

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ ШТУЧНОГО ІНТЕЛЕКТУ ТА РОБОТОТЕХНІКИ
КАФЕДРА ПРОГРАМНИХ І КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ

Методичні вказівки з дисципліни
ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ
(Теоретична частина. Частина друга)
Для студентів інституту штучного інтелекту та робототехніки

Перший (бакалаврський) рівень вищої освіти

Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма: Комп'ютерні технології автоматизації;

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ ШТУЧНОГО ІНТЕЛЕКТУ ТА РОБОТОТЕХНІКИ
КАФЕДРА ПРОГРАМНИХ І КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ

Методичні вказівки з дисципліни
ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ
(Теоретична частина. Частина друга)
Для студентів інституту штучного інтелекту та робототехніки

Перший (бакалаврський) рівень вищої освіти

Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма: Комп'ютерні технології автоматизації;

Схвалено на засіданні кафедри ПКІТ
протокол №7 від 26.01.2022 р.

Одеса 2022

Методичні вказівки з дисципліни «Обчислювальна техніка та комп'ютерні технології» (Теоретична частина. Частина друга): для студ. спец. 151 – «Автоматизація та комп'ютерно-інтегровані технології» ден. та заоч. форм навч./уклад.: В.О. Давидов - Одеса: ОП, 2022. - 15 с.

Укладачі: **В.О. Давидів**, канд. техн. наук

Зміст

4	Динаміка зв'язків	5
4.1	Динаміка зв'язків	5
4.2	Життєві цикли зв'язків	5
4.3	Динамічні зв'язки поза життєвими циклами	5
4.4	Конкуруючі зв'язки: монітор	5
4.5	Загальний випадок конкуруючого зв'язку	6
4.6	Конкуруючі зв'язки із життєвими циклами екземплярів	7
4.7	Поради щодо моделювання	7
5	Динаміки систем	8
5.1	Модель взаємодії об'єктів	8
5.2	Ієрархічне уявлення об'єктів	8
5.3	Схеми взаємодій	9
5.4	Канали керування	10
5.5	Імітування	11
5.6	Як розглядати час	13
5.7	Багаторазове управління	14

4 ДИНАМІКА ЗВ'ЯЗКІВ

4.1 Динаміка зв'язків

Відносини між деякими предметами у світі розвиваються у часі за певними правилами поведінки, які у різних точках протягом відносини.

Коли ми формалізуємо ставлення у реальному світі як зв'язок на інформаційній моделі, ми зберігаємо лише статичний аспект відносини: чи має воно чи може існувати? Те, що ми не зберігаємо, є динамікою зв'язку.

Як зв'язок змінюється у часі?

За яких умов створюється чи знищується екземпляр зв'язку?

Як вибираються екземпляри об'єктів для участі у зв'язку?

На питання такого типу відповідають або моделі станів об'єктів, що беруть участь, або окремі моделі станів для самих зв'язків. У цьому розділі ми розглядаємо різні компоненти динамічної поведінки зв'язку та показуємо, як ця поведінка зберігається у моделях станів.

4.2 Життєві цикли зв'язків

У деяких випадках відношення між предметами реального світу еволюціонує через чіткі фази, у яких застосовуються різні правила та лінії поведінки. І тут відношення абстрагується як зв'язок у вигляді асоціативного об'єкта з атрибутом поточного стану. Надалі модель станів формується для формалізації життєвого циклу зв'язку.

Модель станів наказує поведінка типового невизначеного екземпляра зв'язку. Всі екземпляри зв'язку виконують ту саму модель станів. Як і у випадку з об'єктом, ми використовуємо термін "кінцевий автомат", щоб вказувати виконання моделі станів приватним екземпляром зв'язку.

4.3 Динамічні зв'язки поза життєвими циклами

Якщо навіть зв'язок не має життєвого циклу, вона може мати динамічну поведінку за умови, що екземпляри зв'язку створюються або знищуються в межах тимчасової області аналізу. Наприклад, у банківських додатках зв'язок між клієнтом та його рахунком постійна і немає стадій, у яких застосовувалися різні правила. Однак кожного разу, коли клієнт або рахунок з'являється або зникає, екземпляр Клієнт володіє Рахунком повинен бути створений відповідно або видалений.

4.4 Конкуруючі зв'язки: монітор

Конкуренція у реальному світі

У світі може існувати конкуренція, що з встановленням зв'язку. У прикладі Клієнт-Пункт Замовлення клієнти не повинні конкурувати один з одним, щоб зробити Пункт Замовлення. Дистриб'ютор може прийняти чимало замовлень. Подібним чином у прикладі з банком існує нескінченний запас нових рахунків, і будь-яка вимога клієнта про відкриття нового рахунку може бути задоволена.

