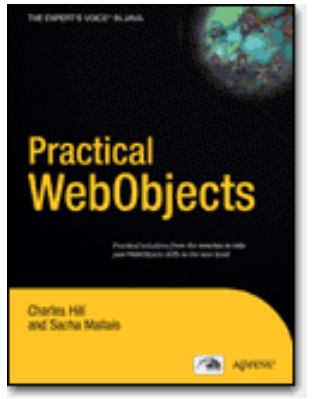
Practical Webobjects
Chapter 9 (Page 237-270):



Copying Enterprise Objects

WR

WR at Csus4.net
http://www.csus4.net/

目次

- Understanding the Problem
  - OOP一般におけるコピーの難しさ
- Types of Copies
  - OOP一般におけるコピーの種類
- Naive Copy
  - 単純にEOのアトリビュートをコピーしてもうまくいかない
- EOEnterpriseObject Metadata
  - EOのメタデータ (構造データ)
- EOModel-Based Copy
  - メタデータに基づくコピー
- EOCopyable Interface
  - 柔軟なコピーを実現する実装
- Object Migration
  - EOModelベースのデータ移行
- Summary

目次

- Understanding the Problem
  - OOP一般におけるコピーの難しさ
- Types of Copies
  - OOP一般におけるコピーの種類
- Naive Copy
  - 単純にEOのアトリビュートをコピーしてもうまくいかない
- EOEnterpriseObject Metadata
  - EOのメタデータ (構造データ)
- EOModel-Based Copy
  - メタデータに基づくコピー
- EOCopyable Interface
  - 柔軟なコピーを実現する実装
- Object Migration
  - EOModelベースのデータ移行
- Summary

まずは問題を定義しましょう。

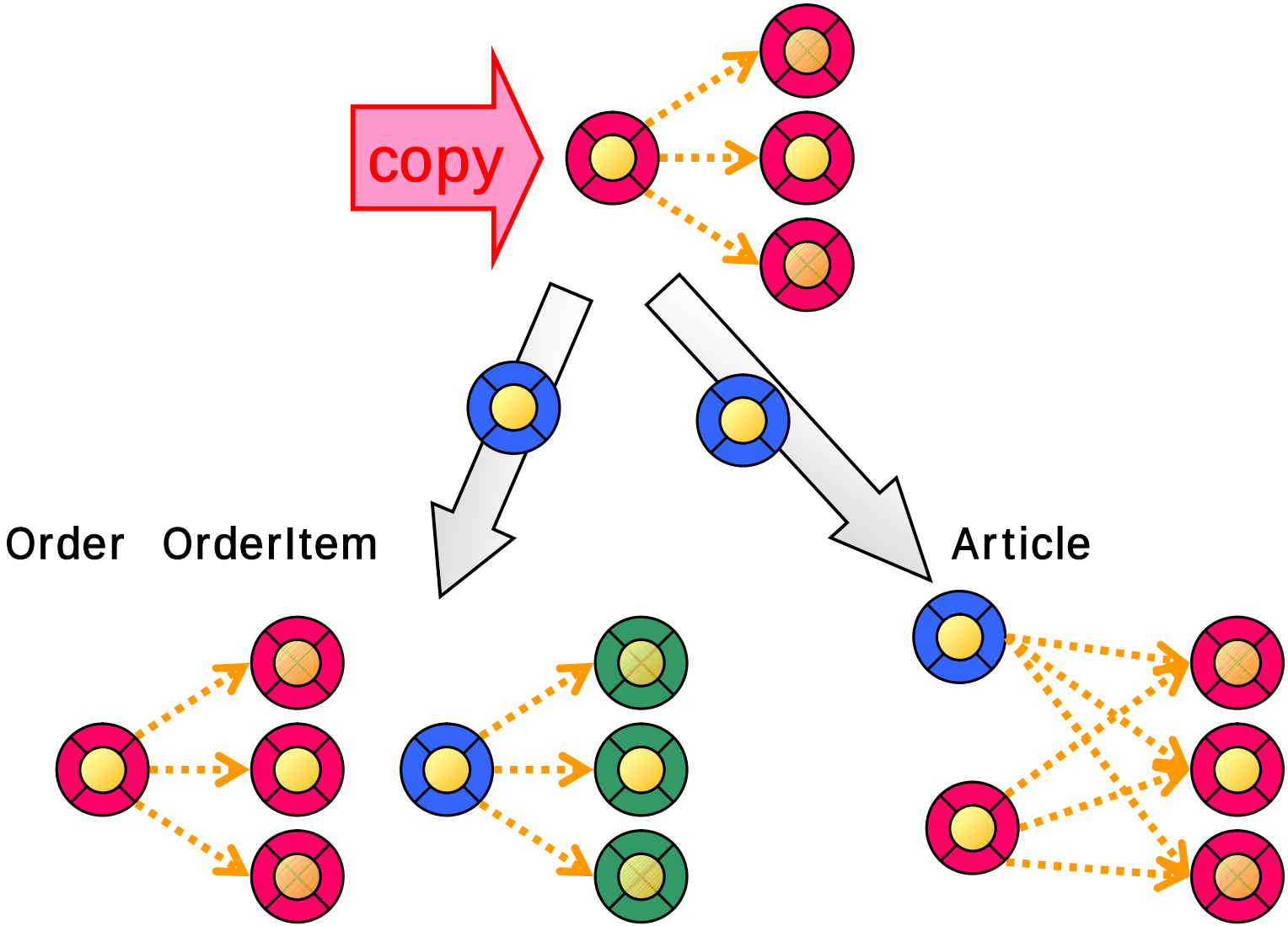
P237-238．Understanding　the Problem

- コピーすべきは、属性とリレーションシップ

- 属性
  - 「単純に属性をコピーすればよい。」というわけではない
  - コピーしてはいけない属性もある
    - プライマリ・キー
    - サロゲート・キー（代理キー）
- リレーションシップ
  - 適切なコピーのしかたが一概に判断できない（後述）
  - リレーションシップが循環していると、単純なコピーではうまく行かない

リレーションシップのコピー：どちらが適切?

