

Software Design

6. Review of Modeling

Natsuko Noda
nnoda@shibaura-it.ac.jp

Copyright© Natsuko NODA, 2014-2024

1

ソフトウェア設計論

6. モデリングの復習

野田 夏子
nnoda@shibaura-it.ac.jp

Copyright© Natsuko NODA, 2014-2024

2

Today's topics

- Review of modeling we learnt
 - Answers of exercises

Copyright© Natsuko NODA, 2014-2024

3

本日のお題

- モデリング復習
 - これまでの練習問題の回答

Copyright© Natsuko NODA, 2014-2024

4

Review of Modeling – What we learnt?

Copyright© Natsuko NODA, 2014-2024

5

What is software modeling?

- Software modeling is the task of abstracting an information system and its related things and expressing them in a formal notation.
- Various targets of software modeling:
 - Real world related to an information system
 - Specification of an information processing system
 - Anything related to software design
 - Anything related to software implementation
 - etc.
- Software modeling is utilized in various processes of software development.
 - It is important in designing software.
 - But, software modeling ≠ designing software.

Copyright© Natsuko NODA, 2014-2024

7

モデリング復習

Copyright© Natsuko NODA, 2014-2024

6

ソフトウェアモデリングとは？

- 情報システムに関わる対象を抽象化し、それを形式性のある記法で表現する作業
- ソフトウェアモデリングの対象は様々
 - 情報システムに関わる現実世界
 - 情報処理システムの仕様
 - ソフトウェアの仕様に関わるもの
 - ソフトウェアの設計に関わるもの
 - ソフトウェアの実装に関わるもの
 - etc.
- ソフトウェア開発の様々な工程で利用される
 - 設計においても重要
 - ただし、モデリング＝設計ではない

Copyright© Natsuko NODA, 2014-2024

8

Key point: Abstraction

- Abstract (verb): (abstract something from)
Consider something theoretically or separately from (something else)
 - By Oxford Living Dictionaries
- Systems and software are very complexed.
We cannot consider all things at once.
- Separate problems by viewpoints and concentrate on one specific aspect.

モデリングのキー：抽象化

- 抽象：「事物または表象の或る側面・性質を抽(ぬ)き離して把握する心的作用」(広辞苑)
- 抽象化：抽象すること
- システムやソフトウェアは非常に複雑であり、関連する様々なことを一度に考えることは不可能
- 問題を視点にそって分割し、一時には特定の視点・側面についてのみ考える

UML

- Unified modeling language
 - *The OMG's Unified Modeling Language™ (UML®)* helps you **specify**, **visualize**, and **document** models of software systems, including their structure and design.
 - can be used for business modeling and modeling of other non-software systems too.
 - not programming language, but modeling language
- It has a syntax and semantics.
- You must follow the rules of the language by using the language.

UML

- Unified modeling language 統一モデリング言語
 - *OMG's Unified Modeling Language™ (UML®)* はソフトウェアシステムおよびその構造や設計のモデルを、**仕様化**し、**視覚的**にし、**文書化**することを助ける
 - これはビジネスのモデリングやその他ソフトウェア以外のモデリングにも使われる得る
 - プログラミング言語ではなく、モデリング**言語**であることに注意
- 言語は、書き方(文法、シンタクス)と読み方(解釈の仕方、セマンティクス)を持つ
- 言語が持つルールに従って、その言語を使わなければならない

Class diagram and object diagram

- A class diagram describes the structure of the modeling target using classes.
 - the classes that consist the target.
 - the various kinds of static relationships that exist among them.
 - A class diagram describes an abstract structure.
- An Object diagram describes the structure of the modeling target using objects.
 - the objects that exist in the modeling target at a certain point in time.
 - the links that exist between objects.
 - An object diagram describes a concrete structure.
 - It shows one concrete example of the abstract structure the class diagram describes.

Copyright© Natsuko NODA, 2014-2024

13

クラス図とオブジェクト図

- クラス図は、クラスを用いて対象の構造を表す
 - 対象を構成する要素としてのクラスと
 - 要素間に存在する様々な関係を用いる
 - クラス図は抽象的な構造を表す
- オブジェクト図は、オブジェクトを用いて対象の構造を表す
 - モデリング対象について、ある時点で存在するオブジェクトと
 - オブジェクト間に存在するリンクを用いる
 - オブジェクト図は具体的な構造を表す
 - クラス図が示す抽象構造のひとつの具体例を表す

Copyright© Natsuko NODA, 2014-2024

14

Multiplicity (on association)

- Indicates how many objects may be related to an object of the class on the other end.

