

# ソフトウェア工学

## 第3回

土田 隼之

週	授業内容・方法	週ごとの到達目標
1週	ソフトウェアの性質と開発	ソフトウェア開発の特徴および課題について少なくとも一つ上げられ、その理由を言える。
2週	ソフトウェア開発プロセス	複数の開発プロセスモデルを挙げ、それぞれの特徴を言える。
3週	要求分析	要求分析とプロトタイピングの関係性や有用性について言える。
4週	ソフトウェア設計	モジュールの結合度の低い場合と高い場合のモジュール間の依存性について述べ、モジュール結合度の低い具体例を言える。
5週	プログラミングとテスト	誤り混入をさせないためのプログラミング手法およびテスト効率を向上させる技法について言える。
6週	テストと保守	保守容易性を確保するための方策について、考察し、述べることができる。
7週	グループワーク	前半6週に関する課題を、グループワークで取り組む。
8週	中間試験	前半に習得した項目について確認する。
9週	オブジェクト指向 1	身の回りのモノに関して、クラスとインスタンスという言葉を用いて説明できる。
10週	オブジェクト指向 2	オブジェクト指向プログラミングの特徴について言える。
11週	ソフトウェア再利用	ソフトウェア再利用の重要性とその困難さについて言える。
12週	プロジェクト管理	プロジェクト管理の重要性を述べることができる。
13週	品質管理	品質管理手法について言える。
14週	ソフトウェア開発規模と見積もり	ソフトウェア開発規模の見積もり手法について言える。
15週	グループワーク	後半6週に関する課題を、グループワークで取り組む。
16週	期末試験	後半に習得した項目について確認する。

# 今日の内容

## 1) 要求分析のおさらい、要求の表現

#何をこのソフトウェアで行うのか明らかにする

## 2) 要求の表現、UML

課題3-1,3-2) ユースケース図を作成

余裕があれば課題3-3) シーケンス図を作成

# 1) 要求分析とは

- ソフトウェア開発の初期段階で行われるもので、何をこのソフトウェアで実現するべきかを明らかにする作業。
- 要求分析者がユーザの要求を的確に把握し、ユーザに満足してもらえる実現可能な要求モデルを作成し、ソフトウェア設計者が正確に判断できる要求仕様書にまとめる。
- 要求仕様は、機能仕様と非機能仕様に分けることができる。

システムは～できる

システムの持つべき性能、信頼性、セキュリティなど

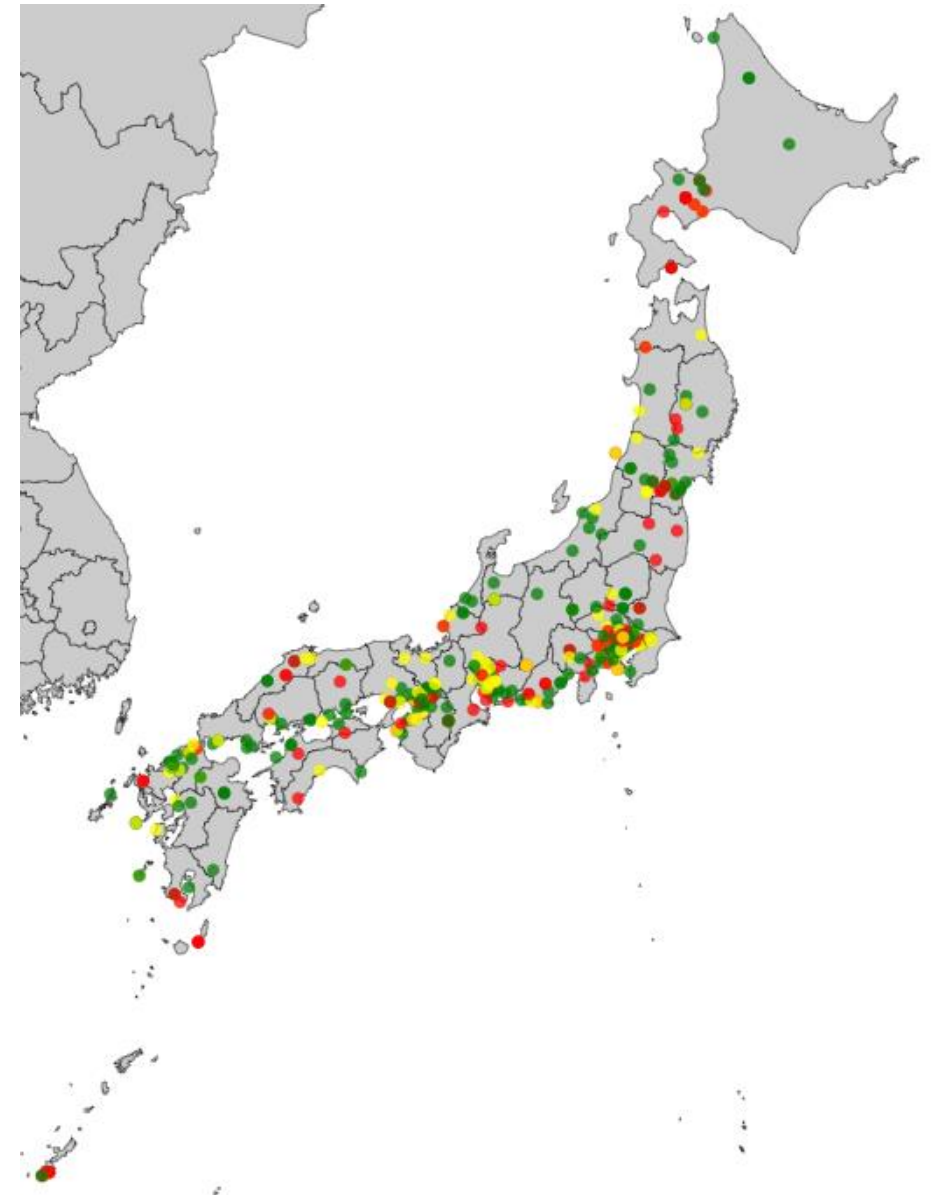
# 1)要求の表現

ユーザから獲得された情報・知識を整理・分析して、要求仕様書を作成することとなる。要求仕様書は、システム構成/外部インタフェース,システムの持つべき機能(機能的要求)、システムに要求される性能など(非機能要求)からなる。

**ユーザ要求を開発者がわかる表現(例えば、図)に変換する**

# 1) 要求の表現(例: ツイッター分析のデモ)

ツイートで各党を応援  
(#きのこ党, #たけのこ党, #どっちも党)  
ハッシュタグ解析して、  
現状調査(地理分布)