P237-238. Understanding the Problem



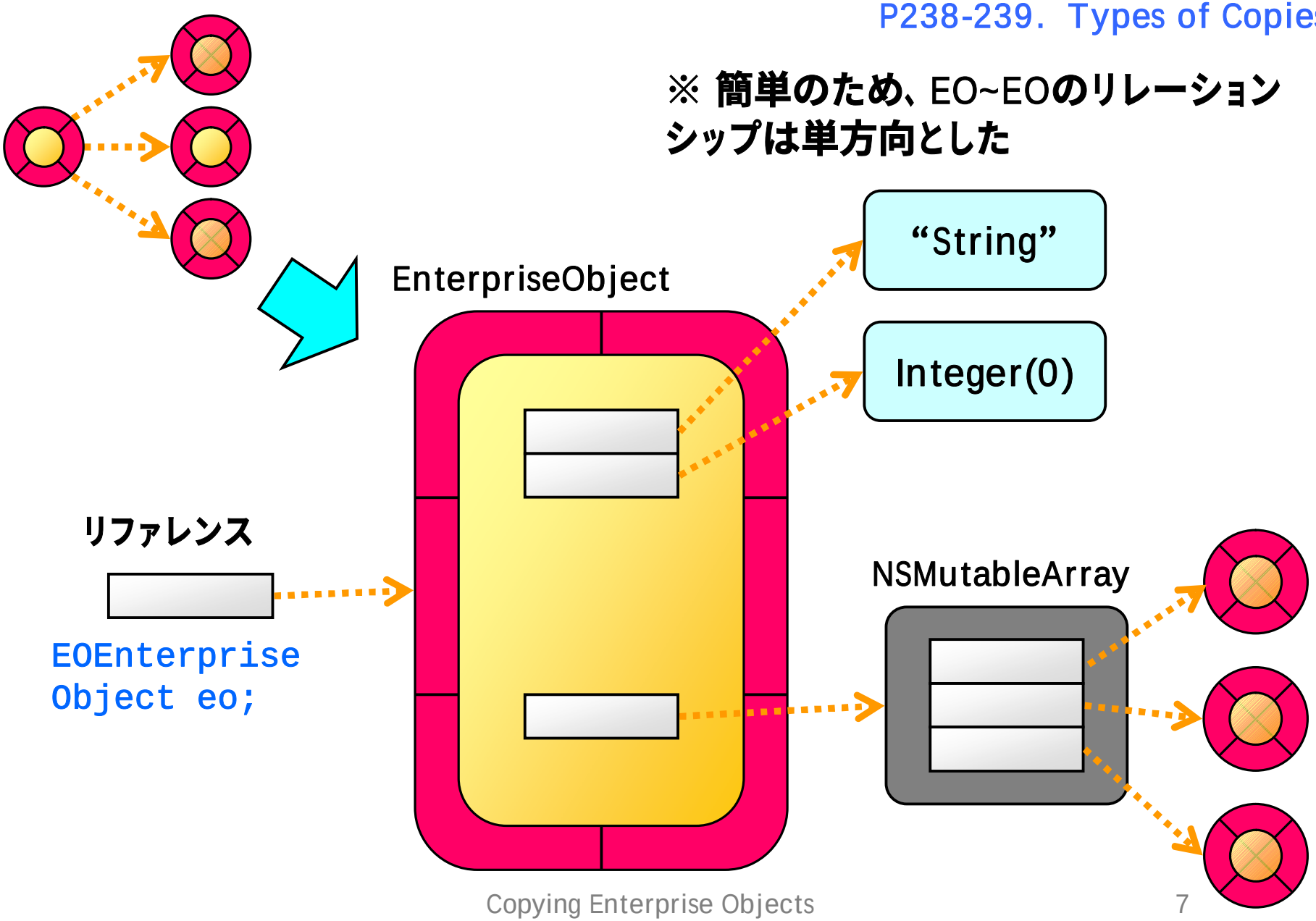
目次

- Understanding the Problem
  - OOP一般におけるコピーの難しさ
- Types of Copies
  - OOP一般におけるコピーの種類
- Naive Copy
  - 単純にEOのアトリビュートをコピーしてもうまくいかない
- EOEnterpriseObject Metadata
  - EOのメタデータ (構造データ)
- EOModel-Based Copy
  - メタデータに基づくコピー
- EOCopyable Interface
  - 柔軟なコピーを実現する実装
- Object Migration
  - EOModelベースのデータ移行
- Summary