Indicator	Meaning
0..1	Zero or one
1	One only
0..*	Zero or more
1..*	One or more
n	Only n (where $n > 1$)
0..n	Zero to n (where $n > 1$)
1..n	One to n (where $n > 1$)



Copyright© Natsuko NODA, 2014-2024

15

多重度

- そのクラスのあるオブジェクトが、関連する他方のクラスのいくつかのオブジェクトと関係し得るかを示す

多重度の表現	意味
0..1	0か1
1	1
0..*	0以上
1..*	1以上
n	n (ただし $n > 1$)
0..n	0から n (ただし $n > 1$)
1..n	1から n (ただし $n > 1$)



Copyright© Natsuko NODA, 2014-2024

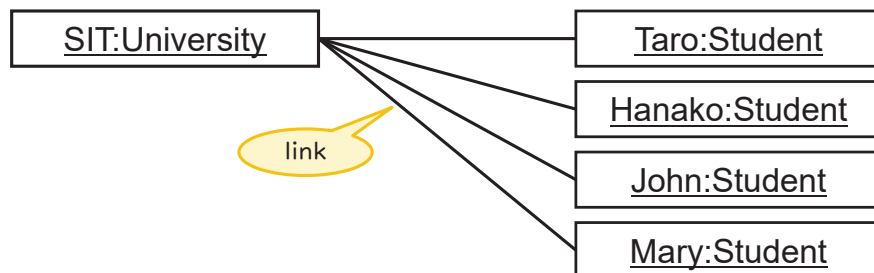
16

Association and link

- Association: structural relationship between classes.



- Link: An instance of an association; between objects.



Copyright© Natsuko NODA, 2014-2024

17

Example

- Object diagram:



- Class diagram:



Fill in multiplicities in the class diagram above by reference to the object diagram.

Copyright© Natsuko NODA, 2014-2024

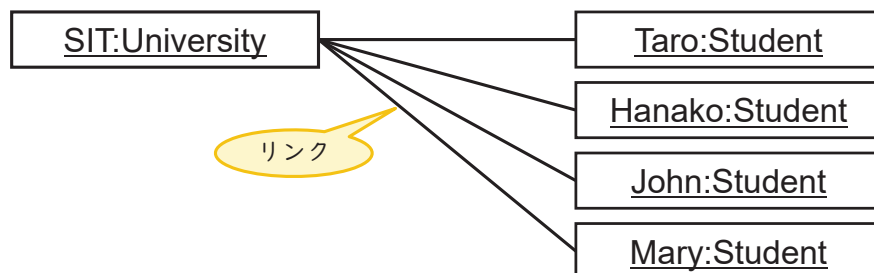
19

関連とリンク

- 関連：クラス間の構造上の関係



- リンク：関連のインスタンス(具体化したもの)。オブジェクト間に存在



Copyright© Natsuko NODA, 2014-2024

18

例

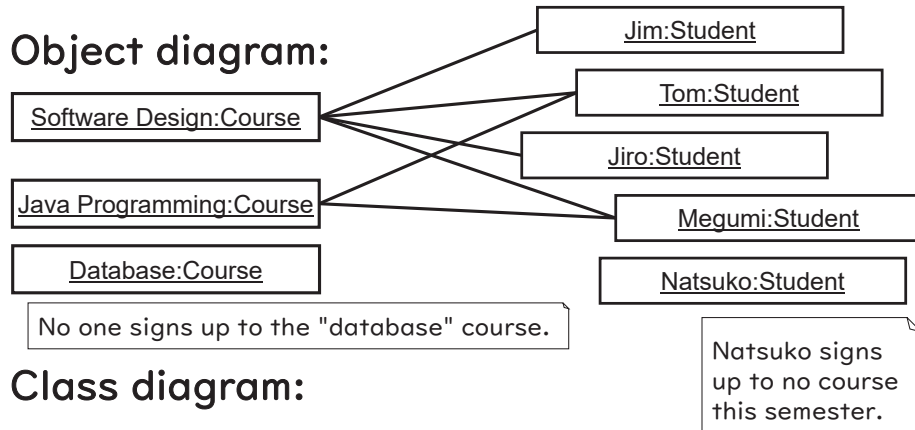
- (日本語訳省略)

Copyright© Natsuko NODA, 2014-2024

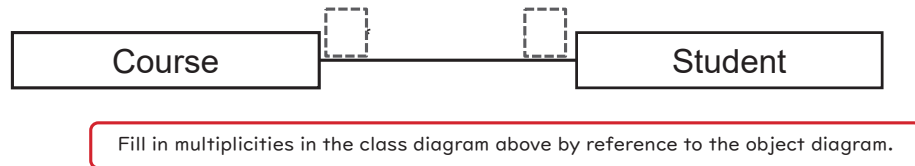
20

Example

• Object diagram:



• Class diagram:



Copyright© Natsuko NODA, 2014-2024

21

例

- (日本語訳省略)

Copyright© Natsuko NODA, 2014-2024

22

Exercise

• Draw a class diagram describing the following:

- There are three classes : University, Teacher, Student
- There are three associations:
 - between University and Teacher,
 - between University and Student,
 - between Teacher and Student
- add an appropriate association name
- depict multiplicities on associations

Copyright© Natsuko NODA, 2014-2024

23

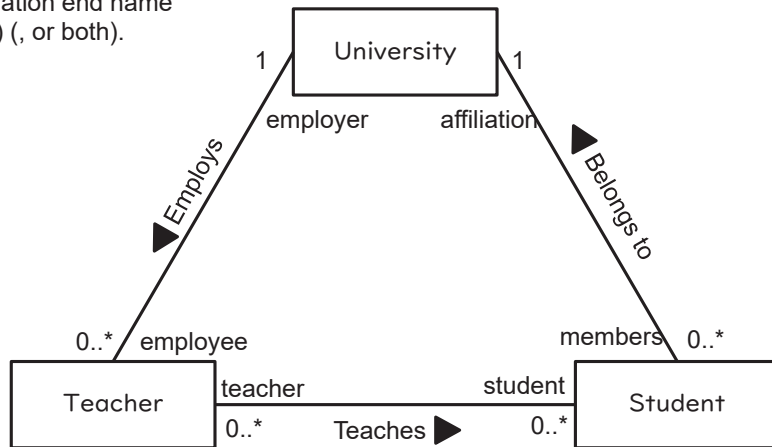
練習

- (日本語訳省略)

Copyright© Natsuko NODA, 2014-2024

24

You may specify association name or association end name (=role) (, or both).



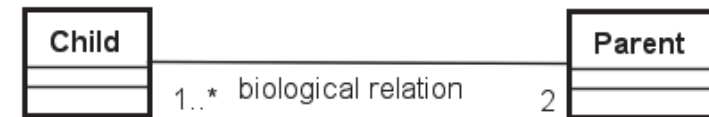
Copyright© Natsuko NODA, 2014-2024

25

Multiplicity

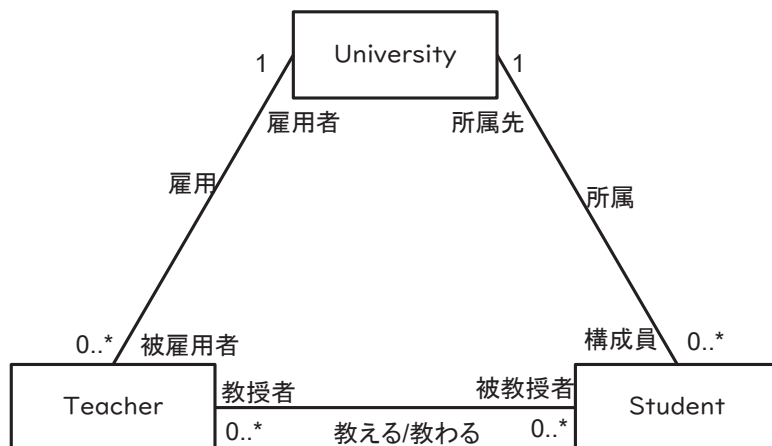
• "n" and "*" : different

- "n" - specific number
- "*" - any number
- n is often a small number
- n can be a large number. If n is a large number, still 0(1)..n is different from 0(1)..*.



Copyright© Natsuko NODA, 2014-2024

27



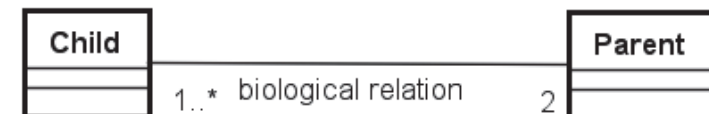
Copyright© Natsuko NODA, 2014-2024

26

多重度

• "n" と "*" : 違う

- "n" - 具体的な特定の数
- "*" - 任意の数
- n は小さい数であることが多い
- n は大きな数であってもよい。その場合も、0(1)..n と 0(1)..*は違う



Copyright© Natsuko NODA, 2014-2024

28

Multiplicity

- "0..*" and "1..*" : interchangeable?
 - Not interchangeable
 - But it may depend on the purpose of the model and the viewpoint of modeler.