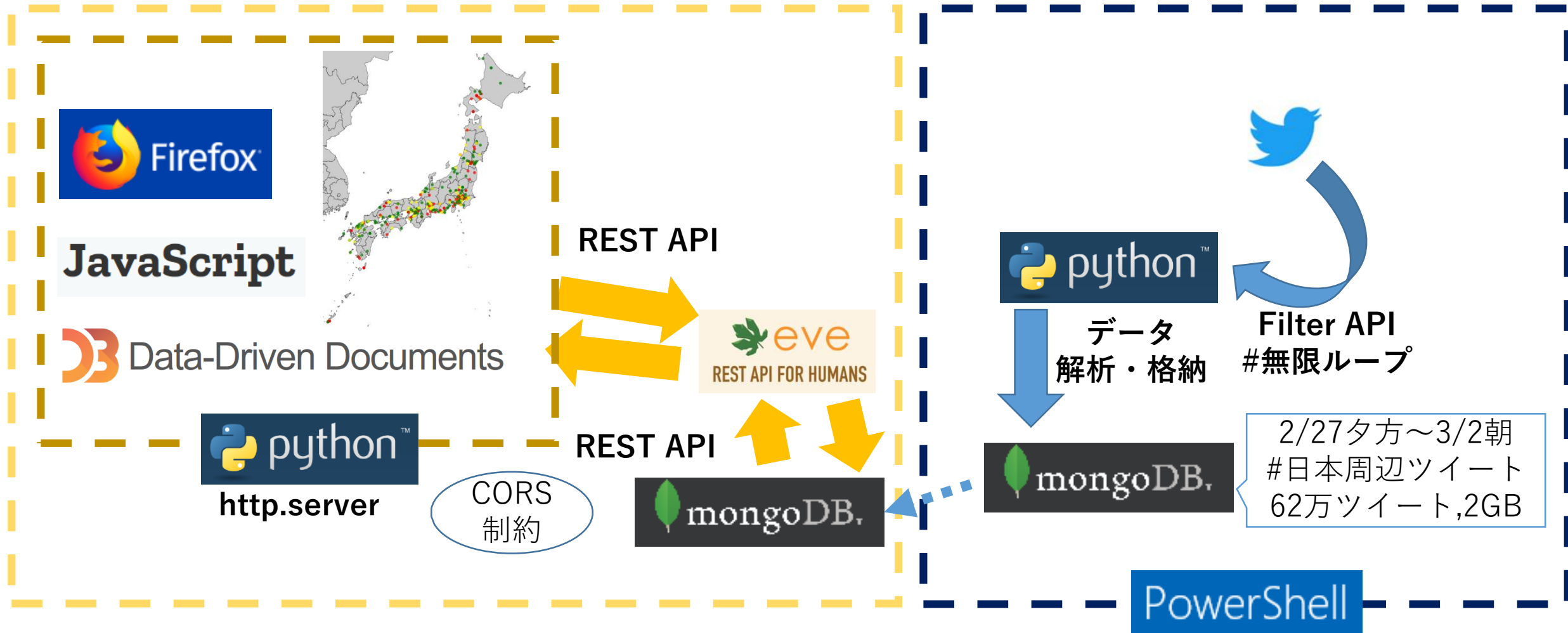


# 1) 要求の表現(例: ツイッター分析のデモ)

できるだけ実装がイメージできるほうが良い

ノートPC2)  
解析結果の表示

ノートPC1)  
ツイートの収集・解析

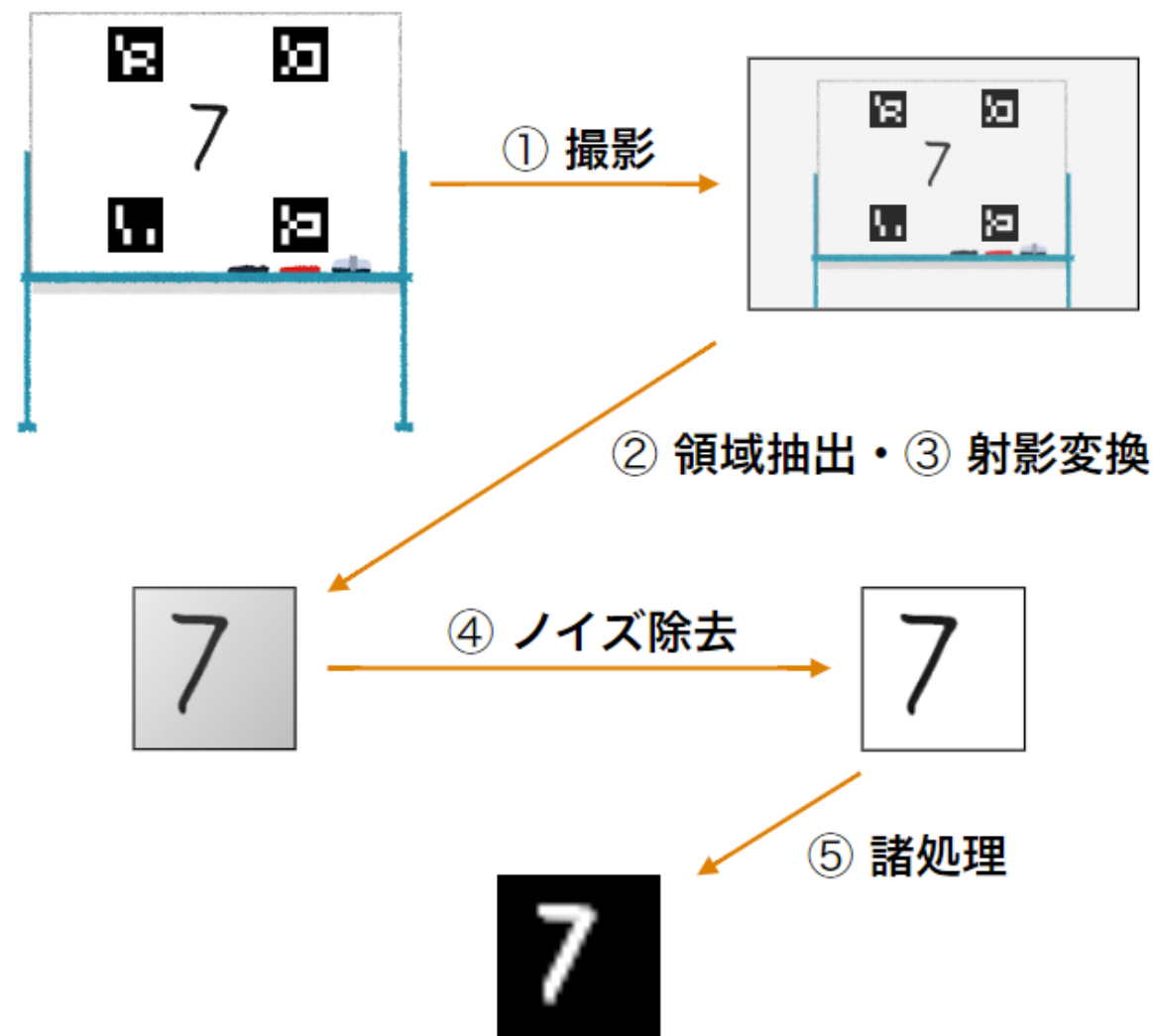


# 1) 要求の表現(例:手書き数字認識のデモ)

できるだけ実装がイメージできるほうが良い

## システムの流れ

- ① 撮影
- ② 領域抽出
- ③ 射影変換
- ④ ノイズ除去
- ⑤ 諸処理

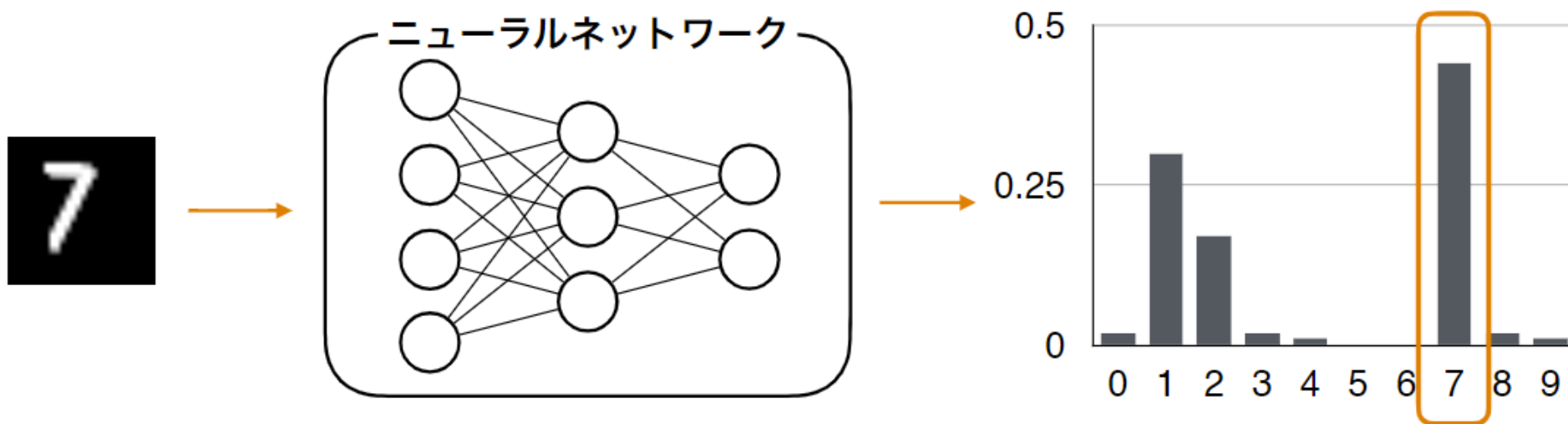




# 1) 要求の表現(例:手書き数字認識のデモ)

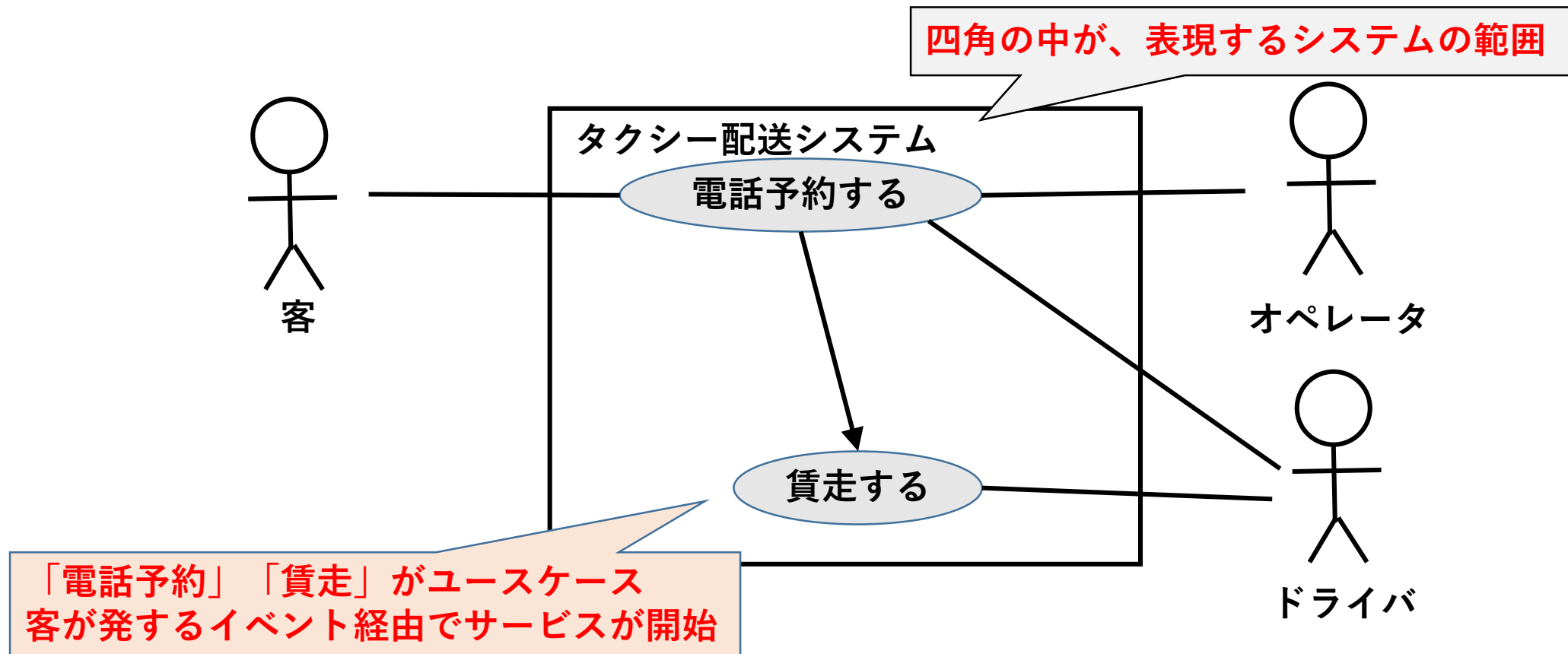
## システムの流れ

- ① 画像をニューラルネットワークへ入力
- ② 各数字に対する確率分布を出力
- ③ 最も確率の高い数字を認識数字とする



# 1) 要求の表現(例:UMLのユースケース図)

**ユースケース図**: システム利用者から見たシステムの利用の仕方を記述する記法である。「タクシー電話予約システム」の例が下記である。



# 1)会社での「要求の表現」

「相手の要求」を表現することもあるれば、自分の案を表現することもある

→会社では、決裁権者にOKをもらう必要が基本有り、説明資料を作る。

## • パッケージソフト(例:データベース)の新規機能を提案

製品開発部門の担当者に新規機能の内容を提案しOKもらって、開発し問題無く動くことを確認して、開発部署の担当者に実装説明する。#提案～技術説明まで、1年間

## • 特許執筆

新規機能が発売(一般公開)されると、特許化出来なくなるので公開前に特許出願できるように、特許化する。特許化内容をまとめて説明し、決裁をもらう。

## • 自社製品のユースケース作成

自社製品の拡販のためにユースケース案を作成し、決裁権者にOKをもらう。

#UMLで資料作成

# 1)会社での「要求の表現」

## Suica利用履歴販売、JR東は「個人情報に当たらない」との見解

2013年07月19日 22時22分 公開

[吉岡綾乃, Business Media 誠]



