

# Software Design

## 5. Modeling of Services

Natsuko Noda  
nnoda@shibaura-it.ac.jp

Copyright© Natsuko NODA, 2014-2024

1

# ソフトウェア設計論

## 5. サービスのモデリング

野田 夏子  
nnoda@shibaura-it.ac.jp

Copyright© Natsuko NODA, 2014-2024

2

## Today's topics

- Modeling of service
  - Use case
  - Use case diagram
- Common notations of UML diagrams

Copyright© Natsuko NODA, 2014-2024

3

## 本日のお題

- サービスのモデリング
  - ユースケース
  - ユースケース図
- UML図に共通な記法

Copyright© Natsuko NODA, 2014-2024

4

## State machine diagram

- Every software has static and dynamic structures (dynamic structure = behavior) and has various aspects.
- Which aspects is more important? It depends on each software.
  - Ex. bookkeeping system - data transform from input to output; reaction to the event is not so important.
  - Ex. embedded system - event-driven.

## Modeling of service ~ Use case

## ステートマシン図

- すべてのソフトウェアは、静的な構造と動的な構造(振舞い)を持ち、様々な異なる側面を持つ
- どの側面が重要であるかは、ソフトウェアによって異なる
  - 例. 経理システム - 入力から出力への変換。イベントに対する反応という側面はあまり重要ではない
  - 例. 組込みシステム - イベント駆動(イベントに対する反応)

## サービスのモデリング ~ ユースケース

## Why is software modeling useful?

- It provides basic viewpoints and description methods.
  - Although many types of artifacts of software development (programs, specification, design, ...) exist, there are patterns or common types of viewpoints and description methods that are used.
- It lightens scale and/or complexity of software.
- It provides an accurate way to describe software.

## ソフトウェアモデリングが有用な理由

- 基本的な視点や記述方法の提供
  - ソフトウェア開発中に扱うものは、プログラム、要求、仕様、設計など様々であるが、用いられる視点や記述方法には類型がある
- 規模や複雑さの軽減
- 正確な表現方法の提供

## Modeling viewpoints

- Structure of the target
  - Class diagram, object diagram
- Flow of control and data
  - Activity diagram
- Example of message interaction
  - Sequence diagram, communication diagram
- Behavior as a state machine
  - State machine diagram
- How the system is used

## モデリングの視点

- 対象の構造
  - クラス図、オブジェクト図
- 制御やデータの流れ
  - アクティビティ図
- メッセージのやりとりの例示
  - シーケンス図、コミュニケーション図
- 振舞いの状態機械としての表現
  - ステートマシン図
- システムの利用方法

## Use case

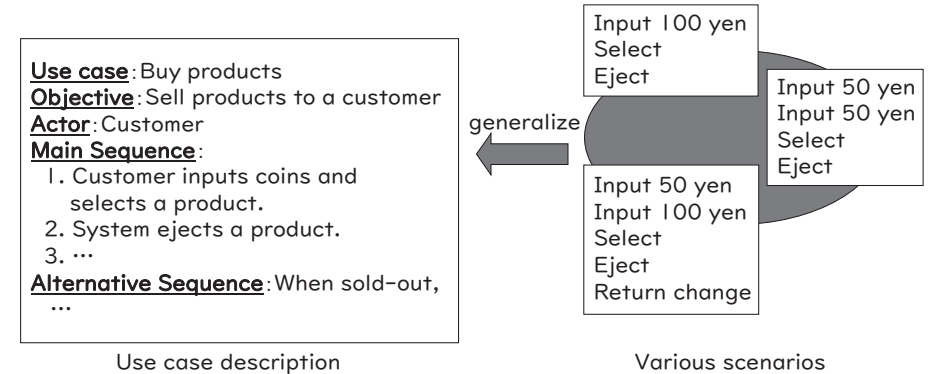
- A use case shows how a system is used.
  - Required usage.
- Users of the system use its functions that are observable from outside the system.
  - Ex. About online sales system, Customers use the function of product purchase. An enterprise uses the function of merchandise management and customer management.
  - A use case is such a function that is observable from the outside.