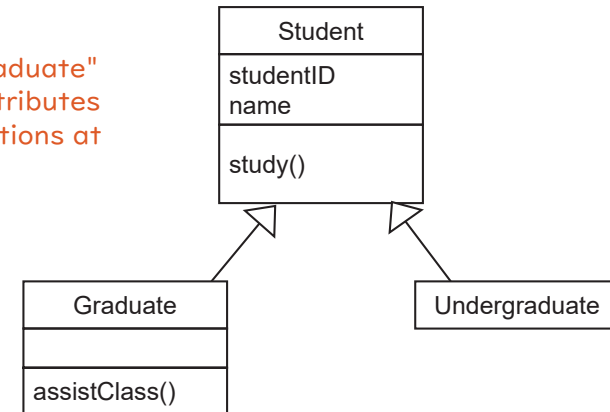
多重度

- "0..*" と "1..*" : 交換可能？
 - 交換可能ではない
 - ただし、モデルの目的やモデル作成者の視点により異なる

Exercise

- Write down all attributes and operations of each class.

Hint:
"Undergraduate"
has no attributes
nor operations at
all?



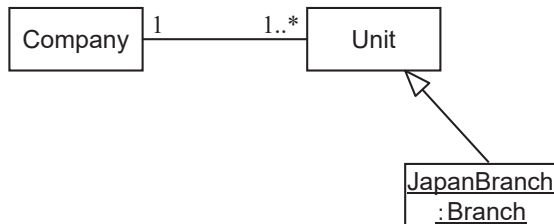
練習

- (日本語訳省略)

Exercise

- What is wrong?

Hint: "Unit" is a class.
"JapanBranch" is an object. What is a line with triangle between them?



Copyright© Natsuko NODA, 2014-2024

33

練習

- (日本語訳省略)

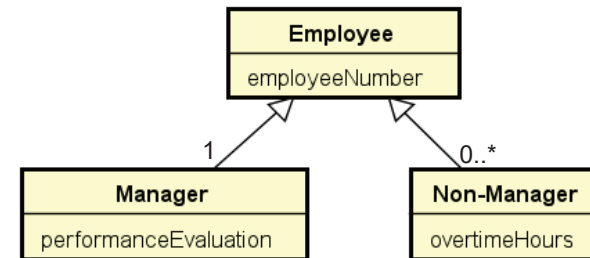
Copyright© Natsuko NODA, 2014-2024

34

Exercise

- What is wrong?

Hint: What are "1" and "0..*" at the end of the symbols between classes? What are these symbols?



Copyright© Natsuko NODA, 2014-2024

35

練習

- (日本語訳省略)

Copyright© Natsuko NODA, 2014-2024

36

Exercise

- Draw a class diagram describing the following:
 - A building consists of one or more floors (for example, ground floor, 1st floor, 2nd floor, and so on).
 - There are one or more rooms on a floor.



Copyright© Natsuko NODA, 2014-2024

37

練習

- (日本語訳省略)

Copyright© Natsuko NODA, 2014-2024

38

Activity Diagram

- The purpose of an activity diagrams is to describe one activity.
 - An activity diagram shows how what kinds of actions collaborate to realize the activity.
- An activity may be
 - a work (task) in the real world.
 - a process executed on a computer.
- Similar to flowchart, but the principal difference is that an activity diagram supports parallel behavior.

Copyright© Natsuko NODA, 2014-2024

39

アクティビティ図

- アクティビティ図の目的は、アクティビティを記述すること
 - つまり、どのようなアクションが協調してそのアクティビティを実現するのかを示す
- アクティビティの内容は様々
 - 現実世界の仕事、業務
 - コンピュータ上の処理
- フローチャートに類似している。しかし重要な違いは、アクティビティ図は並行な振る舞いを表現できること

Copyright© Natsuko NODA, 2014-2024

40

Example

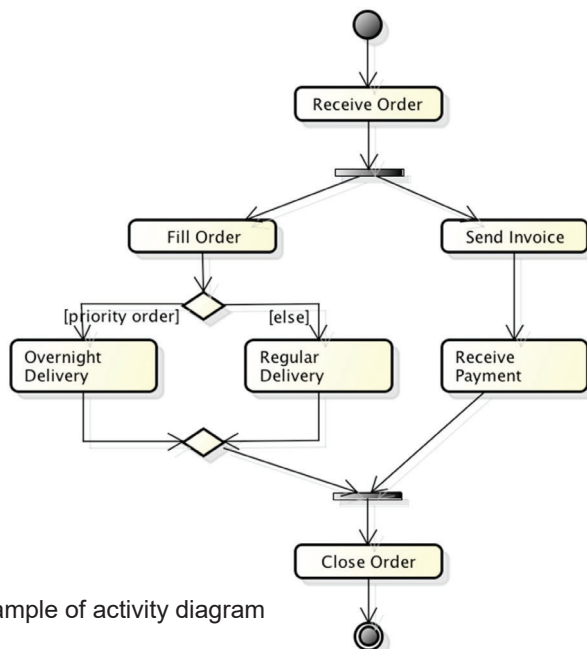


Figure 1. Example of activity diagram

Copyright© Natsuko NODA, 2014-2024

41

Two types of flow

Two types of "flow" itself

- In activity diagrams, flows may show one of the following two types.
 - Control flow
 - Flow of control
 - It shows a sequence in which actions are executed; actions are executed in the order shown by the flows.
 - Data flow
 - Flow of data
 - It shows a sequence in which data are transformed; an action transforms data of input flow (data attached to the input flow) to data of output flow.