На протигагу цьому розглянемо ситуацію, в якій клієнти (пацієнти, пасажери тощо) конкурують з метою опанувати увагу обмеженої кількості клерків (лікарів, офіціанток або стюардес). Коли клерк стає доступним, клієнти, що не обслуговуються, намагаються заволодіти увагою клерка, щоб сформувати екземпляр зв'язку один-до-одному Клієнт обслуговується Клерком. Оскільки клієнти, що розглядаються тепер як екземпляри ООА, діють одночасно, можливо, що два клієнти звернуться до клерка в той самий час.

Проблеми конкуренції такого типу необхідно вирішувати наданням однієї контрольної точки, при проході якої конкуруючі запити перетворюються на послідовну форму. У реальному світі це перетворення на послідовну форму може бути досягнуто різними механізмами: чергою клієнтів (прояв стратегії, що вимагає, щоб клієнти обслуговувалися в порядку їх появи) або людиною, яка визначає клієнтів для клерків на підставі якоїсь іншої стратегії (наприклад, у кімнаті невідкладної допомоги лікарні серйозність становища " клієнта " повинна, ймовірно, бути домінуючим міркуванням). В ООА перетворення на послідовну форму виконується спеціальною моделлю станів для зв'язку, у якій є конкуренція.

Конкуренція в ТОВ

Коли ми маємо справу із зв'язком, у якому існує конкуренція, нам необхідно спочатку формалізувати зв'язок як асоціативний об'єкт на інформаційній моделі. Надалі ми формуємо модель станів, яка відповідає за створення екземплярів зв'язку при взаємодії між собою екземплярів об'єктів, що беруть участь. Асоціативний об'єкт надає ім'я цієї моделі станів.

Визначник <ім'я асоціативного об'єкта>

Події, спрямовані на цю модель стану, позначаються так:

<ключовий літерал асоціативного об'єкта>-А.

Оскільки мета Визначника полягає у тому, щоб діяти як єдина точка контролю, при проходженні якої конкуруючі запити перетворюються на послідовну форму, існує лише один зразок цієї моделі станів. Поточний стан Визначника зберігається в моделі станів і не з'являється як атрибут на інформаційній моделі.

Очевидно, що монітор не бажає вибирати клієнта X знову і перевіряє екземпляри обслуговування, щоб переконатися, що клієнт ще не був обраний.

Зверніть увагу, що події для монітора не несуть дані ідентифікатора. Оскільки існує лише один зразок кінцевого автомата монітора, не потрібно ідентифікувати екземпляр.

Вибір екземпляра

У прикладі Клієнт-Клерк кінцевий автомат, що управляє зв'язком, робить вибір між клієнтом та клерком довільно. У типовому випадку, однак, буде потрібна комплексна стратегія вибору. Для забезпечення бажаної лінії поведінки необхідно, щоб для об'єктів, що беруть участь у зв'язку (або інших близьких об'єктів) були визначені додаткові атрибути. Наприклад, якщо Ви хочете визначити клієнтів до клерків, які обслужили найменшу кількість клієнтів до поточного моменту, атрибут Клерк. Кількість Обслуговуваних Клієнтів слід додати до Клерка. В якості альтернативи стратегія пошуку могла брати до уваги час приходу клієнта (Клієнт. Час Приходу), його вік або те, як довго він мав справу з організацією, яка забезпечує обслуговування.

4.5 Загальний випадок конкуруючого зв'язку

Побудова моделі станів для створення екземплярів конкуруючого зв'язку зазвичай потребує певної обережності, проста форма монітора безпосередньо придатна лише у певних ситуаціях. Цей розділ є систематичною процедурою для побудови необхідної моделі станів Визначника в загальному випадку. Процедура ілюструється з прикладу готелю з кінцевим числом кімнат. Постояльці конкурують один з одним під час бронювання кімнат на різні терміни.

1-й крок: інформаційна модель

Як завжди, спочатку ми формуємо інформаційну модель. Конкуруючий зв'язок має бути формалізований як асоціативний об'єкт. Будь-які атрибути, необхідні для того, щоб забезпечити необхідну стратегію вибору, мають бути наведені у відповідних об'єктах. Інформаційна модель проблеми бронювання кімнати показана на рис.4.5.1.

Атрибути Постояльця та Кімнати забезпечують стратегію вибору, згідно з якою найчастішим клієнтам надаються спокійні (безшумні) кімнати.

2-й крок: моделі станів для об'єктів, що беруть участь.

Формуємо моделі станів обох об'єктів, що у зв'язку. Ідентифікуємо всі події моделей станів об'єктів, які очікуємо отримати із моделі станів Визначника.

