

Семантика

Структурна дія функції працює на статично вказаному структурному елементі деякого класифікатора. Дія виконуватиметься на об'єкті за умови підключення до вхідного контакту. Якщо структурна функція є кінцевою, то дії по функції мають ту ж семантику, як і дії по посиланнях, які мають функцію в якості мети. Див спеціалізацію з StructuralFeatureAction. Семантика не визначена для доступу за структурною ознакою, яка порушує його видимість. Семантика для статичних функцій не визначена.

Структурні особливості та асоціації об'єкта можуть змінюватися часом. Проте, тип об'єкта вхідного контакту структурної дії функції ідентифікується як єдиний класифікатор, і передбачається, що об'єкт, який передається дії класифікують за цим класифікатором прямо або побічно. Структурна функція згадується як елементна модель користувача, тому вона однозначно ідентифікована, навіть якщо є й інші структурні особливості одного і того ж імені на інших класифікаторах.

Позначення

Немає конкретних позначень.

Обґрунтування

StructuralFeatureAction вводиться для абстрактних аспектів структурних дій об'єктів.

Зміни в порівнянні з попереднім UML

StructuralFeatureAction оновлена в UML 2. Вона узагальнює AttributeAction в UML 1.5.

11.3.49 TestIdentityAction (від IntermediateActions)

TestIdentifyAction це дія, яка перевіряє, два значення однакових об'єктів.

Узагальнення

- «Action (від BasicActions)» на стор 249

Опис

Ця дія повертає значення true, якщо два вхідних значення однакові, та значення false, якщо ні.

Атрибути

Немає додаткових атрибутів.

Асоціації

- перша: InputPin [1..1]

Повертає котакт, на якому розміщується об'єкт. (Підмножина Action :: вхід)

- Результат: OutputPin [1..1])

Повідомляє чи є два вхідні об'єкти є ідентичними. (Підмножина Action :: вихід)

- другий: InputPin [1..1]

Повертає контакт, на якому розміщується об'єкт. (Підмножина Action :: вхід)

Обмеження

[1] Вхідні контакти не мають типу.

self.first.type-> size() = 0

i self.second.type-> size() = 0

[2] Множинність вхідних контактів є 1..1.

self.first.multiplicity.is (1,1)

i self.second.multiplicity.is (1,1)

[3] Тип результату UML є стандартним примітивним типом Boolean. (Це не безпосередньо представлено в OCL на рівні метамоделі.).

Семантика

Коли всі передумови дій були задоволені, вхідні значення виходять з вхідних контактів і стають доступні для обчислень. Якщо два вхідних значення представляють собою один і той же об'єкт (незалежно від будь-якого рівня реалізації кодування), справжнє значення поміщається на вихідний контакт виконання дії; в іншому випадку значення помилково поміщається на вихідний контакт. Виконання дії завершено.

Позначення

Немає конкретних позначень.

Обґрунтування

TestIdentityAction вводиться щоб повідомити, що два значення відносяться до одного об'єкту.

Зміни в порівнянні з попередньою UML

TestIdentityAction не відрізняється від UML 1.5.

11.3.50 UnmarshallAction (від CompleteActions)

UnmarshallAction це дія, яка розбиває об'єкт відомого типу в виходи кожен з яких дорівнює значенню з структурною особливістю об'єкта.

Узагальнення

- «Action (від BasicActions)» на стор 249

Опис

Виходи цієї дії відповідають конструктивним особливостям зазначеного типу. Вхід повинен бути такого ж типу.

Атрибути

Немає додаткових атрибутів.

Асоціації

- Об'єкт: InputPin [1..1]

Об'єкт, який буде розпаковано. {Підмножини Action :: input}

- unmarshallType: Класифікатор [1..1]

Тип об'єкта, що підлягає розпакуванню.

- Результат: OutputPin [1 .. *]

Значення структурних особливостей вхідного об'єкта. {Підмножини Action :: output}

Обмеження

[1] Тип вхідного контакту об'єкта повинен бути таким же, як маршалізований класифікатор.