コピーを論じる前に、物理的なイメージを・・・

P238-239．Types of Copies

※ 簡単のため、EO~EOのリレーションシップは単方向とした



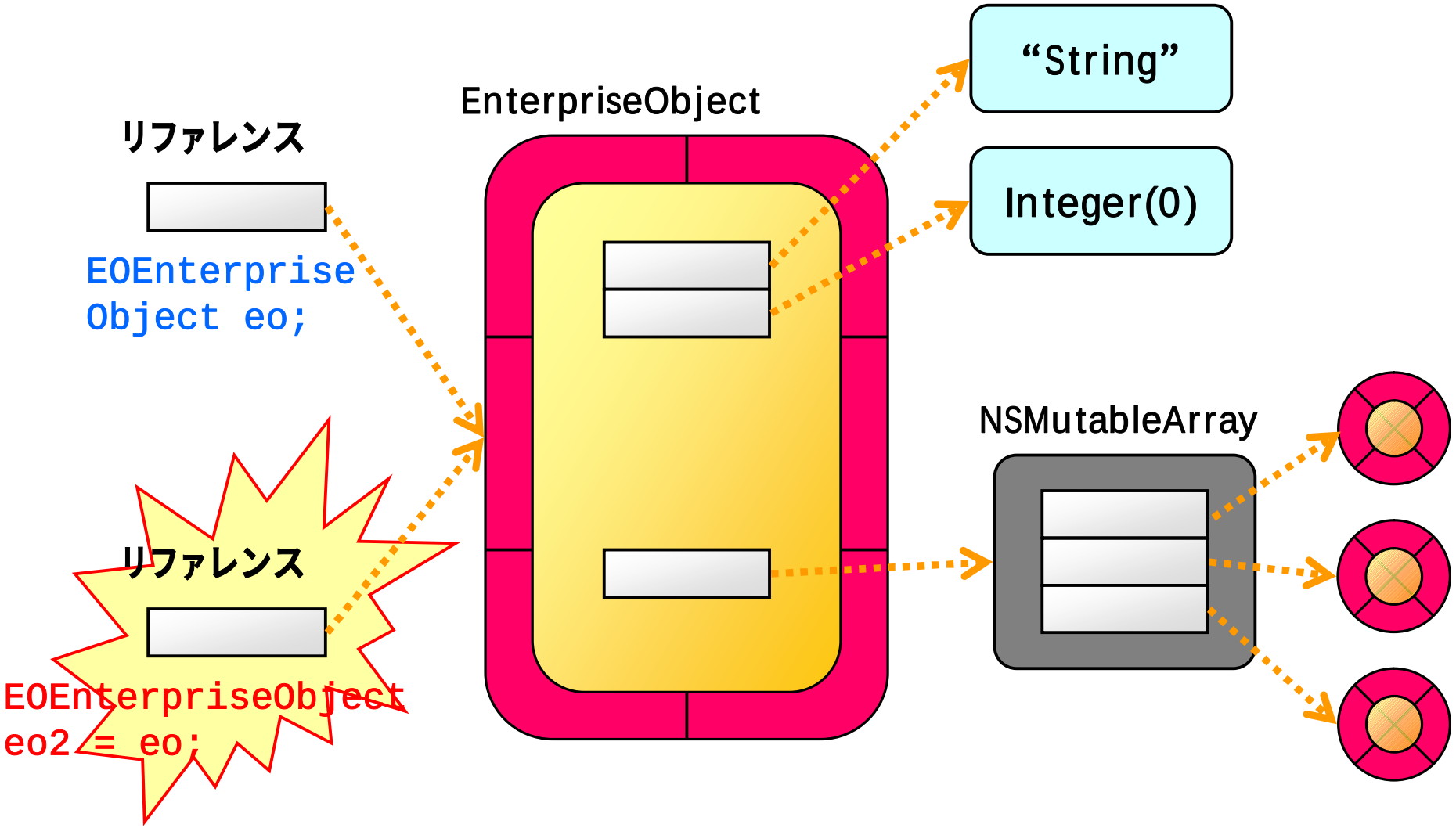
コピーの種類

P238-239．Types of Copies

種類	説明	新しいオブジェクトの生成
参照 Reference	オブジェクトの参照のみをコピーする	×
浅いコピー Shallow	元のオブジェクトの属性をもつ新しいオブジェクト＋関連先オブジェクトへの参照のコピー	1つ生成
深いコピー Deep	元のオブジェクトの属性をもつ新しいオブジェクト＋関連先オブジェクトのコピー	複数生成