印刷



0



Share



61



PR 海外拠点の「ムダ」「ムラ」を解消 ヒントは固定資産管理

PR クラウド管理の新機軸「オブザーバビリティ」とは？

JR東日本が7月1日より、IC乗車券「Suica」の利用履歴を販売していることが、新聞やテレビに取り上げられ、ネットでも話題になっている。



JR東日本のICカード「Suica」

2013年当時は、使用しないことになったが、例えばsuicaデータを使ったシステムを顧客に提案（例:百貨店、銀行、通信キャリア、ショッピングセンター）

データを販売した相手は日立製作所。いわゆる“ビッグデータ”として、同社ではSuicaのデータを駅エリアのマーケティングに活用していく狙いだ。日立製作所から6月27日にプレスリリースが出ており（[参照リンク](#)）、ITmediaでも「日立、Suicaビッグデータから駅利用状況を分析するサービス」として記事を掲載している。

# 今日の内容

1) 要求分析のおさらい、ユーザ要求の表現

#何をこのソフトウェアで行うのか明らかにする

## 2) 要求の表現、UML

課題3-1,3-2) ユースケース図を作成

余裕があれば課題3-3) シーケンス図を作成

## 2) ユーザ要求の表現

ユーザから獲得された情報・知識を整理・分析して、要求仕様書を作成することとなる。要求仕様書は、システム構成/外部インタフェース、システムの持つべき機能(機能的要求)、システムに要求される性能など(非機能要求)からなる。

機能的要求は、仕様の理解のしやすさおよび検証のしやすさから、下記のような形式的な要求モデルを用いて記述されることが多い。

a) 機能階層モデル

b) ERモデル: Entity Relationモデル(実体関連モデル)

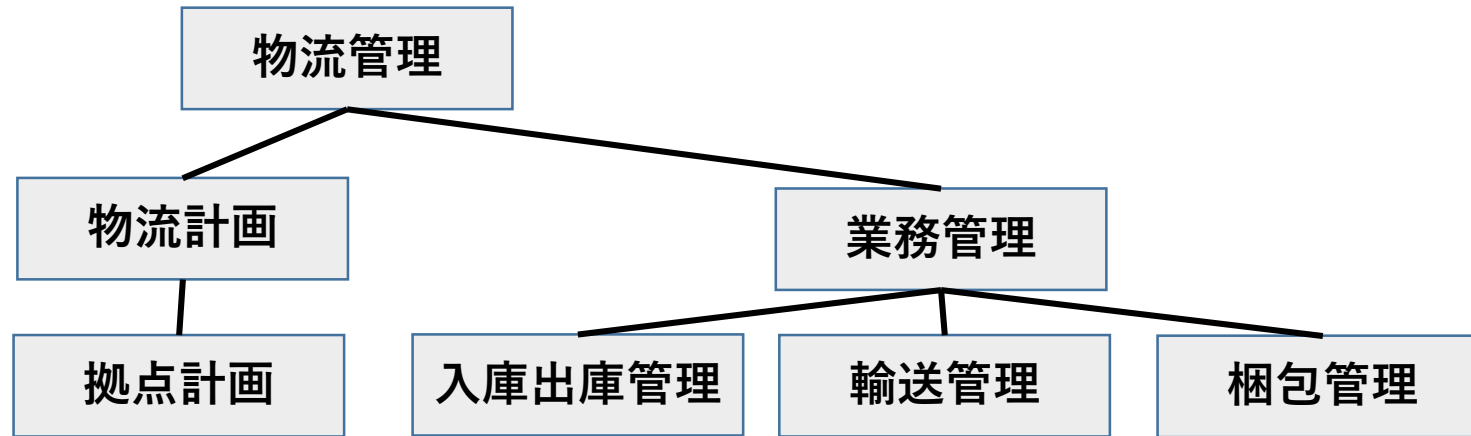
c) UMLモデル(ユースケース図、シーケンス図)

## 2) ユーザ要求の妥当性確認

- 初期の段階で分析の対象となった要求が真のユーザ要求であることを確認し、得られた要求モデルがユーザの問題をコンピュータにより解決するための表現であることを検証することが重要である。
- 要求モデル検証の基準
  - 1) 正当性 : 本当に全部必要な要件なのか？
  - 2) 完全性 : ユーザ要件に合致するのに必要な情報が全てあるか？
  - 3) 一貫性 : 矛盾した結論が導かれないか？
  - 4) 非あいまい性 : 要求を2つ以上の意味に解釈できないか？
  - 5) 最小性 : 要求仕様書に最小のことのみ記載されているか？
  - 6) 検証可能性 : 全ての要求が検証可能であるか？
  - 7) 実現可能性 : 要求仕様が技術面、コスト面からみて実現できるか？

## 2-a)機能階層モデル

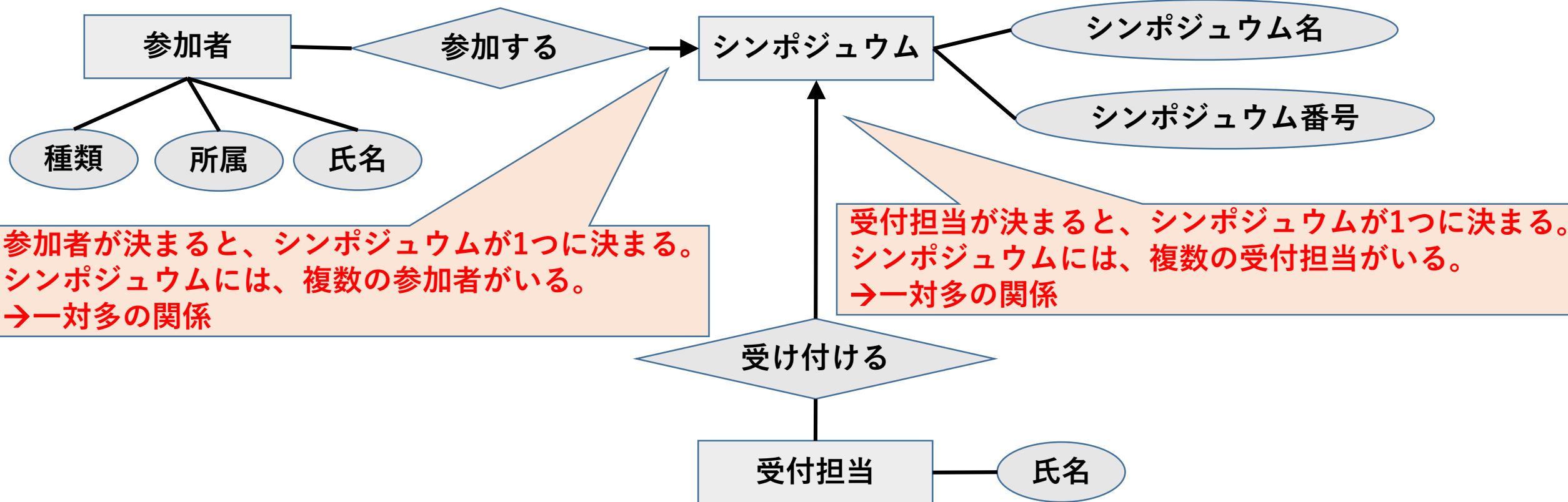
機能を階層的に詳細化し、ある単位機能まで展開された階層集合によって表現したものである。機能全体を組織的、体系的に表現するのに適している。下記は、物流管理業務を機能階層に展開した図である。





## 2-b)ERモデル(実体-関係モデル)

要求という概念を形を持った個体の集合である実体と実体間に存在する関係によって表現したものである。実体は属性を持っている。以下は、シンポジウムの受付業務のER図の例である。複数の参加者(実体)が1つのシンポジウムだけに参加(関係)している。