Copyright© Natsuko NODA, 2014-2024

13

## Use case

- Describes an interaction between a user and a system.
- a generalized description of various similar scenarios.



Copyright© Natsuko NODA, 2014-2024

15

## ユースケース

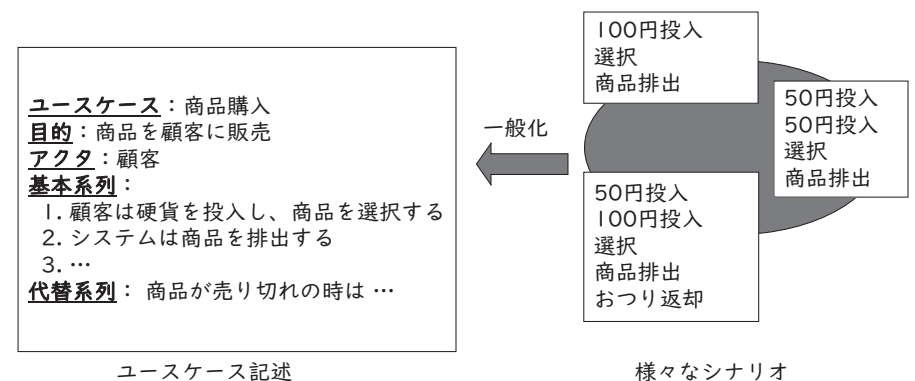
- ユースケースはシステムの使われ方を示す
  - 要求される利用方法、使い方
- ユーザは、システム外部から観測可能な機能を利用する
  - 例: ネット販売システムにおいて  
顧客は商品購入の機能を利用  
事業者は商品管理や顧客管理の機能を利用
  - ユースケースは、こうした外部から観測可能な機能

Copyright© Natsuko NODA, 2014-2024

14

## ユースケース

- ユーザとシステムのやりとりを記述
- 類似する様々なシナリオの一般化した記述



Copyright© Natsuko NODA, 2014-2024

16

## Use case

- A set of scenarios tied together by a common user goal.
- A service performed by the system.

## ユースケース

- 共通するユーザの目的ごとにまとめられたシナリオの集合
- システムによって行われるサービス

## Use case diagram

- shows use cases of a system.
- The purpose of use case diagrams is not to show all specific functions in detail, but to capture roughly what kinds of usages exist.
  - To group functions that have a common goal.
  - To classify functions.
  - As a starting point of consideration.

## ユースケース図

- システムのユースケースを示す
- ユースケース図の目的は、すべての個別具体的な機能を列挙することではなく、どのようなシステムの使われ方があるかを大づかみにとらえることである
  - 機能を共通するユーザのゴール毎にグループ化する
  - 機能の分類として使う
  - 検討の出発点として使う

## Use case diagram (Cont.)

### • Subject

- A system currently under consideration (in focus)

### • Actor

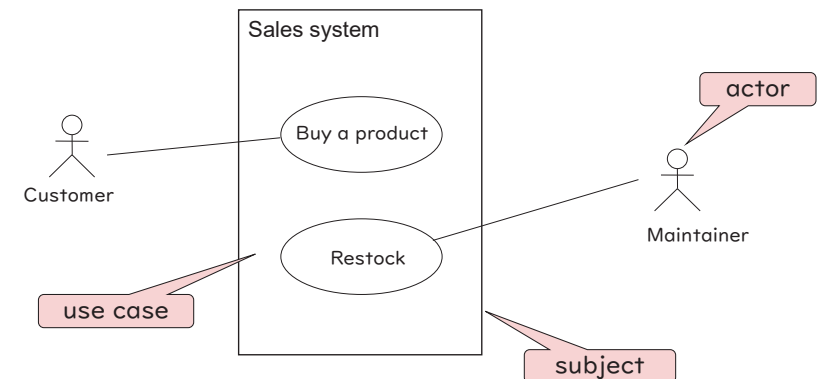
- a role that a user plays with respect to the subject
- user who carries out use cases
  - doesn't have to be a human. Outside entity that interacts with the system.
  - If the system performs a service for another computer system, that other system is an actor.