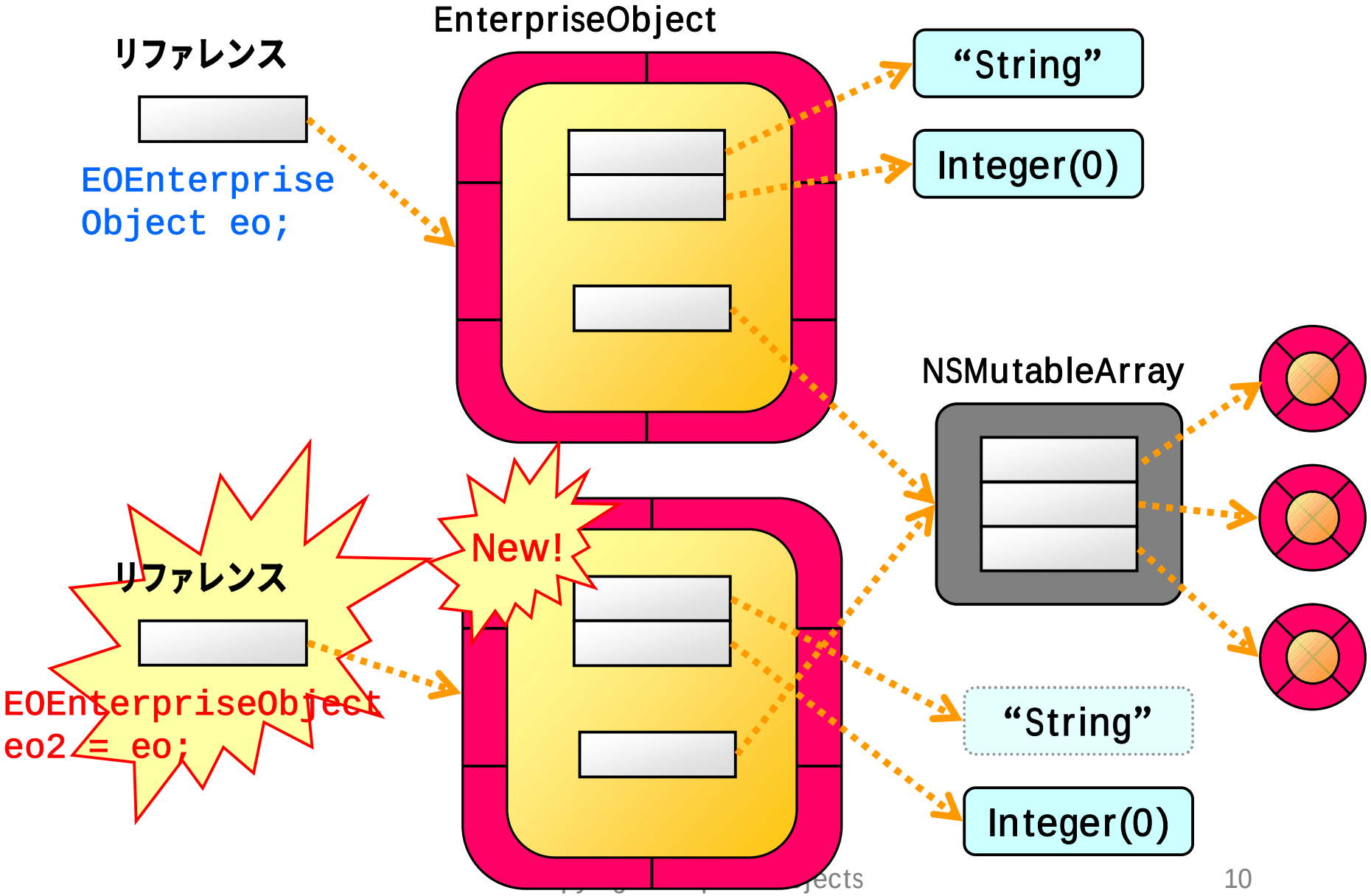
参照のコピー

P238-239. Types of Copies



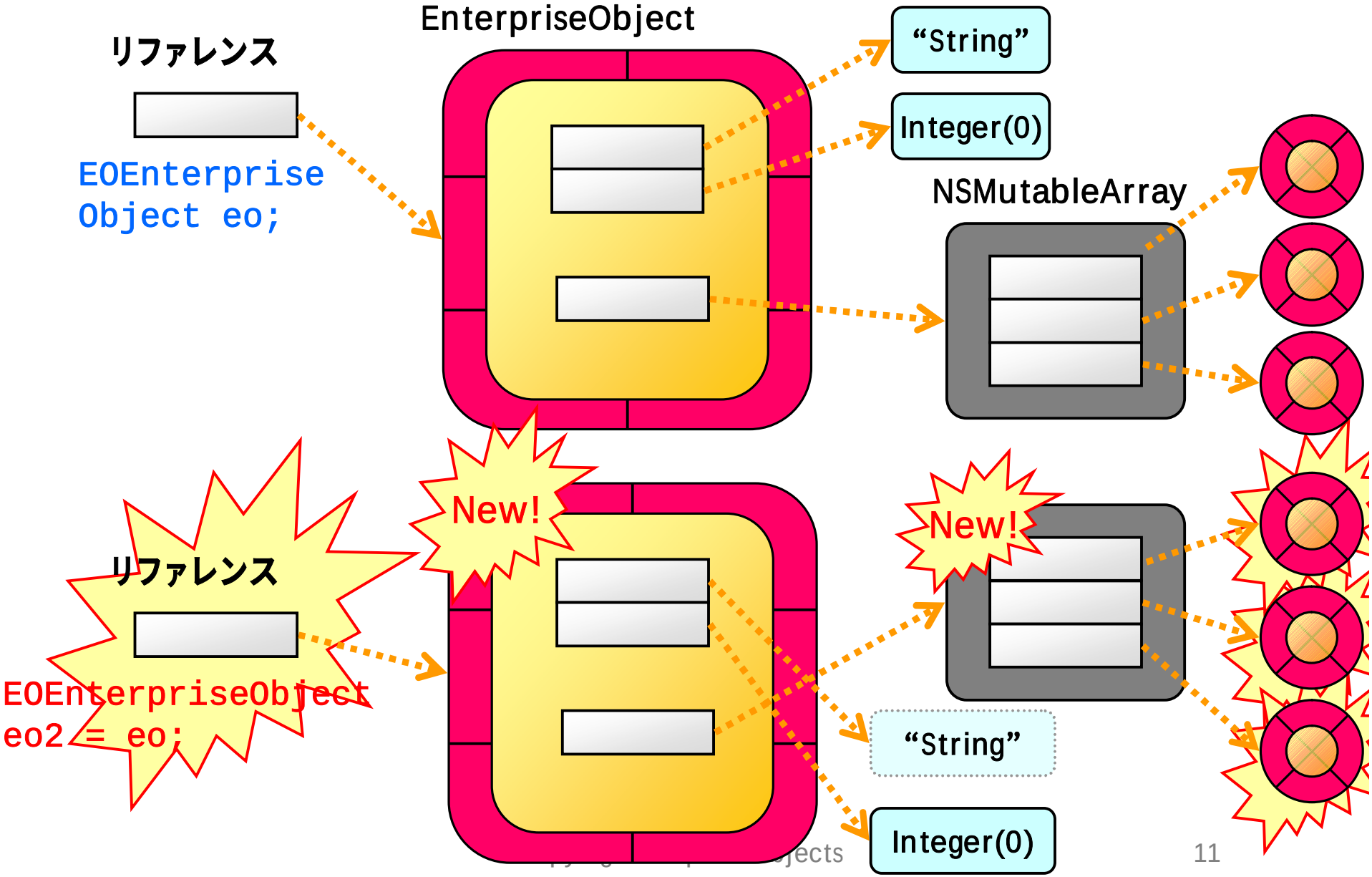
浅いコピー (Shallow Copy)

P238-239. Types of Copies



深いコピー (Deep Copy)

P238-239. Types of Copies



目次

- Understanding the Problem
  - OOP一般におけるコピーの難しさ
- Types of Copies
  - OOP一般におけるコピーの種類
- Naive Copy
  - 単純にEOのアトリビュートをコピーしてもうまくいかない
- EOEnterpriseObject Metadata
  - EOのメタデータ (構造データ)
- EOModel-Based Copy
  - メタデータに基づくコピー
- EOCopyable Interface
  - 柔軟なコピーを実現する実装
- Object Migration
  - EOModelベースのデータ移行
- Summary

なんかよくわかんないけど、こうすればいいんじゃん？

P239．Naive Copy

```
EOEnterpriseObject copy =
  EOUtilities.createAndInsertInstance(
      ec,originalEO.entityName());
copy.takeValuesFromDictionary(
  ec.committedSnapshotForObject(orignalEO));
```