## 2-c)UML(ユースケース図とシーケンス図)

システムの統一記法であるUMLを用いてシステム概要を記述することもできる。

- ・ユースケース図

システム利用者の視点から、システムの提供するサービスを明確にする。個々のサービスの機能的な流れを記述する。システムの仕様イメージを表現する。

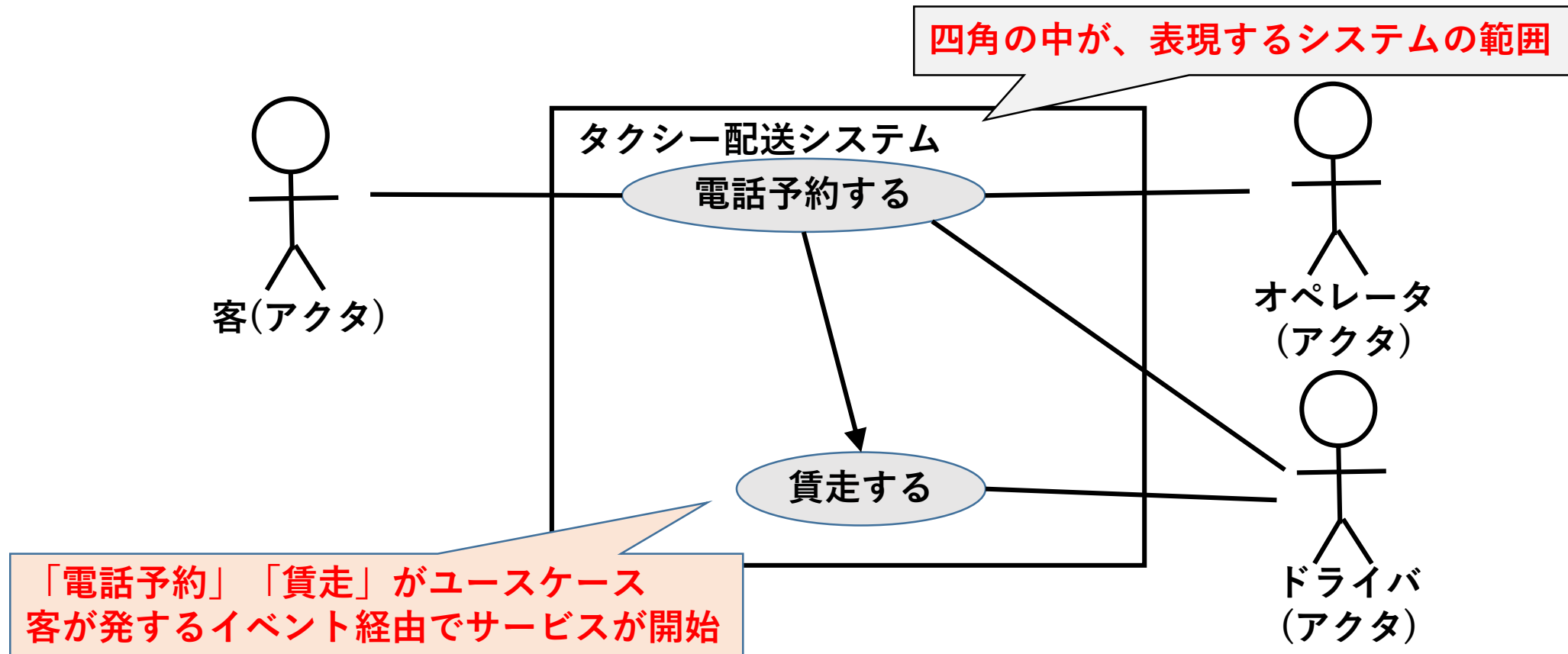
- ・シーケンス図

ユースケースがメッセージのやりとりによってどのように達成されるかを示す。

## 2-c)UML(ユースケース図)

**ユースケース図**:システム利用者から見たシステムの利用の仕方を記述する記法である。「タクシー電話予約システム」の例が下記である。

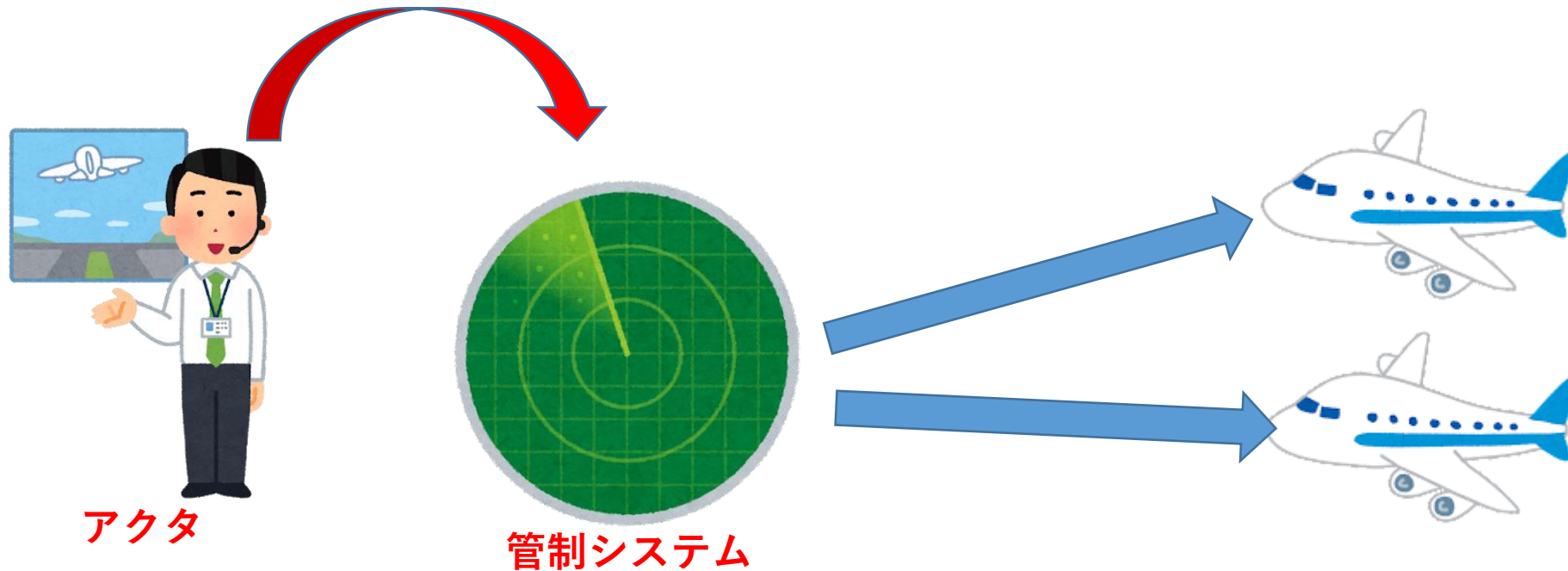
**ユースケース**:システムが**アクタ(外部のユーザや外部システム)**に提供するサービスのこと。



## 2-c) アクタ

**アクタ:**システムが動作をはじめる最初のイベントを起こす役割を持ったオブジェクトである。アクタは、システムの制御対象ではないので属性を持たない。

アクタは、システムに外からのイベントを与えるものであり、システム内部の情報モデルではない。例えば、レーダ管制システムの場合には、レーダ操作者がアクタとなる。



## 2-c) ユースケースの表記法

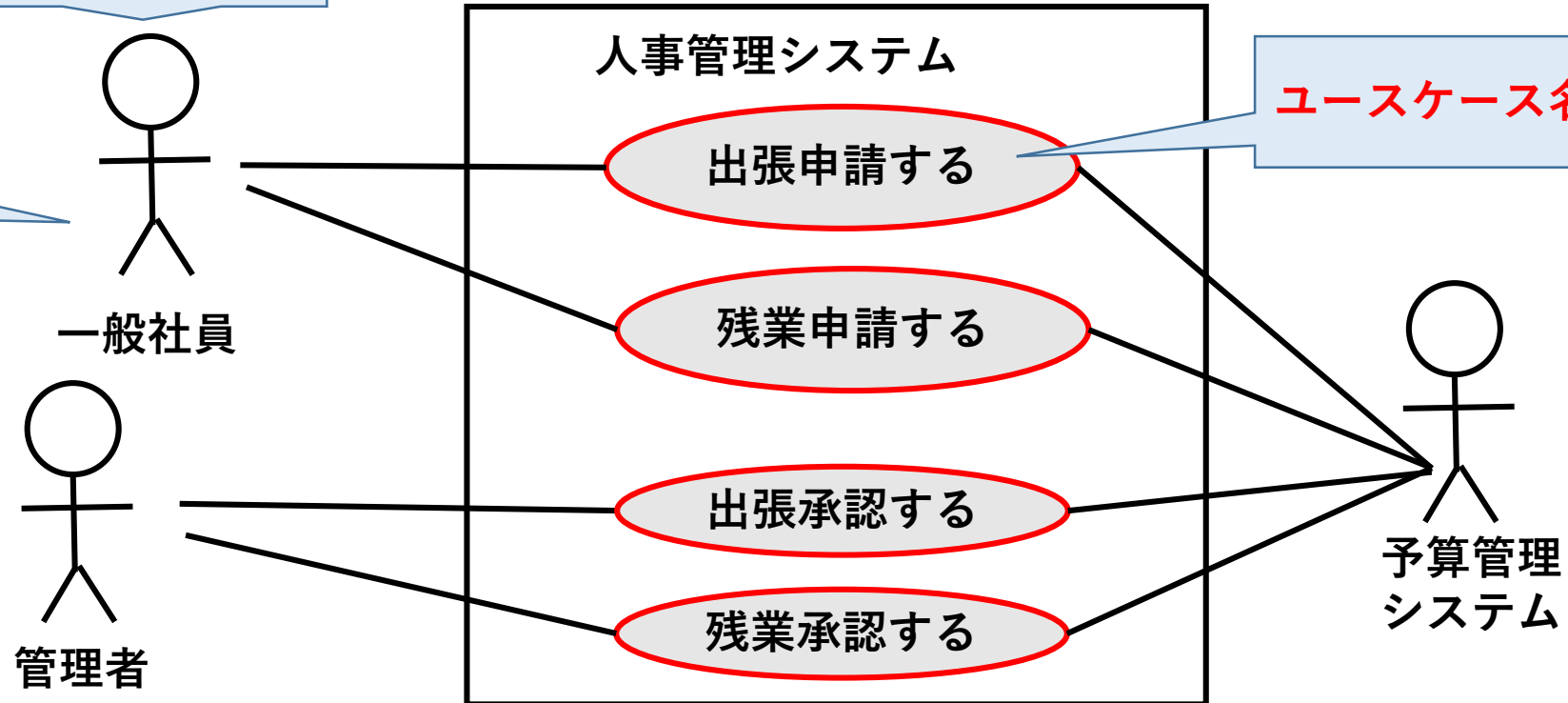
楕円の中にユースケース名を記載する。ユースケースは、検討対象のシステムがアクターに対して提供する機能/振る舞いのこと。

