігає ідентифікатор цільового стану переходу. *<condition>* - елемент,

що зберігає сторожову умову. $\langle event \rangle$ - елемент, що зберігає подію. $\langle action \rangle$ - елемент, що зберігає дію/діяльність.

Гібридна ресурсна мережа також подається у вигляді *xml*-файлу. $\langle Graph \rangle$ - це кореневий елемент документу. $\langle Vertices \rangle$ - елемент, що зберігає в собі всі вершини графа. $\langle Vertex \rangle$ - елемент, що зберігає інформацію про вершину. $\langle Arc \rangle$ - елемент, що зберігає інформацію про дугу.

Опис алгоритму конвертації. При розробці конвертера на діаграми станів було накладено наступні обмеження: зіставні стани, як прості, так і ортогональні не розглядаються, стан не має внутрішніх переходів, псевдостани не розглядаються.

Далі наведено кроки алгоритму конвертації діаграми станів *UML* у ГРМ:

1. Знайти стан початку $st_{i=0}$ та його стрілку.
2. Знайти стан st_{i+1} , у який відбувається перехід із стану початку, $i=i+1$.
3. Додати дві вершини $v_{j=0}$ та v_{j+1} у ГРМ. Поєднати їх ребром $a_{k=0}$. В операторі ребра a_k вказати, що поточний стан $currentState = st_i, j=j+2, k=k+1$.
4. Якщо стан $currentState$ має дію при вході $entry/$. Додати вершину v_j та ребро a_k , що поєднує v_{j-1} та v_j у ГРМ. В операторі ребра a_k вказати дію при вході в стан $entry/, j=j+1, k=k+1$.
5. Якщо стан має діяльності, вказані після ключового слова $do/$, для кожної діяльності: додати вершину v_j та ребро a_k , що поєднує v_{j-1} та v_j у ГРМ; в операторі ребра вказати діяльність; $n=n+1, k=k+1$.
6. Якщо стан має дію при виході $exit/$, додати вершину v_j та ребро a_k , що поєднує v_{j-1} та v_j у ГРМ. В операторі ребра a_k вказати дію при виході в стан $exit/, j=j+1, k=k+1$.
7. Якщо стан має лише 1 вихідний перехід: додати вершину v_j в ГРМ; поєднати v_{j-1} та v_j ребром a_k ; в умові ребра a_k вказати подію (або диз'юнкцію подій), що спричинює перехід, якщо перехід має сторожову умову, додати вершину v_{j+1} та поєднати її із вершиною v_j ребром a_{k+1} ; якщо перехід має дію, що виконується, додати вершину v_{j+2} та поєднати її із вершиною v_{j+1} ребром a_{k+2} ; знайти стан st_{i+1} , у який відбувається перехід із поточного стану $currentState$; додати вершину v_{j+3} у ГРМ, поєднати v_{j+2} та v_{j+3} ребром a_{k+3} ; в операторі ребра a_{k+3} вказати, що поточний стан $currentState = st_i, j=j+5, k=k+5, i=i+1$; для стану st_i перехід до пункту 4.
8. Інакше, якщо стан має більше, ніж 1 вихідний перехід, для кожного переходу: v_{j-1} є вихідною вершиною для переходу; додати дві вершини v_j та v_{j+1} в ГРМ; v_{j+1} є вихідною вершиною для наступного переходу; поєднати v_{j-1} та v_j і v_{j-1} та v_{j+1} ребрами a_k та a_{k+1} відповідно; в умові ребра a_k вказати подію (або диз'юнкцію подій), що спричинює перехід; в умові ребра a_{k+1} вказати заперечення умови ребра a_k , якщо перехід має сторожову умову, додати вершину v_{j+2} та поєднати її із вершиною v_j ребром a_{k+2} ; якщо перехід має дію, що виконується, додати вершину v_{j+3} та поєднати її із вершиною v_{j+1} ребром a_{k+3} ; знайти стан st_{i+1} , у який відбувається перехід із поточного стану $currentState$; додати вершину v_{j+4} у ГРМ, поєднати v_{j+3} та v_{j+4} ребром a_{k+4} ; в операторі ребра a_{k+3} вказати, що поточний стан $currentState = st_i, j=j+5, k=k+5, i=i+1$; для стану st_i перехід до пункту 4.
9. Інакше, якщо стан має 0 вихідних переходів, кінець алгоритму.

Висновки. Було описано формат *xml*-файлу діаграми станів *UML* для конвертера діаграми станів у внутрішнє представлення системи *CAITS*. Було розроблено алгоритм конвертації діаграми станів *UML* у ГРМ. Завдяки конвертації у внутрішнє представлення, а не у вихідний код напряму, було розширено функціональність модуля конвертації діаграм станів *UML* у вихідний код.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пригожев А. С. Архитектура среды тестирования на основе модели гибридных ресурсных сетей / А. С. Пригожев, Д. А. Неизвестный, О.С. Ларионова // Радиоэлектронні і комп'ютерні системи. - 2016. - № 5. - С. 84–92. - Режим доступу: http://nbuv.gov.ua/UJRN/recs_2016_5_15. - Назва з екрана.
2. Пригожев А.С. Тестирование на основе гибридной ресурсно-сетевой тест-модели программного обеспечения / А.С. Пригожев // Международная научная конференция имени Г.А. Таран «Интеллектуальный анализ информации» ИАИ-2015, Киев, 20–22 мая 2015 г. : сб. тр. – К. : Просвіта, 2015. - С. 172-179. - Режим доступу: - http://pma.fpm.kpi.ua/sites/default/files/iai-2015_proceedings.pdf. - Назва з екрана.