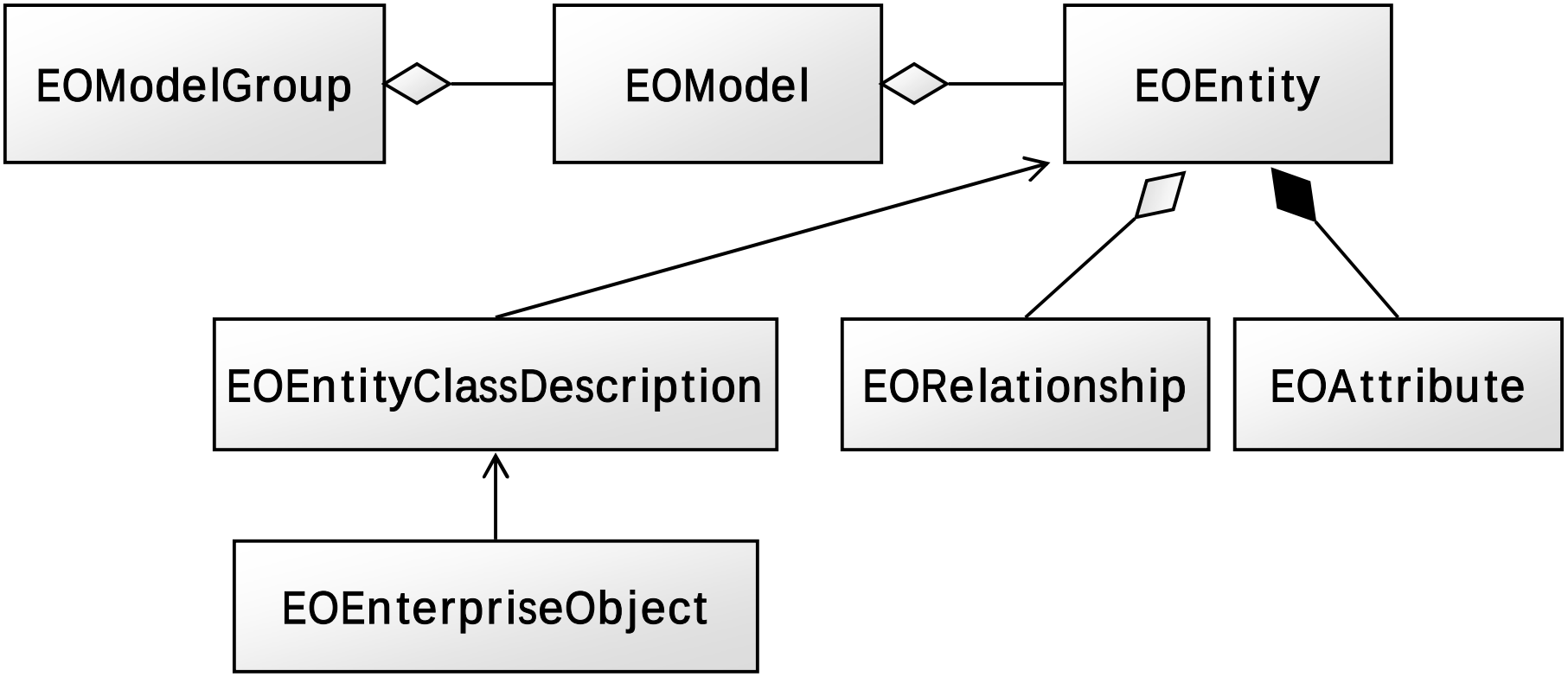
- んなわきゃない。
  1. プライマリキーもコピーされるため、コミットできない
  2. 深いコピーができない
  3. 元のオブジェクトの属性値=nullの場合、awakeFromInsertion()内で設定されたデフォルト値が上書きされない
- もうちょっと真面目にやらないとダメだね・・・。

目次

- Understanding the Problem
  - OOP一般におけるコピーの難しさ
- Types of Copies
  - OOP一般におけるコピーの種類
- Naive Copy
  - 単純にEOのアトリビュートをコピーしてもうまくいかない
- EOEnterpriseObject Metadata
  - EOのメタデータ(構造データ)
- EOModel-Based Copy
  - メタデータに基づくコピー
- Object Migration
  - EOModelベースのデータ移行
- Summary

EOEnterpriseObject メタデータ

P240．EOEnterpriseObject Metadata



- これから、メタデータベース（モデルベース）のコピーのやり方を模索してゆく。

目次

- Understanding the Problem
  - OOP一般におけるコピーの難しさ
- Types of Copies
  - OOP一般におけるコピーの種類
- Naive Copy
  - 単純にEOのアトリビュートをコピーしてもうまくいかない
- EOEnterpriseObject Metadata
  - EOのメタデータ (構造データ)
- EOModel-Based Copy
  - メタデータに基づくコピー
- EOCopyable Interface
  - 柔軟なコピーを実現する実装
- Object Migration
  - EOModelベースのデータ移行
- Summary

Owns Destinationとコピーの種類

P242-243．EOModel-based Copy

- EOModeler上でリレーションシップが “Owns Destination”と設定されている・・・
  - リレーションシップの意味的には「コンポジション」
    - 例) [Order]◆-[OrderItem]
  - 深いコピー(Deep Copy)が適切な場合が多い

- 深いコピーの実現
  - PracticalUtilities.frameworkの EOEntityCopier#deepCopy(ec, origEo)を使え!

どのプロパティをコピーすべきか？

P242-243．EOModel-based Copy

- 属性
  - クラスプロパティ(EOModelerで“◆”マークをつけたもの)の属性のみコピーすればよい
  - クラスプロパティな属性を得る方法はP243を参照
    - eo#attributes() ∧ eo#classProperties()
- リレーションシップ
  - クラスプロパティなリレーションシップをコピーすればよい
    - eo#relationships() ∧ eo#classProperties()
  - ただし、to-Oneとto-Manyではコピー方法が異なる

リレーションシップのコピー

P242-243．EOModel-based Copy

- **リレーションシップ** = to-One
  - Owns Desitination = YES
    - → Deep Copy & addObjectsToBothSidesOfRelationshipWithKey()
  - Owns Desitination = NO
    - → Shallow Copy & addObjectsToBothSidesOfRelationshipWithKey()
- **リレーションシップ** = to-Many
  - **（上記と同様の処理を複数回実施・・・）**

モデルベースのコピーの限界

P242-243．EOModel-based Copy

- さて、この方法で万事うまく行くか？
- それはムリ。具体的には・・・
  - Owns Destination指定アリ→ 無条件にDeep Copyというワケではない
  - リレーションシップの循環に対応できない
  - 別のecにコピーできない
  - 特定の属性をawakeFromInsertion()で設定されるデフォルト値としたいケースもある
    - 例) dateCreated
  - awakeFromInsertion()で作成された関連オブジェクトをコピー前に削除する必要がある場合がある