Two types of flow may exist in an activity diagram. It is not correct that there are "control flow diagram" and "data flow diagram."

Copyright© Natsuko NODA, 2014-2023

43

例

Order: 注文
Fill order: 注文に応える
Overnight Delivery: 翌日配送
Invoice: 送り状、請求書

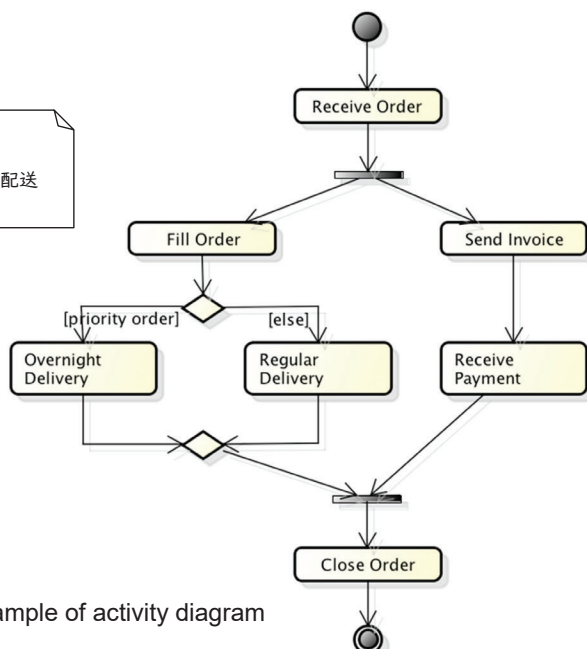


Figure 1. Example of activity diagram

Copyright© Natsuko NODA, 2014-2024

42

2種類のフロー

2種類の"フロー"がある

- アクティビティ図のフローが表すのは、次の2種類のどちらか
- 制御フロー
 - 制御の流れ
 - アクションが実行される系列を表す。つまり、アクションはフローが示す順番に実行されていく
- データフロー
 - データの流れ
 - データが変換される系列を示す。アクションは入力フローのデータを出力フローへと変換すると見なされる

1つのアクティビティ図の中に、制御フローとデータフローが存在し得る。制御フロー図とデータフロー図という2種類の図があるわけではない。

Copyright© Natsuko NODA, 2014-2023

44

"Decision-merge" and "fork-join"

- Decision-merge
 - Conditional branch.
 - Like "if statement" in source codes.
- Fork-join
 - Parallel behavior.
 - There is no "fork-join" in single thread programs.
 - Multi thread programs, multi threads (processes) can be executed in parallel.

"デシジョン-マージ"と"フォーク-ジョイン"

- デシジョン-マージ
 - 条件分岐
 - ソースコードにおける"if文"
- フォーク-ジョイン
 - 平行な振舞い
 - シングルスレッドのプログラムでは、"fork-join"に当たるものは出てこない
 - マルチスレッドプログラムにおいては、複数のスレッド(プロセス)が並行に実行され得る

For your review

1. What does an activity diagram depict?

2. Which statement is true?

- a. Activity diagrams are used only for expression of program logic.
- b. Activity diagrams are used only for business modeling.
- c. Activity diagrams are used only for workflow design.
- d. Activity diagrams can be used for describing and designing various flows in various phases in software development.

For your review

1. アクティビティ図が示すのは何か?

2. どれが正しい?

- a. アクティビティ図はプログラムのロジックを示すためだけに使われる
- b. アクティビティ図はビジネスモデリングだけに使われる
- c. アクティビティ図はワークフローの設計だけに使われる
- d. アクティビティ図はソフトウェア開発の様々な工程における様々なフローの記述や設計に用いることができる