**アクター:** 検討対象のシステムを利用する人間、外部システム

アクターは、役割毎(一般社員、システム管理者)に記載する。

人間は、ステックマンと呼ばれるアイコンで表現する

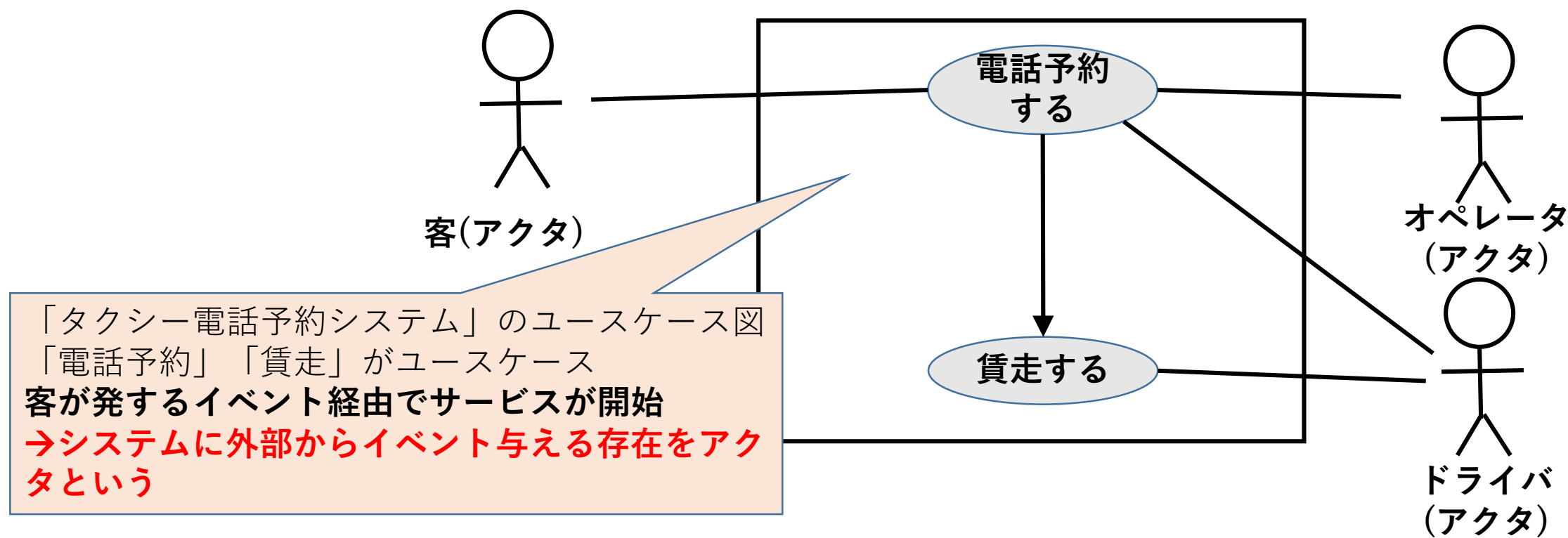
アクター



## 2-c) ユースケース図

**ユースケース:** システムが(外部ユーザや外部システム)に提供するサービス。

**ユースケース図:** ユーザから見たシステムの利用の仕方を記述する記法



## 2-c)ユースケース記述の例

**名前:**電話予約

**概要:**客が配車センターに電話し、タクシーを予約する

**アクタ:**客、オペレータ、ドライバ

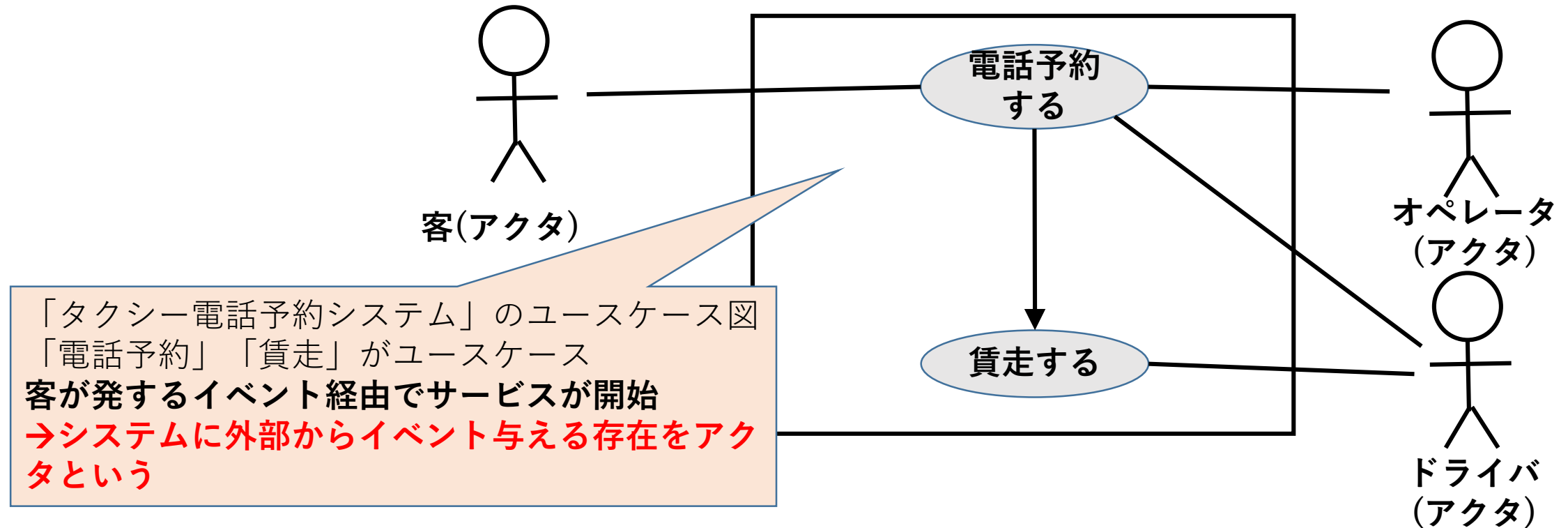
**前提条件:**

**記述:**

- ・ **電話予約** 客が配車センタに電話をし、タクシーを予約する。このとき、オペレータは**1)**客から情報を聞き、**2)**端末に打ち込む。情報が打ち込まれると、タクシーの中から優先順位の高いタクシーから順に、予約を受け付けるか問合せる。了解の場合には、予約が確定する。拒絶の場合には、次に優先度が高いタクシーを選択し、問合せを進める。これは了解となるまで続けられる。
- ・ **貸走** 予約が確定すると、タクシーは客へ向かい貸走する。

## 2-c)ユースケース記述の例2

- ・ **電話予約** 客が配車センタに電話をし、タクシーを予約する。このとき、オペレータは**1)**客から情報を聞き、**2)**端末に打ち込む。情報が打ち込まれると、タクシーの中から優先順位の高いタクシーから順に、予約を受け付けるか問合せる。了解の場合には、予約が確定する。拒絶の場合には、次に優先度が高いタクシーを選択し、問合せを進める。これは了解となるまで続けられる。
- ・ **貸走** 予約が確定すると、タクシーは客へ向かい貸走する。