3-й крок: модель станів Визначника, перший малюнок

Побудуйте перший проект моделі станів Визначника у вигляді монітора. Мета цієї версії моделі станів полягає в тому, щоб розібратися в логіці взаємодіючих подій, тому необхідно включити подробиці процесу вибору, який містить знаходження екземплярів, визначених для участі у зв'язку. Проте ті аспекти процесу пошуку, які вибирають серед уже певних екземплярів (час приходу клієнта, визволення кімнати тощо), мають бути опущені.

4-й крок: коригування першого начерку

Дослідіть перший малюнок моделі Визначника, щоб побачити, чи він вдалий, а якщо ні, то чому. Переконайтеся, що події, що очікуються об'єктами, що беруть участь, породжуються.

На першій, чорновій моделі (рис.4.5.3), можливо, Ви змогли потрапити в стан 3 і виявити, що всі постояльці хотіли забронювати кімнати на ті дні, коли готель був повний, і що були кімнати, не заброньовані на інший час. І тут дію було бути виконано.

5-й крок: коригування моделей станів об'єктів, що беруть участь

Щоб зробити моделі станів об'єктів, що не суперечать моделі Визначника, зазвичай потрібні незначні коригування. Зокрема, переконайтеся, що події, породжені об'єктами, що беруть участь, несуть усі (і тільки всі) дані, що фактично використовуються моделлю Визначника. Також може статися, що деякі події не є необхідними. Приміром з готелем Кімната Термін не є більш необхідною для створення Б-А2: Кімната вільна.

4.6 Конкуруючі зв'язки із життєвими циклами екземплярів

4.7 Поради щодо моделювання

Ось деякі специфічні пропозиції, які можна знайти корисними при побудові моделей стану як для об'єктів, так і для зв'язків.

Отримайте правильну інформаційну модель. Досвід вказує, що якщо при побудові моделей станів у Вас виникли значні проблеми, ймовірно, щось не в порядку з інформаційною моделлю. Найкраща стратегія тоді - повернутися до інформаційної моделі та подивитися, які додаткові здібності щодо проникнення в суть Ви набули при розробці даної моделі станів. Часто буває досить незначне коригування інформаційної моделі.

Зберігайте контроль над подіями. Ніщо так не збиває з пантелику, як набір моделей з неузгодженими мітками подій, різними значеннями та невизначеними даними події. Зазвичай краще покладатися на список подій як на офіційне визначення подій і створювати модель станів, що відповідає списку подій.

Стережіться пасток монітора. Конкуруючі зв'язки можуть бути настільки складними, що без багаторазового уявлення специфічного становища легко потрапити в пастку опис форми монітора для Визначника моделі станів.

Використовуйте модель взаємодії об'єктів. Модель взаємодії об'єктів (описана у 5-му розділі) є графічним уявленням взаємодії подій між різними об'єктами та зв'язками. Якщо Ви заздалегідь робите ескіз грубої моделі взаємодії об'єктів і підтримуєте його в міру реалізації, навряд чи виникнуть якісь несподіванки, пов'язані з подіями.

5 ДИНАМІКИ СИСТЕМ

5.1 Модель взаємодії об'єктів

Модель взаємодії об'єктів, або МВО забезпечує короткий графічний виклад подій взаємодії між моделями станів і зовнішніми сутностями, такими як оператори, фізичні пристрої та об'єкти в інших підсистемах. Кожна модель станів представляється МВО вирівняним овалом, позначеним ім'ям моделі станів. Кожна зовнішня сутність, яка може породжувати або приймати події, є прямокутником, відомим як термінатор. Подія, породжена однією моделлю станів або зовнішньою сутністю і прийнята іншою, позначена стрілкою від компонента, що породжує, до приймача. Стрілка анотується міткою події, значенням та (необов'язково) даними події.

Проста МВО показана на рис.5.1.1. Зверніть увагу, що за згодою події, що породжуються або приймаються таймерами, опускаються з МВО, оскільки досвід показує, що такі події мало сприяють розумінню системи як єдиного цілого. З тієї ж причини ми опускаємо події, які породжуються та приймаються тією самою моделлю станів.

Хоча МВО може бути побудована безпосередньо з моделі станів, ми вважаємо, що простіше працювати зі списком подій, витягуючи всі окремі записи, які мають джерело і призначення події різні.

5.2 Ієрархічне уявлення об'єктів

Інтелект, контекст та ціль

На прикладі мікрохвильової печі Ви могли спостерігати, що життєвий цикл печі якісно відрізнявся від життєвих циклів силового елемента та електричної лампочки. Мета життєвого циклу печі полягала в тому, щоб скоординувати роботу силового елемента та електричної лампочки, щоб цей набір обладнання діяв як мікрохвильова піч. В результаті модель станів печі має такі якості, як інтелект, контекст та ціль. Навпаки, моделі станів силового елемента та електричної лампочки мають слухняний характер, сумісний з їх обмеженою поінформованістю про систему як єдине ціле.