- A use case diagram shows what kind use cases a subject has and which actor each use case is associated with.

Copyright© Natsuko NODA, 2014-2024

21

## Notation



Copyright© Natsuko NODA, 2014-2024

23

## ユースケース図 (Cont.)

### • サブジェクト

- 今検討している(今注目している)システム

### • アクタ

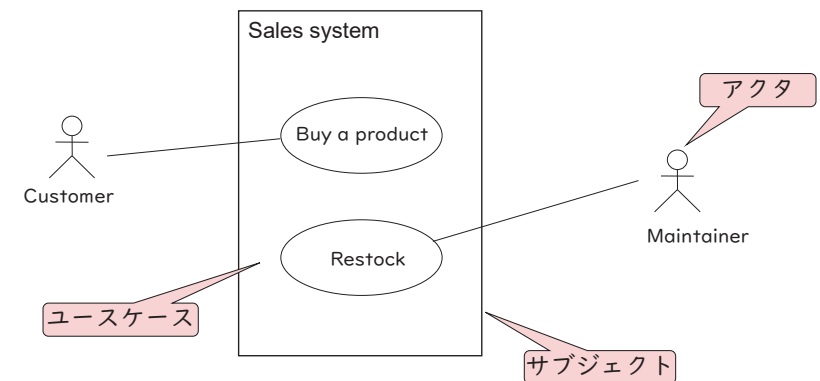
- サブジェクトに関してユーザが果たす役割
- ユースケースを実行するユーザ
  - 人である必要はない。システムとやりとりする外部実体
  - システムが他のコンピュータシステムに対するサービスを提供しているなら、その他のシステムはアクタになる

- ユースケース図はサブジェクトがどのようなユースケースを持ち、各ユースケースがどのアクタと関わっているかを記述する

Copyright© Natsuko NODA, 2014-2024

22

## 記法



Copyright© Natsuko NODA, 2014-2024

24

## Notation (Cont.)

### • Use case

- Ellipse, either containing the name of the use case or with the name of the use case placed below the ellipse



### • Actor

- "stick man" icon with the name of the actor in the vicinity (usually above or below) the icon



Copyright© Natsuko NODA, 2014-2024

25

## 記法 (Cont.)

### • ユースケース

- 楕円で表現。名前を中、あるいは楕円の下に書く



### • アクタ

- "棒人間"のアイコンで表現。名前をアイコンの近く(通常は上か下)に書く



Copyright© Natsuko NODA, 2014-2024

26

## Relationships in use case diagram

### • Relationship between actor and use case

- Association: shows that the actor carries out the use case
- A use case is related to at least one actor.
- A use case may be related to two or more actors.
- An actor is related to at least one use case.
- An actor may be related to two or more use cases.

Copyright© Natsuko NODA, 2014-2024

27

## ユースケース図中の関係

### • アクタとユースケースの関係

- 関連: アクタがユースケースを実行すること示す
- ユースケースは少なくともひとつのアクタと関連する
- ユースケースは2つ以上のユースケースと関係する場合もある
- アクタは少なくともひとつのユースケースと関連する
- アクタは2つ以上のユースケースと関係する場合もある

Copyright© Natsuko NODA, 2014-2024

28

## Relationships between use cases

- (informative) three types of relationships between use cases
  - Extension: inserts other use case at the extension point.
    - I recommend not to use this, at least for beginners!
  - Inclusion: includes other use cases.
    - This relationship is intended to be used when there are common parts of the behavior of two or more use cases.
    - The primary use of this is for reuse of common parts.
  - Generalization: shows generalization/specialization of use cases.