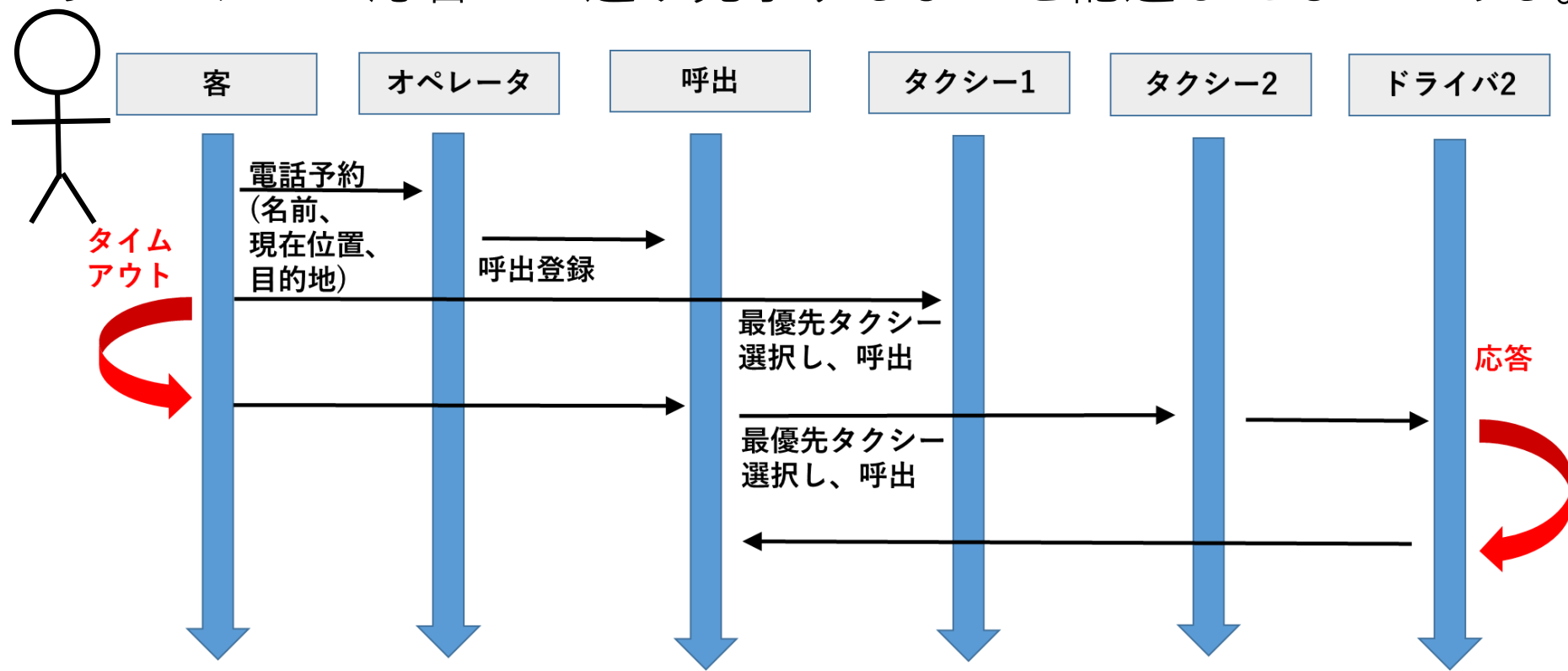


## 2-c) シーケンス図

シーケンス図では、1つのユースケースに関する1つのシナリオ(システムが実行される1例)をメッセージのやり取りによって表現する。

シーケンス図では、オブジェクト間のやり取りを時間軸に沿って記述することで、システムの動的な側面を表現する。

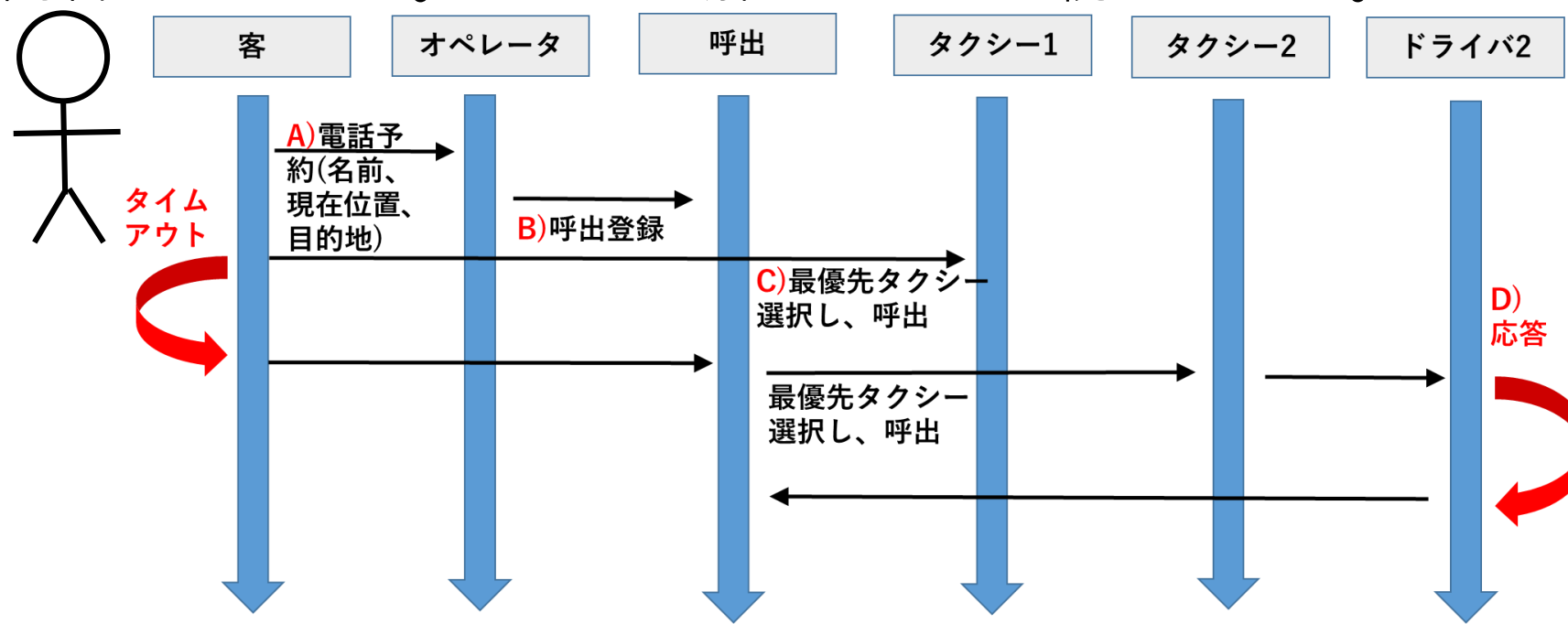
シーケンス図は、ユースケース記述と同様に、アクタからのメッセージを契機として、開始してからシステムの応答が一通り完了するまでを記述したものである。



## 2-c)シーケンス図の例

記述:**A)**客が配車センタに電話をし、タクシーを予約する。このとき、**B)**オペレータは1)客から名前と現在位置および目的地を聞き、2)端末に打ち込む。

**C)**情報が打ち込まれると、タクシーの中から優先順位決定アルゴリズムによって決定した優先順位の高いタクシーから順に、予約を受け付けるか問合せる。問合せを行うと、簡易端末に客の名前と目的地が表示される。**D)**タクシーのドライバは、予約を受けける場合は了解ボタンを、そうでない場合は拒絶ボタンを押す。10秒以上応答がない場合には、拒絶とみなされる。了解の場合には、予約が確定する。拒絶の場合には、次に優先度が高いタクシーを選択し、問合せを進める。これは了解となるまで続けられる。



## 2-c) ユースケースの詳細(関連と多重度)

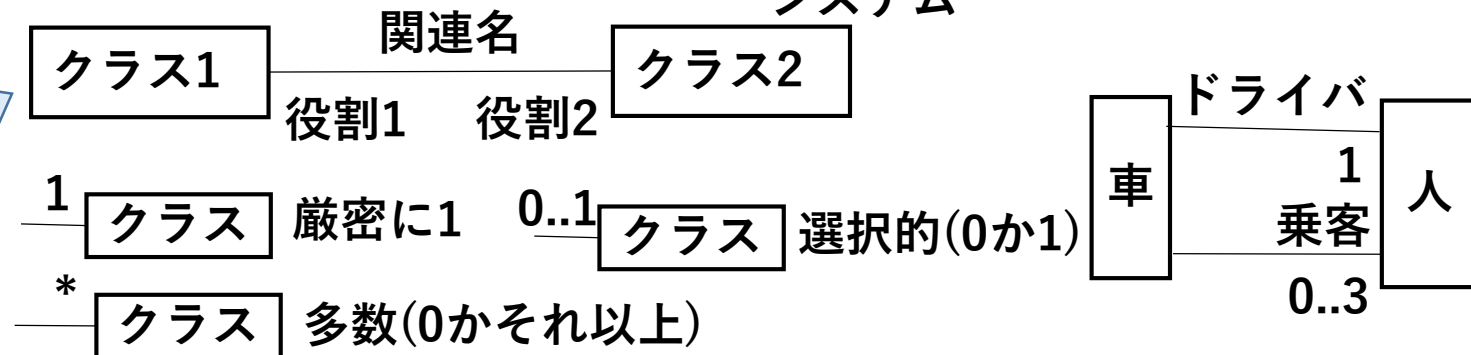
ユースケース図では、どのユースケースがどのアクターから利用されるかを、「ユースケースとアクターのつながり」である関連で表現する。

**多重度:** 関連を持つ2つの間(例:アクター、ユースケース)で、片方の1つに対するもう一方の数に対する制約である。

例:車からみて、「ドライバ(人)は厳密に1人」「乗客(人)は3人まで可能」



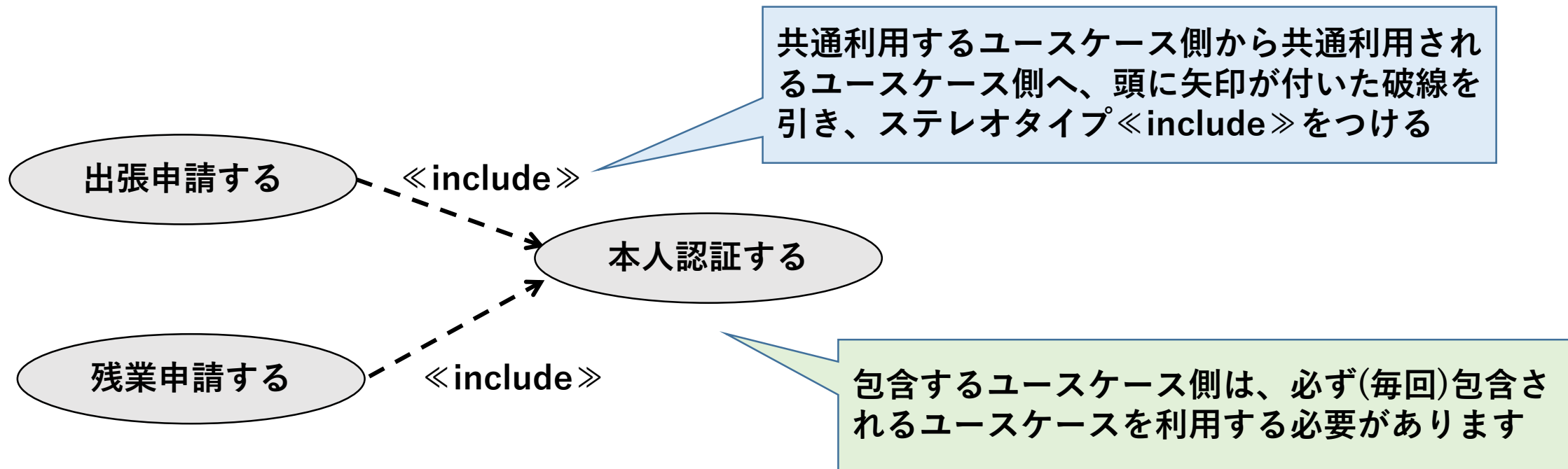
多重度は、(オブジェクト指向の)クラス間の関係でも使われる。  
関連は、そのクラス間を結ぶ線で表す。