У більшій моделі цю схему спостерігають послідовно. Об'єкти з великим інтелектом та метою делегують роботу менш інтелектуальним об'єктам та координують їх життєві цикли, у міру того, як виконується робота. Це добре показано з прикладу підприємства з виробництва соку [1]. Схематичне зображення частини підприємства з виробництва соків представлено на рис.5.2.1, а відповідна інформаційна модель наводиться на рис.5.2.2. Модель взаємодії об'єктів показано на рис.5.2.3.

Вона створює іншу Зміну Температури, щоб витримати сік при температурі пастеризації протягом визначеного часу (подія TR9O: Здійснити зміну температури). І нарешті, вона створює об'єкт процесу консервації, для переміщення порції соку до лінії консервування та упаковки його для постачання споживачеві. При такому делегуванні Порція зобов'язана знати подробиці того, як впливати обладнання, щоб виконати необхідний виробничий процес. Подібним чином об'єкт Переміщення Соку повторює цю схему делегування та координації. Спочатку він вимагає, щоб Шлях 1 По Трубам був встановлений між баком для зберігання та баком для приготування (подія PP1: Призначити шлях). Далі він просить Шлях По Трубам запустити якісь насоси, що знаходяться на ділянках колії (подія PP2: Почати перекачування).

Об'єкт Порція є порцією соку починаючи з того моменту, коли складається план для його виробництва, і доти, доки він не буде поміщений у консервні банки для продажу.

Підтримкою Порції є об'єкти, що становлять стадії виробництва: Переміщення Соку, Зміна Температури та Процес Консервування. Примірник Порції робить себе делегуванням.

Модель станів Порції створює один або більше об'єктів Переміщення Соку, щоб змусити сік пуститися трубами в Бак Для Приготування (подія JT1: Здійснити транспорт соку).

Потім вона створює об'єкт Зміна температури таким чином, що сік буде повільно нагрітий до температури пастеризації (подія TR9O: Здійснити зміну температури).

Коли потрібну кількість соку переміщено, Переміщення Соку просить Шлях По Трубам вимкнути насоси і потім знову повідомляє Порції, що процес завершено (подія В21: Переміщення завершено).

Об'єкт Шлях По Трубам продовжує тему приховування інформації від інтелектуальнішого об'єкта, чийх інтересах він діє. Шлях По Трубам намагається визначити безперервний маршрут трубами від джерела до адресата (визначених у термінах баків). Для того, щоб ізолювати шлях, об'єкт закриває всі клапани, розташовані між трубами, які становлять цей шлях, і трубами, що не мають до нього відношення. Він звільняє шлях для соку, відкриваючи всі вентиля, які з'єднують дві труби у дорозі, і, нарешті, відкриває ті клапани, які пов'язують шлях із двома баками.

З іншого боку, модель станів об'єкта Шлях По Трубах не знає (чи не повинна знати), чому це відбувається. Вона реагує на вимоги об'єкта, який має великі знання та контекст: у цьому випадку це або Переміщення Соку, або Процес Консервування.

Ієрархічна вистава на МВО

За угодою МВО планується з найбільш добре обізнаними та потужними об'єктами, розташованими вгорі сторінки, та з менш обізнаними об'єктами, що знаходяться нижче. Це забезпечує приблизне ієрархічне уявлення об'єктів, яке суперечить лініям поведінки взаємодії подій, необхідним системою.

Найнижчий рівень. На найнижчому рівні ми розміщуємо об'єкти з обмеженим знанням про мету системи. Ці об'єкти, як і Клапан, просто покійно реагують на події. Об'єкти на найнижчому рівні відповідають виконавцям у термінології проектування. У системах реального часу такі об'єкти служать у тому, щоб робити невидимими подробиці сигнального інтерфейсу для апаратних засобів, вони відповідають парнасовским віртуальним пристроям.

Найвищий рівень. На найвищому рівні ми поміщаємо об'єкти, які містять призначення та мету системи.

Ці об'єкти породжують події, які проводять об'єкти нижчого рівня через їх життєві цикли, і отримують події знизу, коли екземпляр нижчого рівня досяг певного стану. Об'єкти на найвищому рівні відповідають об'єктам, що впливають у трмінології об'єктно-орієнтованого проектування.

Середні рівні. Середні рівні МВО складаються з об'єктів, що мають відношення до підпроцесів, взаємодії людей або обладнання, корисним змінам нижчих об'єктів і ролям об'єктів. Об'єкти середнього рівня іноді розглядають як посередників, оскільки вони діють під впливом інших об'єктів: коли об'єкт середнього рівня приймає подію зверху, він зазвичай породжує подію інших об'єктів нижче його.