Copyright© Natsuko NODA, 2014-2024

29

## ユースケース間の関係

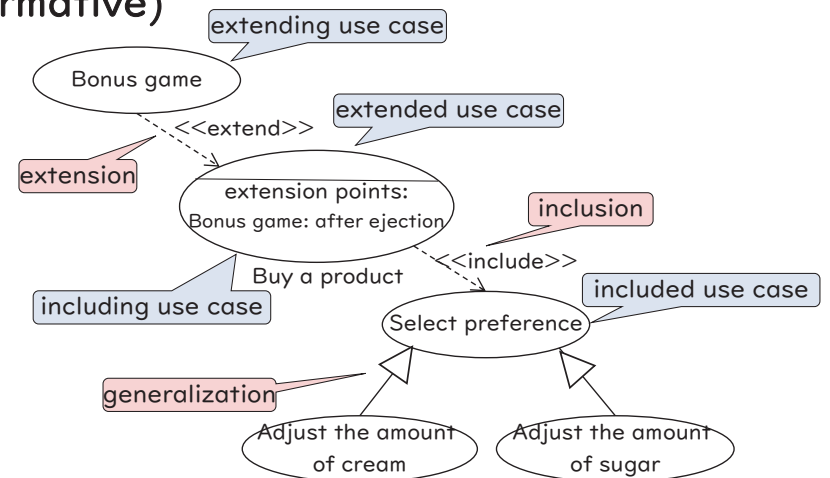
- (参考) ユースケース間の関係は以下の3つ
  - 拡張：拡張点に他のユースケースを挿入する
    - 特に初心者場合は、使わないことをおすすめします
  - 包含：他のユースケースを含む
    - この関係は、2つ以上のユースケースがその振舞いに関して共通部分を持つ場合に用いられることを意図している
    - この関係の第一の利用方法は、共通部分の再利用にある
  - 汎化：ユースケースの汎化/特化の関係を示す

Copyright© Natsuko NODA, 2014-2024

30

## Example

- (informative)

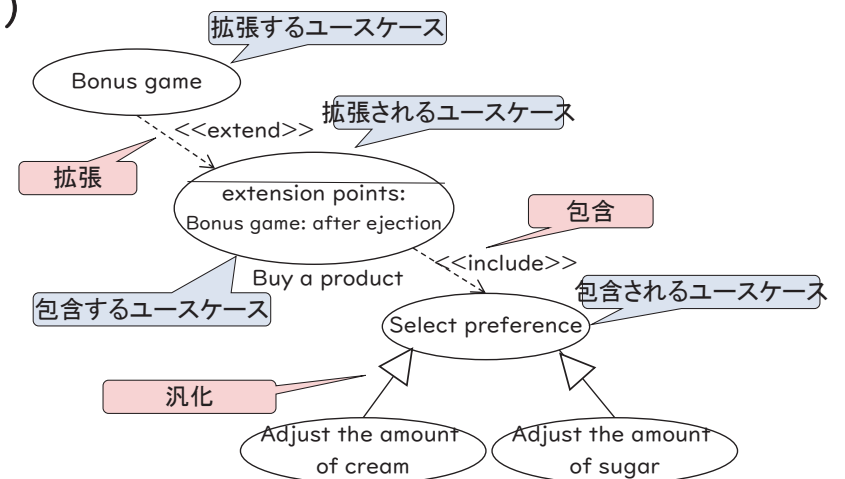


Copyright© Natsuko NODA, 2014-2024

31

## 例

- (参考)



Copyright© Natsuko NODA, 2014-2024

32



## Notation of use cases

- Use cases are well known as an important part of the UML.
- But the UML doesn't prescribe how to describe the content of use cases.