## 2-c) ユースケースの詳細(ステレオタイプと包含)

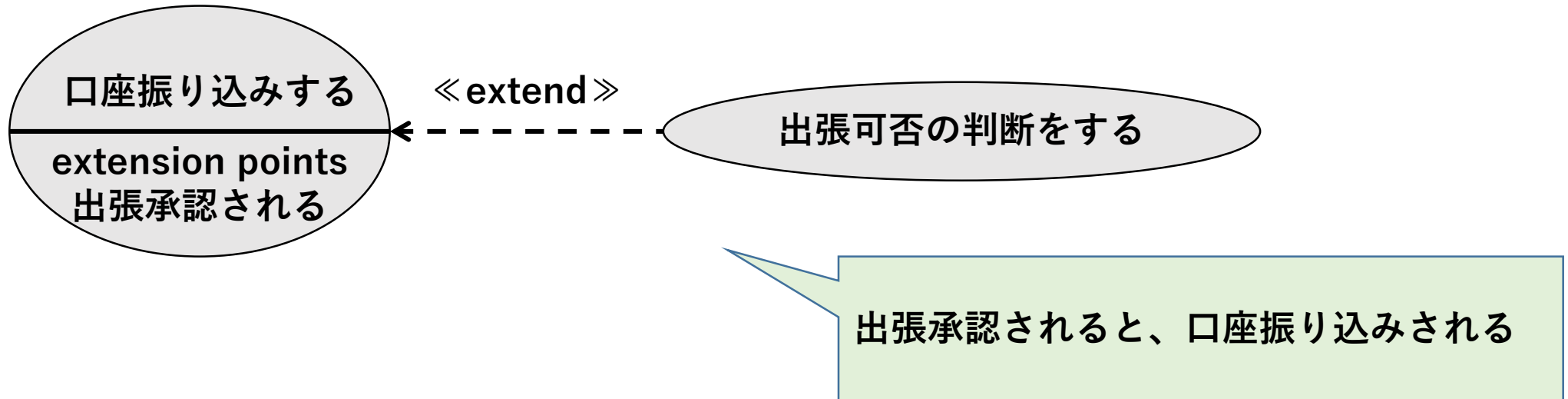
**ステレオタイプ:**ある要素のグループを意味付けにより分類できる。ステレオタイプの名称はギューメ”《》“でくくって表現する。ステレオタイプの名称は、基本的には自由に利用者が命名できる。

**ユースケースの包含:**複数のユースケースから共通利用されるユースケースとの間に存在する関係を包含という。あるユースケースが他のユースケースを呼び出すことを表現する場合に用いる。



## 2-c) ユースケースの詳細(拡張)

拡張: 条件付きで他のユースケースを利用することを指す関係を拡張という。包含と似ているが、他のユースケースの利用が必須ではなく条件付きとなる。



## 2-c) ユースケース作成の流れと注意点

- ① システムで実現したいことをまとめる
- ② まとめたアイデアから、ユースケースとアクターの候補を見つける  
# 「誰が(アクター)」 「何をする(ユースケース)」 に該当するものを列挙
- ③ システムで実現する範囲を明確にし、ユースケースとアクターの候補を再検討する  
# 例: 勤怠管理システムが別にあるなら、残業申請はそちらで扱うので削除
- ④ 再検討後のユースケースとアクターでユースケース図を作成する

# 今日の内容

1) 要求分析のおさらい、ユーザ要求の表現

#何をこのソフトウェアで行うのか明らかにする

2) 要求の表現、UML

**課題3-1, 3-2) ユースケース図を作成**

余裕があれば課題3-3) シーケンス図を作成

# 課題3-1ユースケース図の作成(締切:5/11)

航空会社の顧客管理システムのユースケース図をdraw.ioで作成ください。

#提出形式は、画像データ(スクリーンショット)をwordファイルに貼り付けてPDF化ください。PDF化出来ない場合は、wordファイルのままでも可

ユースケースの内容は各自で考案ください。なお、航空便の予約・キャンセル(フライトナンバーと日付も指定可能)および搭乗手続きは含めてください。

まずは、個人で作成し、その後に隣の人と作成したユースケース図を見せあってください。





# Untitled Diagram

File Edit View Arrange Extras Help

保存されていません。ここをクリックして保存してください

Share



100%



Advanced

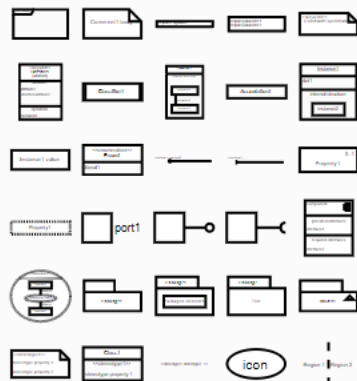
Basic

Arrows

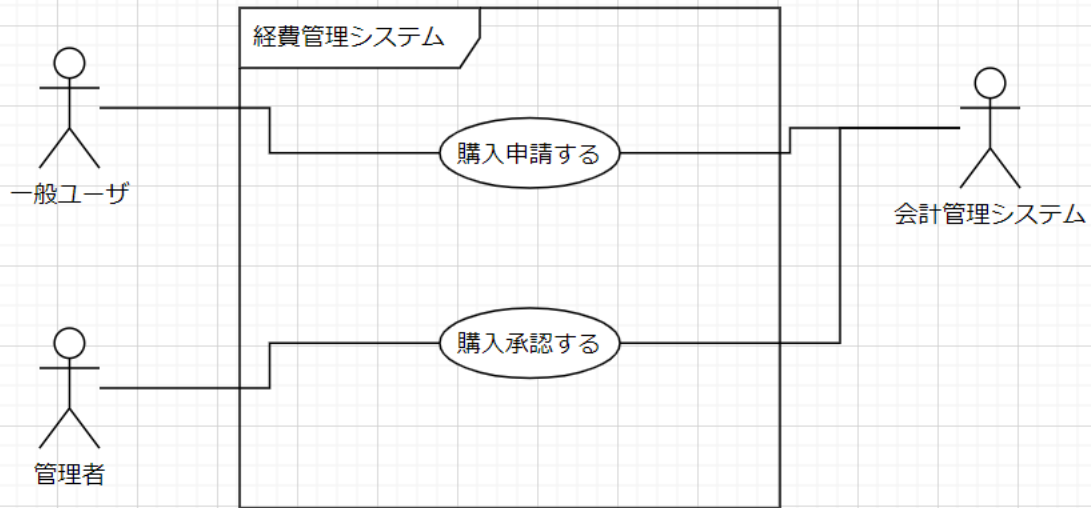
Flowchart

Entity Relation

UML 2.5



+ More Shapes...



Page-1





draw.io



File Edit View Arrange Extras Help



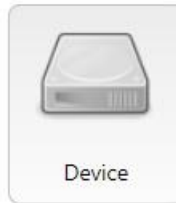
Save diagrams to:



Google Drive



OneDrive



Device



Dropbox



GitHub



GitLab

Decide later



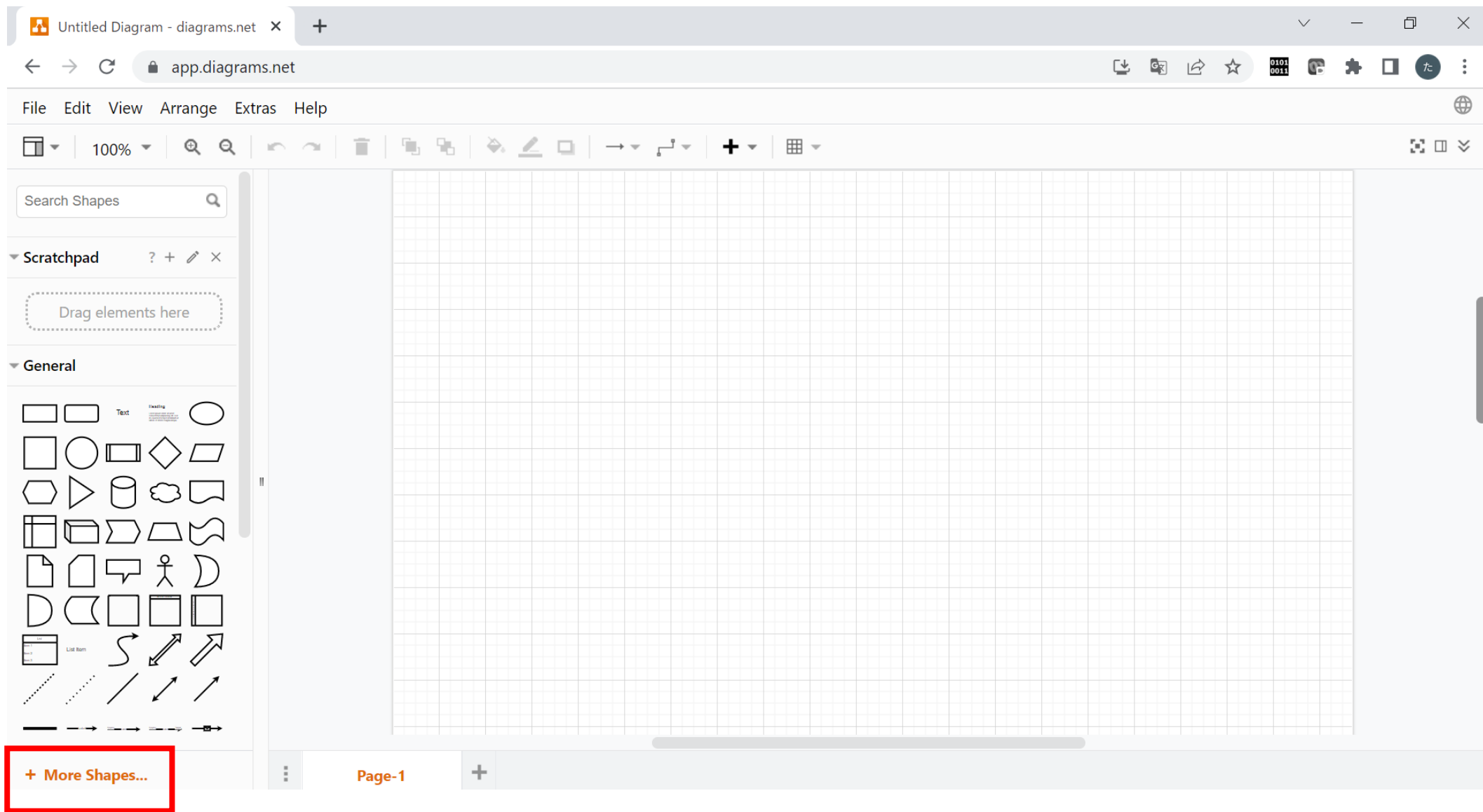
Device

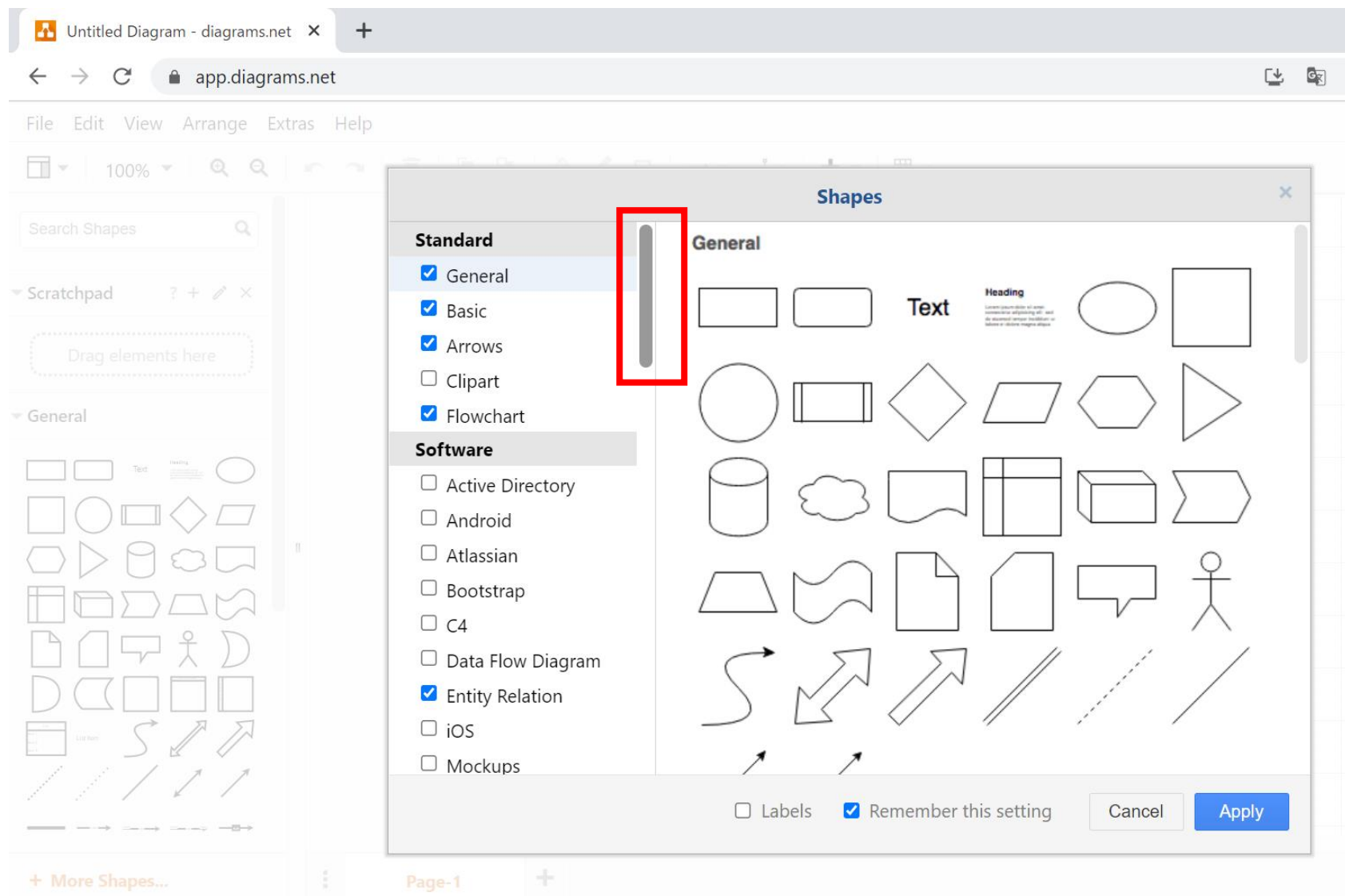
Create New Diagram

Open Existing Diagram

Change storage





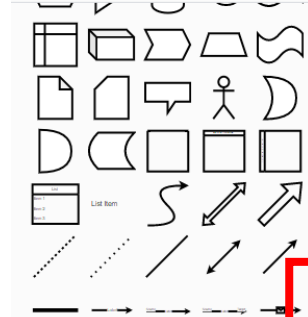


×

- ## Networking

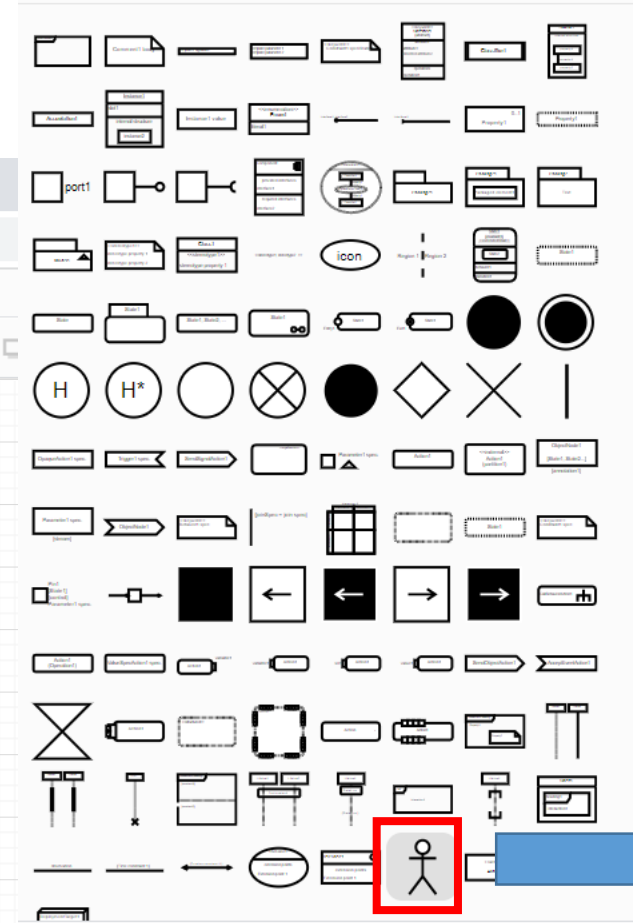
[illegible]☒ Remember this setting

## Apply

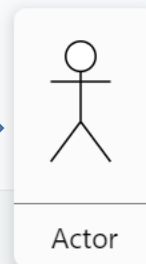


- ▶ Misc
- ▶ Advanced
- ▶ Basic
- ▶ Arrows
- ▶ Flowchart
- ▶ Entity Relation
- ▶ UML 2.5
- ▶ UML

## UML 2.5



+ More Shapes...



Page-1

Untitled Diagram - diagrams.net x +

← → ↻ app.diagrams.net

File Edit View Arrange Extras Help Unsaved changes. Click here to save. ⬇

Style ▾ Helvetica ▾ 12 ▾ **B** *I* U [List] ▾ [Align] ▾ [Code] </> + ▾ [Grid]

▼ UML 2.5

Actor

# 課題3-2レポート案の「要求の表現」(締切:5/11)

課題2-2のレポートの内容についてユースケースを作成ください。

課題2-2は以下の内容でした。

- MECAの1年生が、データサイエンスに興味をもってもらえるような教材の案を作成してください。扱う範囲は、深層学習、機械学習、IoT、HTMLを想定しています。各技術を使った、各学科に関係ありそうなデモ案でもよいです。

技術的には作れるものにしてください。

#前期:データサイエンス入門

160人を一度に講義するため、受講生はパソコン使用不可

#後期:データサイエンス演習

80人毎に講義するため、受講生は演習室PCが使用可能



# 余裕があれば課題3-3(締切5/11)

課題)救急車呼出しのシーケンス図を作成せよ

記述:急患からの救急呼出しを救急管理センターが受け付けて、オペレーターは端末に情報を入力し、優先度の高い救急車に順次問合せをする。救急車は患者を受け付けることになったら、受け入れ先の病院を探す。救急車は優先順位の高い病院に順次問合せを行い、受け入れ可能な病院が見つかるまで、問合せを繰り返すとする。なお、問合せが拒絶されたとみなす秒数や患者受け入れなどの詳細は自由に決めてよい。

#タクシーと基本的に同じです