EOCopyable Interface

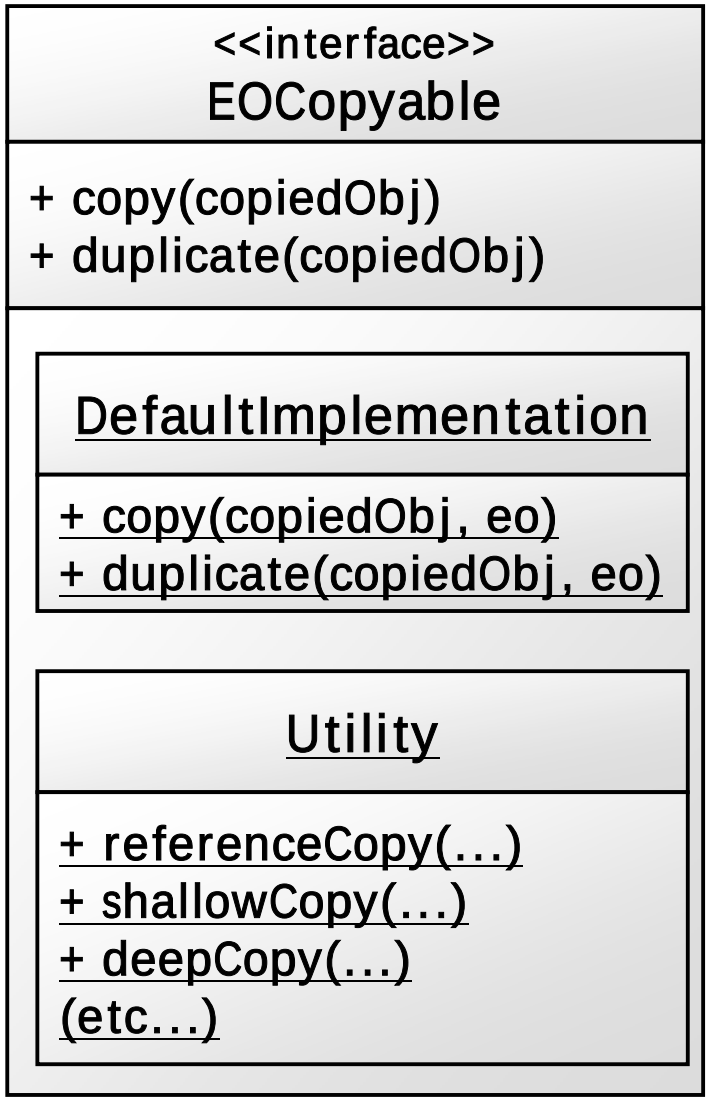
P246．EOCopyable Interface

- モデル情報を元に自動的にコピー戦略を決定するアプローチ → 限界アリ

- EOCopyable Interfaceが提供するインタフェースと、EOCopyable.DefaultImplemantation, EOCopyable.Utilityが提供する実装を好き勝手に使って、あなたが適切と思うコピー戦略を構築しては？
  - という提案 from Charles Hill