## ユースケースの記法

- ユースケースはUMLの重要な一部を成す
- しかしUMLはユースケースの内容をどのように記述するかについては規定していない

## Template of use case description

- No standard.
- There are some introduced templates.

## ユースケース記述のテンプレート

- 標準はない
- いくつかのテンプレートが紹介されている

## Template example (detailed version)

1. **Use case name and identifier** (unique identifier and short name of the use-case)
2. **Objective** – What is the ultimate objective of the use-case. What is it trying to achieve? What was the source of the use-case requirement?
3. **Priority** – What is the priority? Low, Medium, High.
4. **Source** – Who gave us this use-case. Typically a person with their role in parenthesis: "John Smith (marketing)"
5. **Actors** – Who is involved in the use-case? Which actors/stakeholders?
6. **Flow of Events**
  1. Basic Flow – Flow of events normally executed in the use-case
  2. Alternative Flow – A secondary flow of events due to infrequent conditions
  3. Exceptions – Exceptions that may happen during the execution of the use case
7. **Includes** (other use case IDs)
8. **Preconditions** – Any condition that must be satisfied before the use case begins
9. **Post conditions** – The conditions that will be satisfied after the use case is successfully completed
10. **Notes/Issues** – Any relevant notes or issues that need to be resolved

Copyright© Natsuko NODA, 2014-2024

37

## テンプレートの例 (詳細なもの)

1. **Use case name and identifier** (unique identifier and short name of the use-case)
2. **Objective** – What is the ultimate objective of the use-case. What is it trying to achieve? What was the source of the use-case requirement?
3. **Priority** – What is the priority? Low, Medium, High.
4. **Source** – Who gave us this use-case. Typically a person with their role in parenthesis: "John Smith (marketing)"
5. **Actors** – Who is involved in the use-case? Which actors/stakeholders?
6. **Flow of Events**
  1. Basic Flow – Flow of events normally executed in the use-case
  2. Alternative Flow – A secondary flow of events due to infrequent conditions
  3. Exceptions – Exceptions that may happen during the execution of the use case
7. **Includes** (other use case IDs)
8. **Preconditions** – Any condition that must be satisfied before the use case begins
9. **Post conditions** – The conditions that will be satisfied after the use case is successfully completed
10. **Notes/Issues** – Any relevant notes or issues that need to be resolved

Copyright© Natsuko NODA, 2014-2024

38

## Template example (simple version)

- Use case name
- Goal level
- Main Success Scenario
- Extensions

Copyright© Natsuko NODA, 2014-2024

39

## テンプレートの例 (シンプルなもの)

- Use case name
- Goal level
- Main Success Scenario
- Extensions

この中身が重要であるわけではないので、日本語訳は省

Copyright© Natsuko NODA, 2014-2024

40

## Template in this class

- Use case name
- Actors
- Objective : A brief description of the objective of use case, typically one or two sentences.
- Preconditions (if any) : One or more conditions that must be true at the start of use case, from the perspective of this use case.
- Postcondition (if any) : Condition that is always true at the end of the use case.
- Description of main sequence : a narrative description of the main sequence of the use case. The description is in the form of the input from the actor, followed by the response of the system. Usually written in itemized formation.
- Description of alternative sequences (if any) : Narrative description of alternative branches off the main sequence.
- Remarks

Copyright© Natsuko NODA, 2014-2024

41

## この授業でのテンプレート

- ユースケース名
- アクタ
- 目的: このユースケースが何をするかの概要
- (あれば) 事前条件: このユースケースが行われる前に成り立っていないなければならない1つ以上の条件
- (あれば) 事後条件: ユースケースが行われた後に成り立っていないなければならない条件
- 基本系列: ユースケースが正常に行われる際の基本的な手順。アクタからの入力と、それに対するシステムの反応の形で記述される。通常、箇条書きにする
- (あれば) 代替系列: 基本系列とは別の重要な手順。基本系列からの代替の分岐
- 備考

Copyright© Natsuko NODA, 2014-2024

42

## Use case and actor

- If a use case is related to two or more actors, those all actors must participate to the use case.
- All actors must be appeared in the description of main sequence and/or alternative sequence.