5.3 Схеми взаємодій

Незважаючи на те, що будь-який компонент на МВО може в принципі взаємодіяти з будь-яким іншим компонентом, спостерігають лише невелику кількість схем взаємодій. Щоб описати ці схеми, нам необхідно спочатку бачити різницю між типами подій.

Типи подій

Зовнішня подія— це подія, що породжується зовнішньою сутністю (термінатором) та приймається моделлю станів у системі. Існує два види зовнішніх подій.

Незапитана подія— зовнішнє подія, яке було викликано деяким попереднім дією системи. Наприклад, В1: Створити порцію в баку - не запитуване: воно породжується оператором, коли хоче створити порцію. Система не спонукає оператора робити це і, отже, не має жодного попередження про настання події.

Подія, що запитується – зовнішню подію, яка породжена у відповідь якусь попередню діяльність системи. Наприклад, якщо команда (подія) про закриття надіслана на фізичний вентиль, подія у відповідь "клапан зараз закритий" - це подія, що запитується.

Внутрішня подія – це подія, що породжується моделлю станів усередині системи.

Схема верхнього керування

Об'єкт верхнього рівня приймає незапитане подія від операторів (або подібних інтелектуальних зовнішніх об'єктів), щоб ініціалізувати значущі впливи системи як єдиного цілого. У цій ситуації об'єкт верхнього рівня, ймовірно, працює подібно до об'єкта Порція, делегуючи роботу вниз по МВО за допомогою внутрішніх подій. Поки процес триває, об'єкти як верхнього, і середнього рівня можуть запитувати в оператора додаткову інформацію. У цьому випадку об'єкти верхнього та середнього рівня прийматимуть запитувані події, які дозволять їм продовжувати роботу.

Схема нижнього керування

Якщо подія, що не запитується, спрямована на об'єкт нижнього рівня, приймаючий об'єкт зазвичай має достатній контекст, для того щоб визначити відповідну реакцію. Ймовірно, об'єкт нижнього рівня додасть деяку інформацію до даних події і створить внутрішню подію для об'єкта(ів) вищого рівня(ів). Це буде продовжуватися доти, доки не буде досягнуто об'єкта з достатніми знаннями та контекстом. У цій точці може бути визначено, що необхідно виконати для того, щоб система реагувала відповідним чином. Потім приймаючий об'єкт делегуватиме роботу вниз

5.4 Канали керування

Поняття каналу керування

Канал керування- послідовність дій та подій, які відбуваються у відповідь на надходження деякої незапитаної події, коли система перебуває у певному стані. Канал управління - важлива для аналітика концептуальна сутність, яка описує, що робиться в системі, якщо незапитане подія відбувається, коли вона перебуває у певному стані.

Канал управління може включати діяльність, що виходить за рамки системи: в якомусь місці каналу подія може породжуватися для термінатора, змушуючи його виконувати деяку зовнішню діяльність. Якщо термінатор реагує на подію, що запитується, то ця подія, а також зовнішню діяльність, вважають частиною каналу управління.

Якщо дія вздовж каналу управління породжує більше, ніж одна подія, канал управління розчленовується так, що дві або більше гілки того ж каналу управління активні в один і той же час.

Кожна гілка каналу управління зрештою завершується. Це може статися одним із трьох способів.

1. Досягнуто дію, яка не породжує жодних подій, тоді гілка завершується цією дією.
2. Досягнуто дію, яка породжує подію для термінатора, і він не реагує на подію, що запитується. І тут діяльність завершується поза межами системи.
3. Досягнуто дію, що породжує подію звільнення керованих ресурсів. Тоді гілка завершується цією дією. Наступна діяльність тепер може відбуватися інакше каналу управління, що затримувалося очікуванням доступності керованих ресурсів.

Схема каналу керування

Кожен екземпляр з'являється окремо як рядок станів, який він займає у той час, поки канал еволюціонує. Стани, які займають окремі екземпляри, з'єднуються стрілками, кожна позначається подією, що викликає перехід до наступного стану. Якщо екземпляр породжує подію іншого кінцевого автомата і це подія викликає перехід, стрілка малюється стану, яке створила подія, до переходу приймаючого екземпляра. Діаграма планується вздовж відносної осі часу зі станами, поміщеними на діаграмі у порядку (згори до низу), у якому досягаються.

Час та канал управління

Схема каналу управління може використовуватися для аналізу часу, необхідного системі, щоб відреагувати на подію, що не запитується. Для детальнішого обговорення нам спочатку потрібна деяка термінологія.