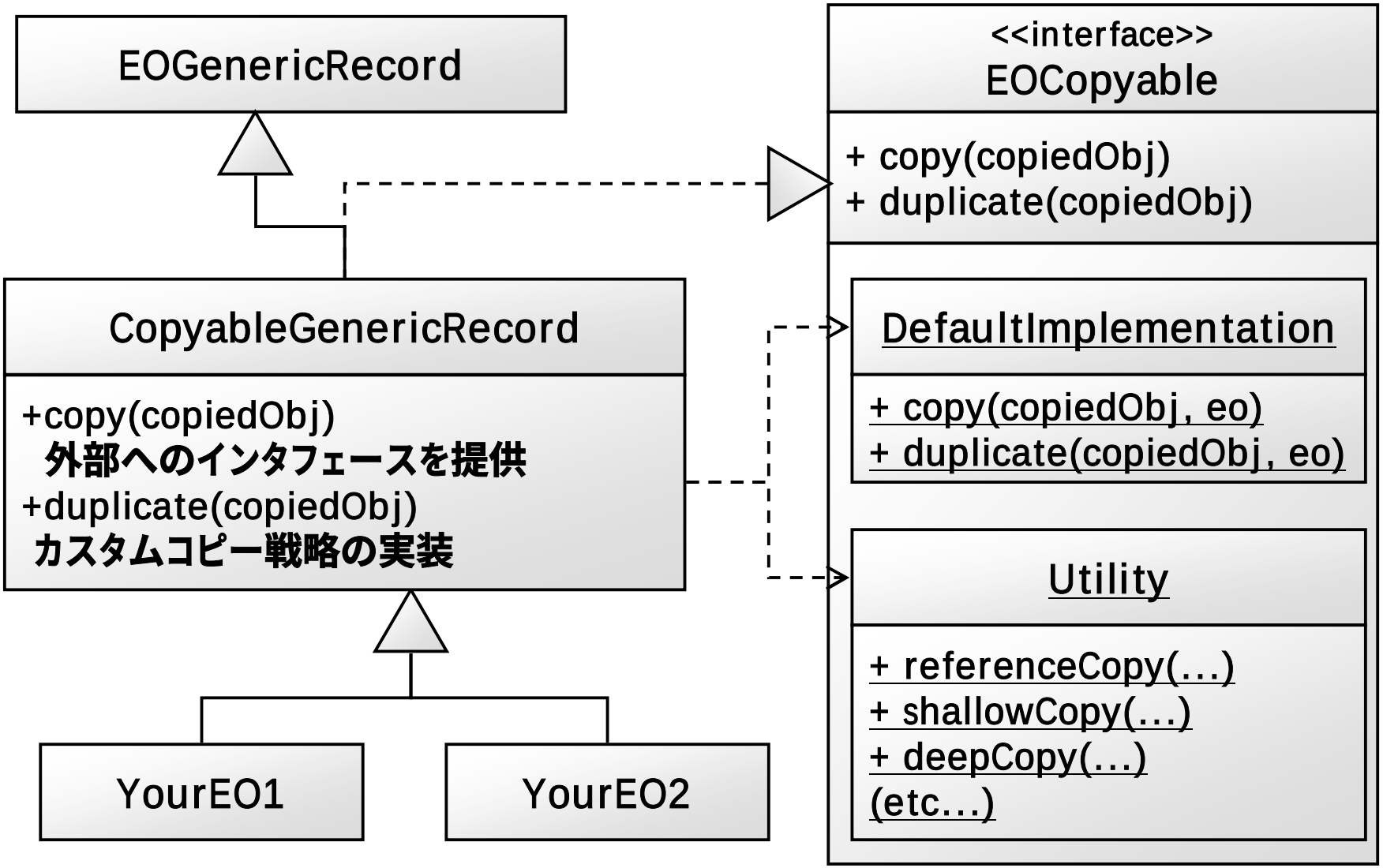
EOCopyableインタフェース

P246．EOCopyable Interface



EOCopyableインタフェース UseCase

P250．Making Use of the Default Implementation



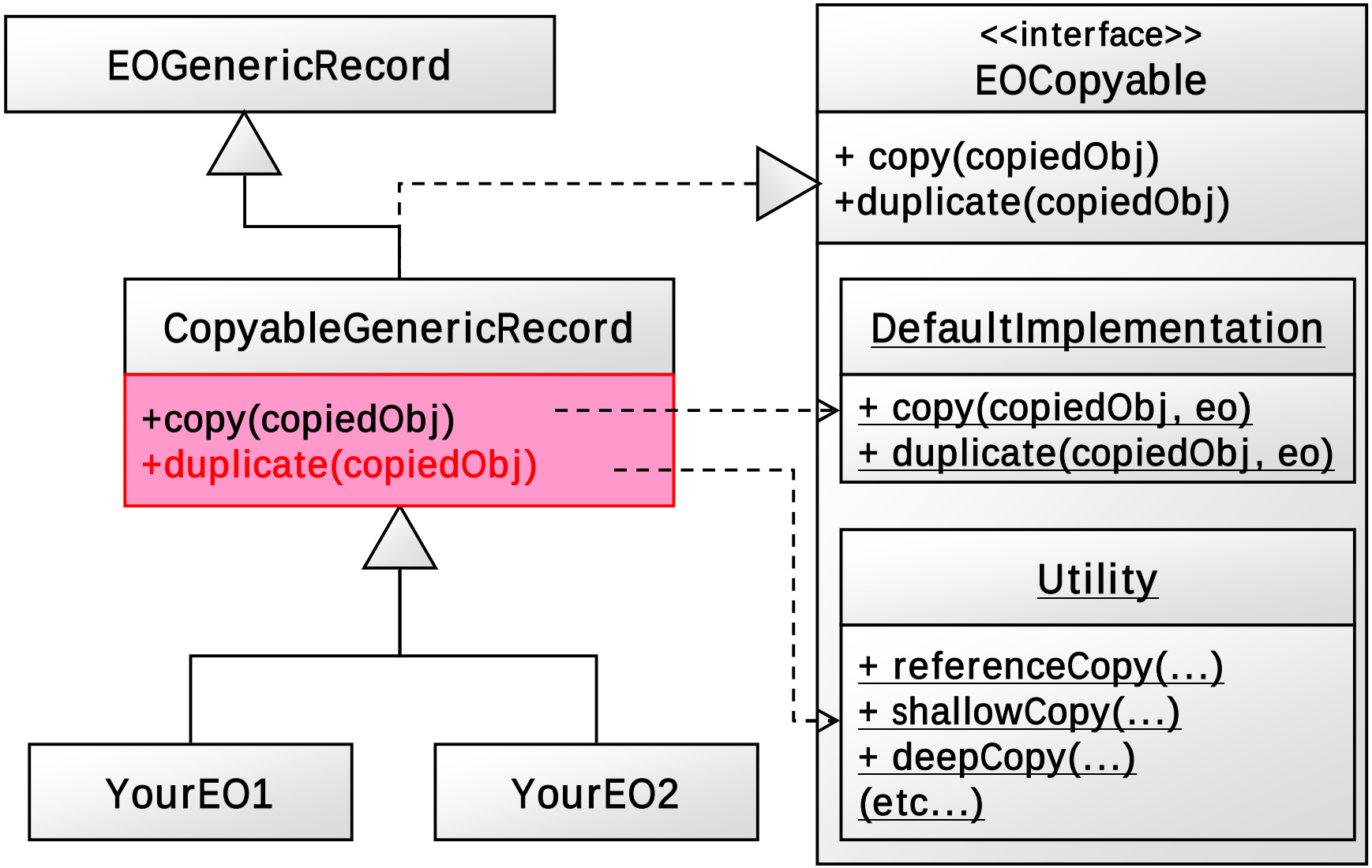
EOCopyable.DefaultImplimentation実装

P248-249．The EOCopyable Default Implementation

- **copy** (NSMutableDictionary copiedObjects, EOEnterpriseObject source) : EOEnterpriseObject
  - sourceがcopiedObjectsに存在したら(= コピー済みならば)
    - コピー処理は行わず、それを返す
  - sourceがcopiedObjectsに存在しなかったら(= 未コピーならば)
    - source#duplicate(copiedObjects)を呼び出す
    - copiedObjectsにsourceを追加(=コピー済みとマーク)
- **duplicate** (NSMutableDictionary copiedObjects, EOEnterpriseObject source) : EOEnterpriseObject
    - Utility#deepCopy(copiedObjects, source)に委譲