Copyright© Natsuko NODA, 2014-2024

43

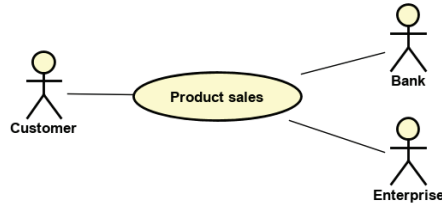
## ユースケースとアクタ

- もしユースケースが2つ以上のアクタと関係していたら、それらのアクタすべてがユースケースに参加していなければならない
- すべてのアクタはその関係しているユースケースの基本系列・代替系列(のどちらかあるいは両方)の記述に登場しなければならない

Copyright© Natsuko NODA, 2014-2024

44

## Example



- Description of main sequence (of "Product sales"):
  1. Customer selects a product and add it to cart.
  2. Customer goes to the checkout and enters necessary information.
  3. Customer directs payment.
  4. System requests Bank to process the payment.
  5. Enterprise performs shipping processing.

Copyright© Natsuko NODA, 2014-2024

45

## Suggestions of description

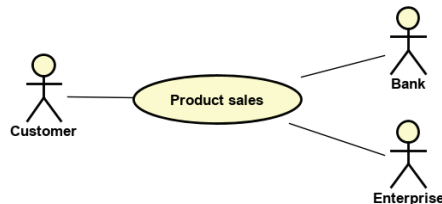
- Each step (in a use case) should be a simple statement and should clearly show who is carrying out the step. The step should show the intent of the actor, not the mechanics of what the actor does.

By “UML distilled” , Martin Fowler

Copyright© Natsuko NODA, 2014-2024

47

## 例



- ("product sales"の)基本系列
  1. Customerは商品を選びカートに入れる
  2. Customerはレジに行き、必要な情報を入力する
  3. Customerは支払いを指示する
  4. システムはBankに決済処理を依頼する
  5. Enterprise(事業者)は商品の発送処理を行う

Copyright© Natsuko NODA, 2014-2024

46

## 記述のヒント

- ユースケースの各ステップはシンプルな文であって、誰がそのステップを行うかを明示すべきである。そのステップは、アクタの意図を示すべきであって、アクタが行うことの方法(やり方、アクタがそれをどのような方法で行うか)を示すべきではない

By “UML distilled” , Martin Fowler

Copyright© Natsuko NODA, 2014-2024

48

## When to utilize use cases

- To analyze the essential usage of the system.
  - not "How", but "What"
- To define the services the system performs.
  - to define the interaction with the outside
  - to define the behavior as the services for the actors.
    - not internal processing, but external services to the users.
- To understand the system from the viewpoint of users.
  - use cases can be the interface between users and developers.

## Common notations of UML diagrams

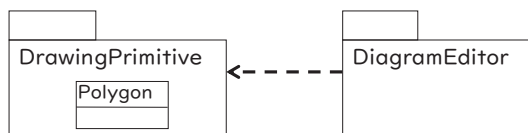
## いつユースケースを利用するか

- システムの本質的な利用方法を分析するため
  - "How"ではなく、"What"
- システムが行うサービスを定義するため
  - 外部とのやりとりを定義する
  - アクタに対するサービスとしてふるまいを定義する
    - 内部の処理ではなく、ユーザに対するサービスを定義
- ユーザの視点からシステムを理解するため
  - ユースケースは、ユーザと開発者のインタフェースとして使われ得る

## UML図に共通する記法

## Package

- A package is used to group elements, and provides a namespace for the grouped elements.
- shown as a large rectangle with a small rectangle (a "tab") attached to the left side of the top of the large rectangle.
- the members of the package may be shown within the large rectangle.