For your review

3. What is the similarity between flowcharts and activity diagrams? What is the principal difference between them?

4. What is an activity? What is an action?

For your review

3. フローチャートとアクティビティ図の類似しているところは？また本質的な違いは？

4. アクティビティとは？アクションとは？

Sequence Diagram

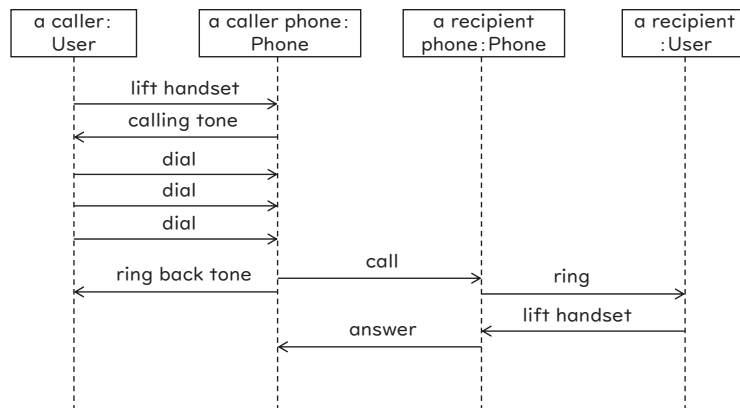
- A sequence diagram shows interactions of objects; these interactions are described by sets of messages passed between objects, which are ordered chronologically.
- Message: communication from sender to receiver
 - message may be sending a signal, calling an operation, and so on.
- A sequence diagram shows just one example of behavior.
 - under the specific context

シーケンス図

- シーケンス図はオブジェクトのやり取りを示す。このやりとりは、オブジェクト間で交換されるメッセージで表現され、それらのメッセージは時系列に沿って並べられる
- メッセージ：送信側から受信側への何らかの通信
 - シグナルの送信、操作の呼び出し、等々
- シーケンス図は、振舞いのひとつの例を示す
 - ある特定の状況下における振舞い例

Sequence diagram

- A simple example:



Copyright© Natsuko NODA, 2014-2024

53

For your review

- Which is correct?
Sequence diagram is ...

A)

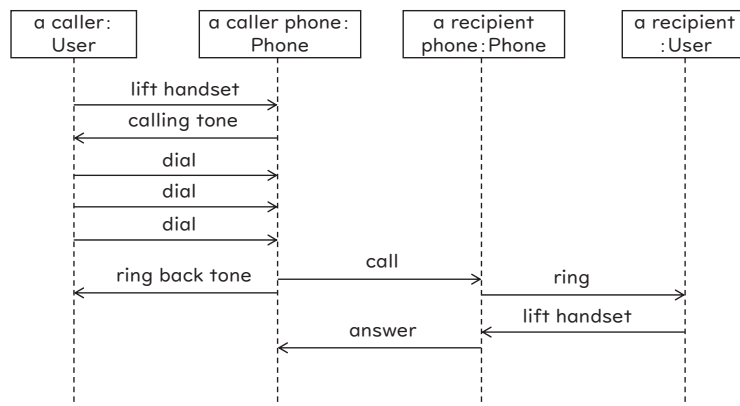
1. a diagram defining a structure of an aspect of the target.
2. a diagram showing an example of an aspect of the target.

Copyright© Natsuko NODA, 2014-2024

55

シーケンス図

- 単純な例:



Copyright© Natsuko NODA, 2014-2024

54

For your review

- (日本語訳省略)

Copyright© Natsuko NODA, 2014-2024

56

For your review

1. What does a sequence diagram depict?

- a. The sequence of external objects communicating with each other
- b. Classes and their relationships
- c. Software objects and the sequence of their interactions
- d. The external objects communicating with the system

2. Is the following explanation correct?

Answer Yes or No.

- A sequence diagram defines the behavior of a system. →

State machine diagram

- It describes the behavior of a target entity as a finite state machine.
- Finite states and transitions between them.

For your review

1. シーケンス図が表すものは?

- a. 互いにやり取りする外部オブジェクトの系列
- b. クラスとその関連
- c. ソフトウェアオブジェクトとその間のやりとりの系列
- d. システムとやり取りする外部オブジェクト群

2. 次の説明は正しいか? Yes か No を答えよ

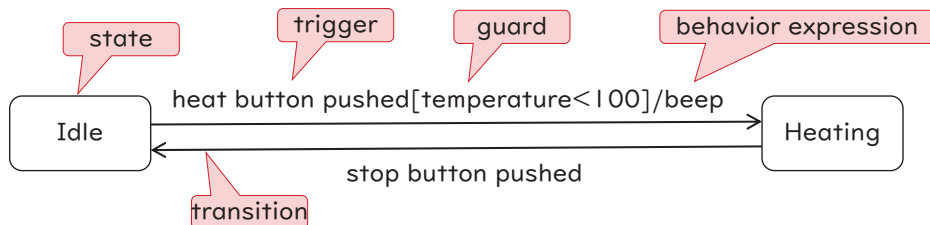
- シーケンス図はシステムの振舞いを定義する →

ステートマシン図

- ステートマシン図は、対象の振舞いを 有限状態機械として表現する
- 有限個の状態と、その間の遷移

Notation

- State: rounded rectangle. The name of the state is shown in the rectangle.
- Transition: arrow connecting states.