- Час, протягом якого екземпляр займає стан, складається з часу дії та часу затримки.
- Час дії - це час, необхідний виконання дії.
- Час затримки – це час, протягом якого екземпляр залишається у цьому стані після завершення дії.

Для опису часу каналів керування на схемі

- анують кожне стан його часом дії;
- якщо час затримки стану визначається виключно екземпляром та станом, анують перехід із цього стану з властивим часом затримки;
- якщо час затримки стану визначається взаємодією між кінцевими автоматами, символ конденсатора поміщають на переході з стану.

Застосування

Схема каналу управління найбільш корисна як інструмент розуміння взаємодій між невеликою кількістю екземплярів. Однак, якщо кількість екземплярів зростає, досягнення зрозумілого розташування схеми дуже важко. Крім того, схема виявляється неконтрольовано великою. Альтернативним підходом розуміння каналу управління є імітування: метод, який виконує взаємодії кінцевого автомата, швидше, у часі, ніж у просторі, як показано на цій схемі.

5.5 Імітування

Щоб визначити, яким чином система як єдине ціле відповідно реагує на незапитану подію, можна відстежувати канал управління за допомогою імітування. Потрібна стратегія, що складається з трьох кроків:

1. Встановити початковий стан системи.
2. Прийняти зовнішню подію, що не запитує, і виконати канал управління, щоб побачити, що відбувається.
3. Оцінити кінцевий результат, якщо він коректний.

Встановлення стану системи

Закінчений стан системи визначається значеннями всіх атрибутів всіх примірників у системі. З метою розгляду деякого каналу управління ми маємо встановити значення тих атрибутів, які впливають в розвитку каналу. Такі атрибути бувають двох типів: атрибути поточного стану (визначають, яка дія виконується, коли подія спрямована до кінцевого автомата) та визначальні атрибути.

Визначення. Логіка дії може бути такою, що події, що породжуються цією дією, залежить від значень атрибутів, відмінних від атрибуту поточного стану. Такі атрибути відомі як визначальні, оскільки визначають, як розвивається канал управління, що він проходить через дію.

За допомогою систематичної процедури необхідні величини атрибутів встановлюють таким чином:

1. Використовуючи МПО та моделі станів, без дотримання формальностей намічають канал управління для визначення того, які моделі станів необхідно розглянути та скільки потрібно різних екземплярів кожного об'єкта. Вибирають ідентифікатори для необхідних екземплярів і встановлюють значення їх атрибутів поточного стану.

2. Аналізують кожну модель станів, яку розглядають у каналі управління для знаходження всіх визначальних атрибутів. Встановлюють значення визначальних атрибутів.

3. З'ясовують, як отримано кожен визначальний атрибут. Якщо він обчислюється моделями стану, що розглядаються в каналі управління, встановлюють величини для всіх атрибутів, що беруть участь у обчисленні.

4. Перевіряють, чи не суперечать одне одному вибрані значення атрибутів:.

Виконання імітування

Як тільки потрібні значення атрибутів встановлені, можна розпочинати відстеження каналу керування за допомогою імітатора. Якщо Ви не маєте доступу до автоматичного імітатора, то може бути використана канцелярська процедура, описана нижче.

Застосовують форму зберігання сліду сутностей, які беруть участь у імітуванні. Вам потрібен один зразок форми для кожного кінцевого автомата (об'єкта, визначника або таймера), що розглядається в каналі управління. Крім того, кожному термінатору, що бере участь у каналі управління, буде потрібна копія.

Початок каналу. Візьміть незапитану подію, яка починає канал управління, і введіть його мітку в 2-й стовпець кінцевого автомата, до якого вона спрямована. Введіть дані події у 3-му стовпці.

Опрацювання події. Для обробки події досліджуйте модель станів для того, щоб знайти відповідний перехід, заданий значенням атрибута поточного стану екземпляра. Існує три можливі варіанти.

1. Якщо подія ігнорується, запишіть ім'я в даний час займаного стану в 4-му стовпці рядка, що містить подію, яку Ви обробляєте. Помістіть галочку-мітку в 1 стовпці, що вказує, що ця подія була оброблена.

2. Якщо подія не може статися, то ми знайшли помилку в якійсь моделі станів. Припиніть імітування, усуньте помилку та спробуйте знову.

3. Якщо подія викликає перехід, запишіть ім'я нового стану в 4 стовпці. Потім виконайте в думці дію, пов'язану зі станом. Відобразіть будь-які зміни у значеннях визначальних атрибутів - на цій або будь-якій іншій таблиці - записом нових значень атрибутів у екземплярах, що впливають.

Запишіть події (мітки та дані події), породжені дією в стовпцях, призначених для цієї мети, використовуючи стільки рядків, скільки потрібно для запису всіх породжених подій.