[2] Множинність вхідного об'єкта контакту 1..1.

[3] Кількість результатів вихідних контактів повинно бути таким же, як кількість структурних особливостей маршалізації класифікатора.

[4] Тип та впорядкування кожного вихідного результату контакту повинен бути таким же, як і відповідні структурні особливості маршалізації класифікатора.

[5] Кратність кожного результуючого вихідного контакту повинна бути сумісна з безліччю відповідних структурних особливостей.

[6] Маршалізований класифікатор повинен мати принаймні одну структурну особливість.

[7] unmarshallType повинен бути класифікатором з впорядкованими атрибутами.

Семантика

Коли об'єкт доступний на вхідному контакті, значення структурних особливостей зазначеного класифікатора витягуються від об'єкта і поміщаються на вихідних контактах, в порядку структурних особливостей зазначеного класифікатора. Порядок значень в вихідному контакті є такого ж, як порядок відповідних структурних особливостей, якщо такі є.

Позначення

Немає конкретних позначень

Опис

Див «UnmarshallAction (як спеціалізовані)» на стор 432.

Обґрунтування

UnmarshallAction вводиться щоб зчитувати всі структурні особливості об'єкта відразу.

Зміни в порівнянні з попередньою UML

UnmarshallAction таке ж, як UML 1.5, за винятком того, що назва metaassociation до вхідного контакту змінюється.

11.3.51 ValuePin (from BasicActions)

Узагальнення

- «InputPin (від BasicActions)» на стор 269

Опис

Значення контакту як у вхідного контакту, забезпечує значення шляхом оцінки специфікації значення.

Атрибути

Немає додаткових атрибутів.

Асоціації

- Значення: ValueSpecification [1..1]

Значення, що контакт забезпечуватиме. {Піднабір Element :: ownedElement}

Обмеження

[1] Тип специфікації значення має бути сумісний з типом результату контакту.

Семантика

Значення контакту є результатом оцінки специфікації значення.

Нотація

Немає конкретного позначення. Див розширення в Activities.

Обґрунтування

ValuePin вводиться, щоб забезпечити самий основний спосіб надання матеріалів для дій.

Зміни в порівнянні з попередньою UML

ValuePin є новим для UML 2.

11.3.52 ValueSpecificationAction (від IntermediateActions)

ValueSpecificationAction це дія, яка оцінює специфікації значення.

Узагальнення

- «Action (від BasicActions)» на стор 249

Опис

Дія повертає результат обчислення значення специфікації.

Атрибути

Немає додаткових атрибутів.

Асоціації

- Значення: ValueSpecification [1]

Специфікація значення для оцінки. {Піднабір Element :: ownedElement}

- Результат: OutputPin [1]

Дає вихідний контакт, на якому результат виводиться. {Підмножини Action :: вихід}

Обмеження

[1] Тип специфікації значення має бути сумісний з типом результату контакту.

[2] Множинність результату штиря 1..1.

Семантика

Специфікація значення обчислюється, коли дію дозволено.

Нотація

Див «ValueSpecificationAction (як спеціалізовані)» на стор 433.

Приклади

Див «ValueSpecificationAction (як спеціалізовані)» на стор 433.

Обґрунтування

ValueSpecificationAction вводиться для вводу константи і інших специфікацій значення в поведінці.

Зміни в порівнянні з попередньою UML

ValueSpecificationAction замінюється LiteralValueAction з UML 1.5.

11.3.53 VariableAction (від StructuredActions)

Узагальнення

- «Action (від BasicActions)» на стор 249

Опис

VariableAction є абстрактним класом для дій, які працюють на статично заданій змінній.

Атрибути

Немає додаткових атрибутів.

Асоціації

- variable : Variable [1..1]

Змінна для читання.

Обмеження

[1] Дія повинна бути в рамках цієї змінної.

self.variable.isAccessibleBy (self)

Семантика

Змінна дія є абстрактним метакласом. Для семантики дивитись його конкретні підтипи.