When an event that triggers the transition occurs and the guard condition is true at that time, the transition is taken.

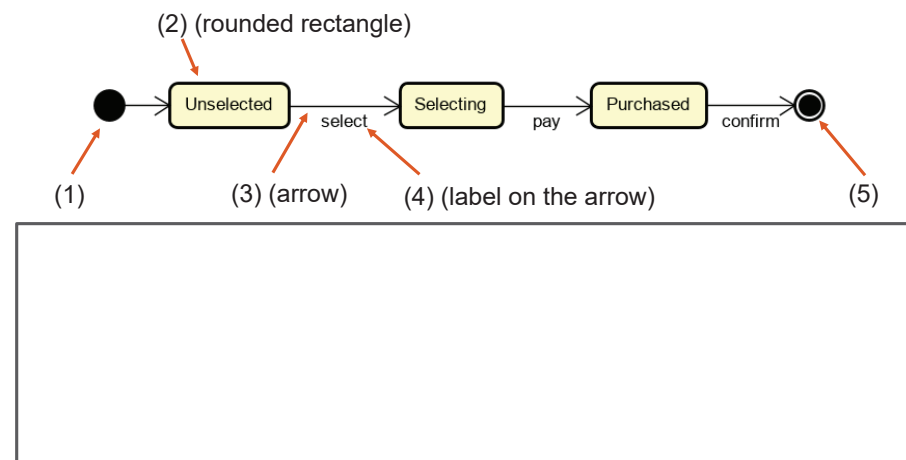
If a transition label has no trigger signature and has a guard, the transition is taken when the guard condition becomes true.

Copyright© Natsuko NODA, 2014-2024

61

For your review (cont.)

- What is this? Answer the name of symbols.

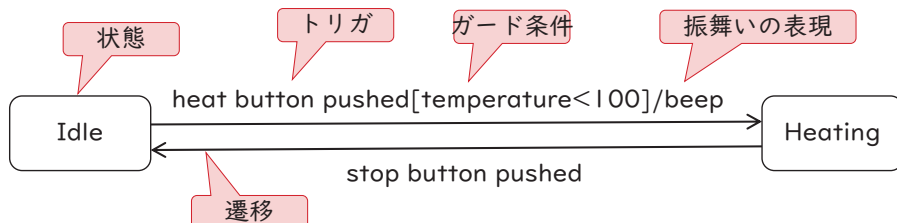


Copyright© Natsuko NODA, 2014-2024

63

記法

- 状態: 角丸四角形. 状態名が角丸四角形の中に示される
- 遷移: 状態を結ぶ矢印



遷移を引き起こすトリガが起こり、かつガード条件が真の場合に、遷移が実行される。

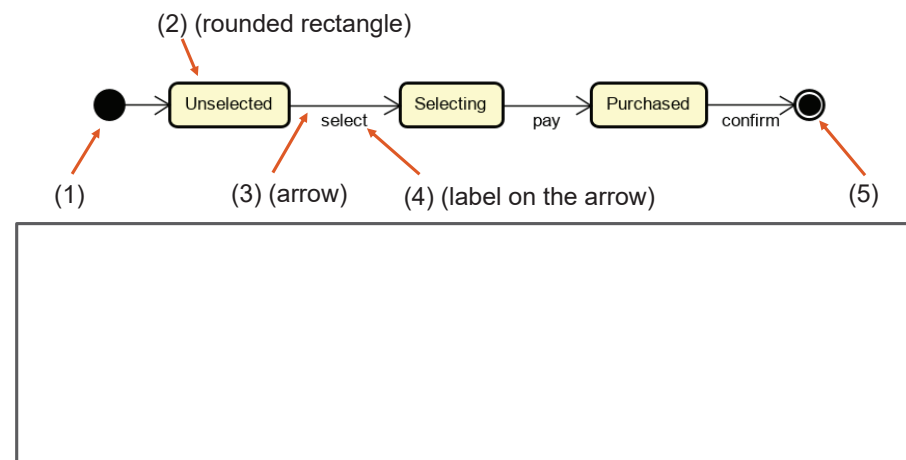
もし遷移にトリガがなくガード条件だけ示されていた場合には、ガード条件が真になった時に遷移が実行される。

Copyright© Natsuko NODA, 2014-2024

62

For your review (cont.)

- What is this? Answer the name of symbols.