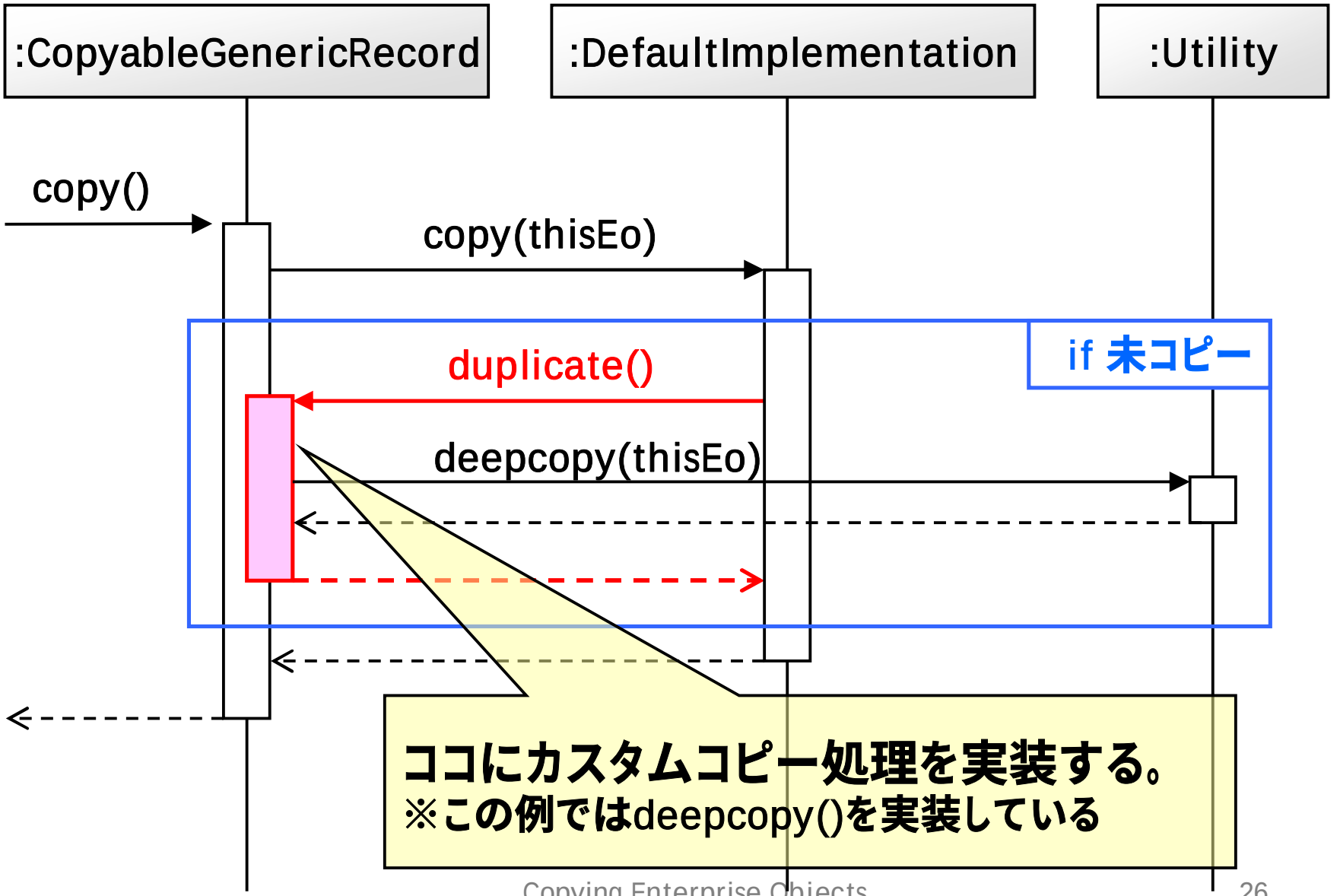
カスタムコピー戦略を実装する

P250．Making Use of the Default Implementation



典型的な実装例

P250．Making Use of the Default Implementation



カスタムコピー処理の実装方法3つ

P250-. The EOCopyable Utility Methods

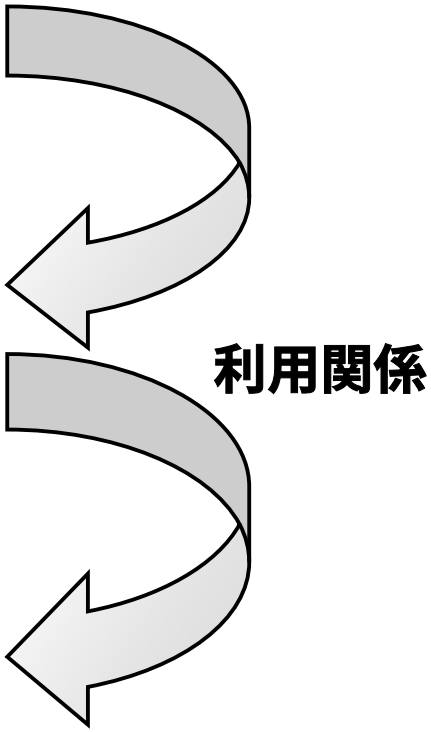
1. EOCopyable.Utilityの提供するメソッドを単に呼び出して済ます
   - 前のスライドの事例（duplicate()内でdeepcopy()を呼び出し）に相当
2. EOCopyable.Utilityのメソッドを呼び出した後に、適宜修正を加える
   - 修正の例）デフォルト属性値の設定、不要なリレーションシップの削除
3. 完全に自前でゴリゴリ コーディング!
   - コピー戦略が非常に複雑な場合

- ま、どの戦略をとるにせよ、EOCopyable.Utilityのメソッドの機能を知る必要があるよね・・・

EOCopyable.Utilityの提供するメソッド

P250-. The EOCopyable Utility Methods

- **様々なコピーメソッド**
  - referenceCopy
  - shallowCopy
  - deepCopy
- **ユーティリティ的メソッド**
  - newInstance
  - cleanRelationships
  - copyAttributes
  - exposedKeyAttributeNames
- **リレーションシップ先のコピーメソッド**
  - shallowCopyRelatedObjects
  - deepCopyRelatedObjects
  - deepCopyRelationship



コピーメソッドと処理内容

P250-. The EOCopyable Utility Methods

メソッド	処理内容
reference()	与えられた参照を返すだけ
shallowCopy()	newInstance() copyAttributes() shallowCopyRelatedObjects()
deepCopy()	newInstance() copiedObjectsに自身を登録 copyAttributes() deepCopyRelatedObjects() deepCopyRelationship()

- リレーションシップを辿ってコピーする場合は、循環による重複コピーを防ぐ必要があることに注意！

コピー処理のカスタマイズ例

P264-. Tweaking the Copy Process

- コピー処理をカスタマイズしたい場合
  - 典型的なduplicateメソッド実装を元に自前のロジックを組み込めばよい

- P264 Listing 9-15
  - 属性dateCreated, dateLastModifiedを現在時刻で上書きする

目次

- Understanding the Problem
  - OOP一般におけるコピーの難しさ
- Types of Copies
  - OOP一般におけるコピーの種類
- Naive Copy
  - 単純にEOのアトリビュートをコピーしてもうまくいかない
- EOEnterpriseObject Metadata
  - EOのメタデータ (構造データ)
- EOModel-Based Copy
  - メタデータに基づくコピー
- EOCopyable Interface
  - 柔軟なコピーを実現する実装
- Object Migration
  - EOModelベースのデータ移行
- Summary

まとめ

- 3種類のEOのコピー方法を提示した
  - Reference, Shallow, Deep
  - 貴方のニーズに合うのがきっとあるはず！
- EOModelメタデータを保持するクラスを紹介した
  - これらのクラスはコピー以外の用途において、もっと役に立つ
- いくつかのオブジェクトmigration方法を見た