Copyright© Natsuko NODA, 2014-2024

53

## Package (Cont.)

- To allow the use of names to refer to package members from other namespaces, package import can be utilized.



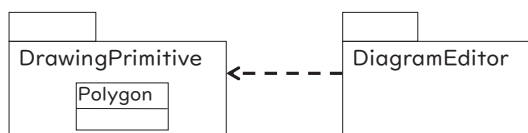
From the "WebShop" package, elements of "ShoppingCart" can be referenced using unqualified names.  
(unqualified name: ex. "Item"  
qualified name: ex. "ShoppingCart::Item")

Copyright© Natsuko NODA, 2014-2024

55

## パッケージ

- パッケージは要素をグループ化し、グループ化された要素のための名前空間を提供する
- 上の左側に小さい四角形("タブ")がついた大きい四角形で示す
- パッケージのメンバ(含まれる要素)を、大きい四角形の内部に示すこともできる

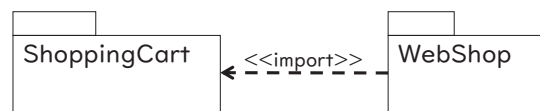


Copyright© Natsuko NODA, 2014-2024

54

## パッケージ (Cont.)

- 他の名前空間からパッケージのメンバへの参照を許すために、パッケージのインポートを使うことができる



Importを使うことにより、"WebShop"パッケージから、"ShoppingCart"パッケージの要素を非修飾名で参照することができる  
(非修飾名: 例 "Item"  
修飾名: 例 "ShoppingCart::Item")

Copyright© Natsuko NODA, 2014-2024

56

## Comment (or Note symbol)

### • Comment

- shows various remarks (to elements).
- shown as a rectangle with the upper right corner bent.
- connections to annotated elements are shown by dashed lines. (May be suppressed.)

#### Comment:

Used like this. Already shown in previous pages. This can be used in all diagrams.

Car

Includes buses and trucks, but not motorbikes

Copyright© Natsuko NODA, 2014-2024

57

## Stereotype

- classifies same type elements.
- one of the extension mechanisms of UML.
  - Using stereotype, we can add model elements to pre-defined model elements (e.g. "class").
  - adds specific meanings to the predefined element.

Copyright© Natsuko NODA, 2014-2024

59

## コメント (または ノート記号)

### • コメント

- 要素に対する様々な注釈を示す
- 右上を折った四角形で示す
- 注釈が付けられる要素への接続は、破線で示す (つけない場合もある)

#### コメント:

このように使います。以前のページにもすでに現れています。これはすべての図で使うことができます。

Car

バスとトラックを含むがバイクは含まない

Copyright© Natsuko NODA, 2014-2024

58

## ステレオタイプ

cf. SE-J, 10.2.4

- 同じタイプの要素を分類する
- UMLの拡張メカニズムのひとつ
  - ステレオタイプを使うことによって、すでに定義済みのモデル要素(例えばクラス)にモデル要素を追加することができる
  - 定義済みの要素に特定の意味を加える

Copyright© Natsuko NODA, 2014-2024

60

## Stereotype

- usage: «NameOfStereotype»
  - « » guillemets



- Ex.



Copyright© Natsuko NODA, 2014-2024

61

## For your review

- Draw a use case diagram of a library, and write down the content of each use case in the diagram.

Copyright© Natsuko NODA, 2014-2024

63

## ステレオタイプ

- 使用方法(記法): «NameOfStereotype»
  - « » ギルメット



- 例



Copyright© Natsuko NODA, 2014-2024

62

## For your review

- 図書館のユースケース図と、ユースケース図中のそれぞれのユースケースに対するユースケース記述を書いてみよう

Copyright© Natsuko NODA, 2014-2024

64



## For your review

- When and for what purpose do we develop use cases (use case diagram and use case descriptions)?

## For your review

- いつ、どんな目的でユースケース(ユースケース図とユースケース記述)を作るのだろう？