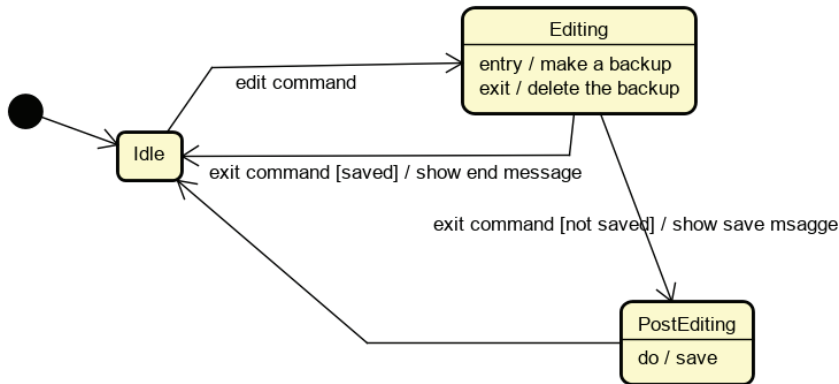


Copyright© Natsuko NODA, 2014-2024

64

For your review

- Answer the questions about the figure.



Copyright© Natsuko NODA, 2014-2024

65

- (日本語訳省略)

Copyright© Natsuko NODA, 2014-2024

66

For your review (cont.)

- What is this diagram called? (What is this?)
- Which state is the initial state of the system described by this diagram?
- In this diagram, no final state is shown. Is it OK? If not, show one example with a final state.
- The current state is "Editing", and "not saved" yet. If an "exit command" occurs, which is the next state? And list the actions and activities to be executed till the system reaches "Idle" in the order of execution.

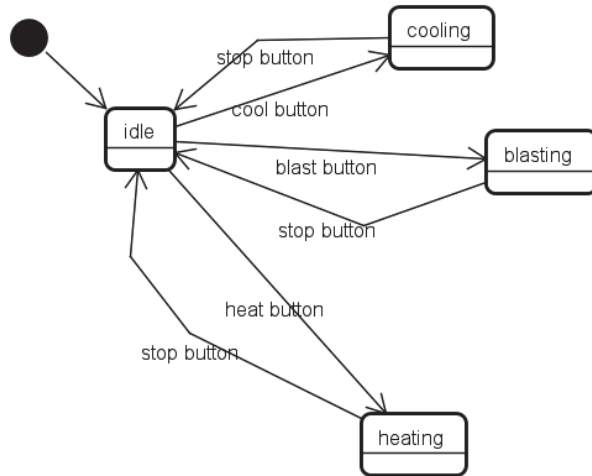
- (日本語訳省略)

Copyright© Natsuko NODA, 2014-2024

68

Exercise

- Revise this using "composite state".



Copyright© Natsuko NODA, 2014-2024

69

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.

Copyright© Natsuko NODA, 2014-2024

73

- (日本語訳省略)

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

Copyright© Natsuko NODA, 2014-2024

70

Copyright© Natsuko NODA, 2014-2024

74

Use case

- A use case is 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

75

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

77

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

76

- (日本語訳省略)

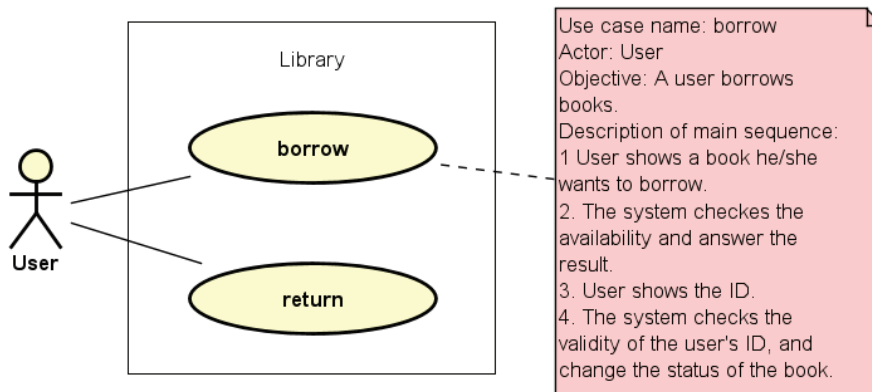
Copyright© Natsuko NODA, 2014-2024

78

Use case diagram

- Too simple? (No meaning?)

- It is very important to consider what kind services are needed and who interact with the service.
- We need use case descriptions along with the use case diagram.
- No specific design information.



- (日本語訳省略)

ユースケース図

- シンプル過ぎる？(使えない？)

- どんなサービスが必要で、誰がこのシステムを使うか、を検討することは非常に重要
- ユースケース図とともにユースケース記述が必要になる
- 特定の設計情報を含むわけではない

For your review

- Describe the following behavior of "fan" using an appropriate diagram.
 - When the start button is pushed, the fan starts running.
 - If the fan is running and the stop button is pushed, the fan stops.



- Describe the behavior of air conditioners.

- (日本語訳省略)