І нарешті, помістіть галочку-мітку в 1-му стовпці, найближчому до мітки події, що впливає, щоб вказати, що Ви повністю завершили його обробку.

Розповсюдження. Тепер візьміть кожну подію, щойно породжену, і скопіюйте її в 2-й і 3-й стовпці кінцевого автомата або термінатора, до якого вона спрямована.

Вибір та обробка наступного входу. Тепер виберіть перший необроблений окремий запис із будь-якої форми. Якщо форма використовується для відстеження кінцевого автомата, обробіть подію, як описано раніше, та зробіть подальше розповсюдження.

Якщо форма відстежує термінатор і він не створює подію, позначте факт "події, що не породжуються" в 5-му стовпці. Якщо термінатор створює подію, що запитується, запишіть цю подію внизу як породжену подію для термінатора. У будь-якому випадку помістіть галочку-мітку в 1-му стовпці і поширіть подію, якщо така є, як описано раніше.

Завершення. Продовжуйте таким чином імітацію, багато разів повторюючи:

- вибір першого необробленого окремого запису у будь-якій таблиці,
- обробку окремого запису,
- поширення подій або результатів таймера, якщо такі є,

доки всі окремі записи у всіх таблицях нічого очікувати оброблені. У цій точці імітування закінчується.

Оцінка

Результати імітування надаються:

- кінцевим значенням усіх атрибутів поточного стану;

- кінцевим значенням всіх визначальних атрибутів (включаючи атрибути, що беруть участь у обчисленні визначальних атрибутів);
- послідовністю подій, спрямованих до кожного термінатора.

Оцініть результати і переконайтеся, що всі значення атрибутів не суперечать одне одному і створена система з бажаною реакцією у відповідь як наслідком прийнятої початкової незапитуваної події.

5.6 Як розглядати час

Інтерпретації паралелізму

Більшість програмістів-практиків розглядають час двома різними способами, один із яких ґрунтується на нашому звичайному щоденному досвіді, а інший на тому, як обробляється час у середовищі багатозадачного режиму. Фундаментальною відмінністю є паралелізм: як інтерпретувати поняття, що полягає в тому, що дві речі відбуваються паралельно?

ООА допускає дві інтерпретації паралелізму. Одночасна інтерпретація означає, що дві речі можуть відбуватися в той самий час саме так, як у повсякденній дійсності: я можу розмовляти по телефону в той же час, коли листоноша опускає пошту в мою поштову скриньку.

Чергована інтерпретація означає, що тільки одна річ може статися будь-якої миті. Для досягнення ефекту одночасного процесу два або більше процесу повинні бути розділені на підпроцеси, і ці підпроцеси виконуються, наприклад, так: спочатку говорить одна людина з двох учасників телефонної розмови, потім листоноша опускає конверт у поштову скриньку, далі говорить другий, після чого листоноша опускає інший конверт тощо. Це погляд на паралелізм, яка підтримується багатозадачними операційними системами.

Правила часу ТОВ

Формалізація в ООА робить певні твердження та припущення щодо часу, згодом звані правилами часу. Більшість правил часу однакові як одночасної, так чергуючої інтерпретації часу. Деякі правила повинні бути встановлені дещо по-іншому, залежно від того, яку інтерпретацію часу Ви дотримуетесь. Нижче представлені та розглянуті правила часу для обох інтерпретацій.

Правила щодо дій

1. Тільки одна дія даного кінцевого автомата може виконуватись у якійсь точці часу.
2. Дія вимагає часу для виконання. Внаслідок цього ООА відрізняється від багатьох методів аналізу, які припускають, що дії відбуваються миттєво.

3-а. Паралелізм для одночасної інтерпретації: Дії у різних кінцевих автоматах можуть виконуватись одночасно.

3-б. Паралелізм для інтерпретації, що чергується: тільки одна дія може виконуватись в якійсь час. Якщо дію отримує керування, вона виконується до завершення без переривання. Події у різних кінцевих автоматах чергуються довільно. Іншими словами, дія є одиницею чергування.

4. Будучи ініціалізованим, дія кінцевого автомата має завершитися, перш ніж він зможе прийняти іншу подію.

Обов'язок аналітика

— гарантувати, що ця дія буде завершена. в одночасній інтерпретації часу ніщо більше не може колись статися в цьому кінцевому автоматі, тому що він не може прийняти жодних подальших подій;

— в інтерпретації часу, щойно, як тільки це дію отримує управління, ніщо не може більше відбутися у всій системі, оскільки не можуть бути прийняті події, здатні змусити цю дію передати управління.

У інтерпретації, що чергується, ця дія ніколи не завершиться (якщо атрибут Запит Оператора. Прапор вже не встановлений до ініціювання дії), тому що ніяка інша дія не може бути виконана, для того щоб встановити прапорець, який дозволив би В йому завершитися, як

передбачалося. В одночасній інтерпретації дія може завершуватися, тому вона допустима. Однак мало ймовірно, щоб нагрівач вмикався і вимикався так швидко, як це зазначено у цій дії. (Див. Розділ 5.7 для більш прийняттого рішення).

Правила несуперечливих даних

1. Коли дія завершується, вона повинна залишати систему несуперечливою або шляхом запису даних для створення несуперечливої картини, або породженням подій, що змушують інші кінцеві автомати прийти у відповідність до змін даних, зроблених відправником подій. Зверніть увагу, що система може знадобитися певний термін для того, щоб стати повністю несуперечливою при іншій інтерпретації часу. Навіть у одночасно чинному реальному світі потрібен час для поширення інформації про досягнення несуперечливості. Наприклад, коли Ви отримуєте банківський бюлетень, можливо, що сальдо, показане на ньому, відрізняється від записаного у Вашій чековій книжці. Це пояснюється тим, що потрібен час для того, щоб чеки та вклади досягли банку та знайшли відображення у сальдо рахунку. У процедурі,

2. Обов'язок аналітика – гарантувати, що дані, необхідні дією, несуперечливі або що дія передбачає суперечливість, зумовлену часом поширення. Ми вже бачили це на простому прикладі Клієнт-Клерка. Нагадаємо, що коли кінцевий автомат Визначника обслуговування входив у стан 1, він шукав клієнта зі станом "Чекання вільного клерка" і не виявлявся в екземплярі обслуговування саме з цієї причини.

Правила подій

1. Подія ніколи не зникає: кожна подія досягне кінцевого автомата або термінатора, до якого вона спрямована.

2. Подія використовується, коли приймається кінцевим автоматом, потім зникає як подія та не може повторно використовуватись.

Це аналогічно більшості механізмів завдань взаємодії, у яких повідомлення може бути отримано двічі. Щойно повідомлення прийнято, воно зникає як повідомлення.

3. Коли подія породжується, вона відразу стає доступною кінцевому автоматом або термінатору, якого направлено.

4. Коли екземпляр завершує дію, то вважається, що він тепер у новому стані. Тоді

- екземпляр зажадає доступну подію, спрямоване щодо нього, якщо взагалі такі події існують в. цей час (одночасна інтерпретація);
- в якийсь невизначений час після завершення дії кінцевий автомат прийме нову доступну подію, якщо така існує (що чергується інтерпретація).

5. Багаторазові події можуть виступати в ролі тих, що чекають обробки даним кінцевим автоматом.

При одночасній інтерпретації часу кілька кінцевих автоматів могли б породжувати події деякого приймача протягом періоду, коли приймач був зайнятий виконанням дії. Це змусило б на події очікувати приймач, який закінчує дію, доти, коли може бути прийнято інше подія.

У інтерпретації часу, що чергується, кілька кінцевих автоматів можуть породжувати події для певного приймача, перш ніж приймаючому кінцевому автоматом дозволяється отримувати події і виконувати дію (тому що дії чергуються довільно). Це призводить до того, що багаторазові події знову чекатимуть на обробку одним кінцевим автоматом.

6. Якщо кінцевий автомат породжує багаторазові події для одного екземпляра, що приймає, події будуть отримані в згенерованому порядку.

7. Якщо є очікування обробки події для деякого кінцевого автомата, які були породжені різними відправниками, то не визначено, яка подія буде прийнята спочатку.

5.7 Багаторазове управління

Багато програм реального часу вимагають, щоб певні значення даних активно і багаторазово контролювалися і керувалися протягом тривалого часу. Наприклад, завдання про підприємство з виробництва соку, розділу 5.2 існують вимоги повільно підвищувати температуру соку до температури пастеризації і потім містити сік при постійній температурі певний інтервал часу.

Визначення частоти

Частота виконання дії стану є важливим питанням аналізу, на яке повинні відповісти прикладні експерти, в даному випадку – інженери на виробництві. Відповідь братиме до уваги кількість теплоти, яка може бути виділена нагрівачем на якомусь інтервалі часу, конкретну теплоємність, кількість соку та точність збігу фактичної температури з температурою, що визначається зміною.

І нарешті, зверніть увагу, що в будь-якому подібному завданні, ймовірно, виявиться, що потрібний ряд частот для збереження різних промислових керованих процесів. Аналітик повинен гарантувати, що така частота визначається окремо. Внаслідок цього проектувальник знатиме, як упакувати такі процеси в задачах та кодах.