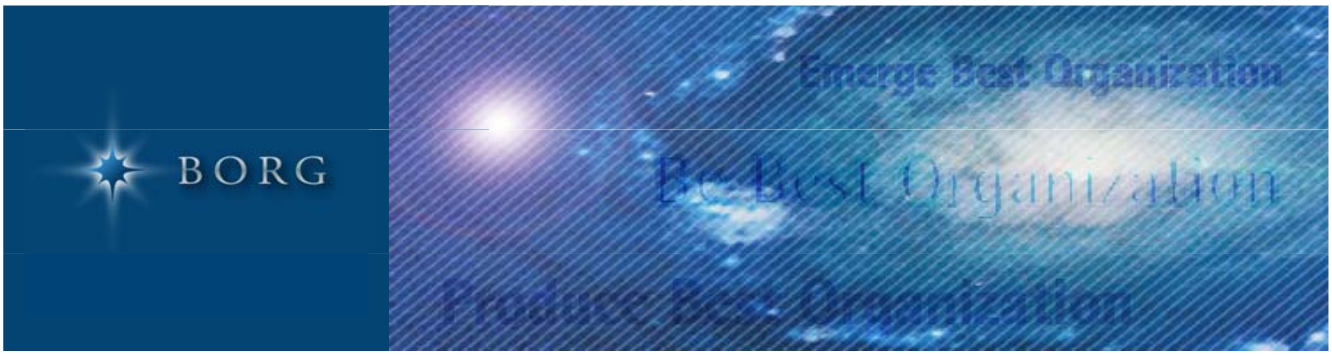


業務アプリケーション開発におけるユースケース・モデリング
～携帯店舗管理システム構築でのUML活用事例～

2009年10月29日
ボーグ株式会社

BORG

目　次

1. BORG社の紹介

社名　　：BORG株式会社　Borg, Inc.
設立年月　：2006年7月
所在地　　：　〒160-0022　新宿区新宿1-17-2 ハギワラビル3階
　　第2オフィス　新宿区新宿1-11-12　岩下ビル7F
URL　　：http://www.borg.co.jp/
代表　　：代表取締役社長　久津　豪
（略歴）早稲田大学商学研究科修了、北陸先端科学技術大学院大学知識科学研究科修了、外資系コンサルティング会社、ITベンチャー会社を経てBORG社設立。専門は経営戦略、管理会計、意思決定科学、組織科学、情報システム論など
事業内容　：Webエージェントサービスの企画・設計・開発・運営
　　組織情報システムの企画・設計・構築・保守・運用
　　IT経営コンサルティングサービス　　等
社名の由来：Best Organization　の略。
　　Be Best Organization,　Produce Best Organization,
　　Emerge Best Organizations

2. UML開発の取り組み

- 2006年にBORG設立後、エージェントシステム構築サービスを企画
  - 設立当初より、ソフトウェア開発のモデリングとしてUMLを採用
  - 開発環境（CASEツール）として、以下の環境を整備

    VISIO（マイクロソフト社）、Enterprise Architect（スパークシステム社）、SI Object Browser（システムインテグレータ社）、Visual Studio Team Suite、Eclipseを活用。

【これまでに開発したソフトウェア】

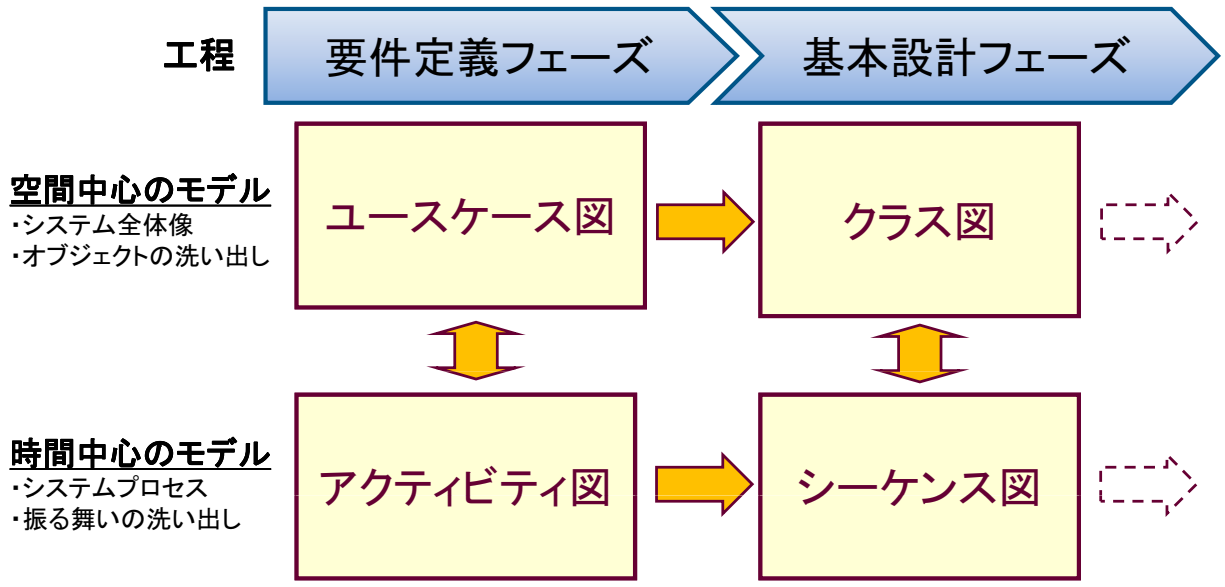
- BAS（Borg Agent Software）（2006年～）C#,.NET Framework 3.0
  - エンタープライズ向け統合ミドルウェア（EAI・EBS・BPM・SyncManagement・CMS・コンテンツ変換）
- PM-ADVANCE（2006）　C#,.NET Framework 2.0
  - プロジェクト収支管理ソフトウェア（個別管理計算による見積・受注・売上・請求管理、見積照会・仕入・支払管理、経費申請・精算、見込・目標・予算・実績管理）
- SHOP ACE Mobile（2008）C#,.NET Framework 3.0,WindowsMobile,Java
  - 携帯電話による勤怠管理・受発注管理・業績管理アプリケーション
- NET SHOP ACE（2008）Java
  - リッチクライアント対応ECパッケージ

3.ユースケースモデリングの目的・意義・活用方針

- ユースケースモデリングの目的
  - 構築するシステムがどのような場面（ユースケース）において、どのようなユーザ（アクター）に対して、どのようなサービス機能（サブジェクト）を提供すればよいかユースケース図を作成することで明らかにする
- ユースケースモデリングの意義
  - システムの範囲・対象を俯瞰的に理解できる
  - ビジュアル化することで開発メンバーやユーザ企業との課題の共有ができる
  - 後工程でおこなうクラス図などの設計において、要求や制約を明らかにできる
- ボーグにおけるユースケース図の活用方針
  - ユースケースモデリングは業務アプリケーションにおいて、システム機能の概要を決めるための最初の作業
  - ユースケース図は、システムの振る舞いに対する空間の意味情報として整理するためのツール
  - ユースケース図を作る際には、ユースケース、アクター、サブジェクトに対して、CONDITION（条件・制約）、ROLE（役割・要求）、RULE（規則）を検討する
  - ユースケース図を「静的なシステム機能表現図」として活用する

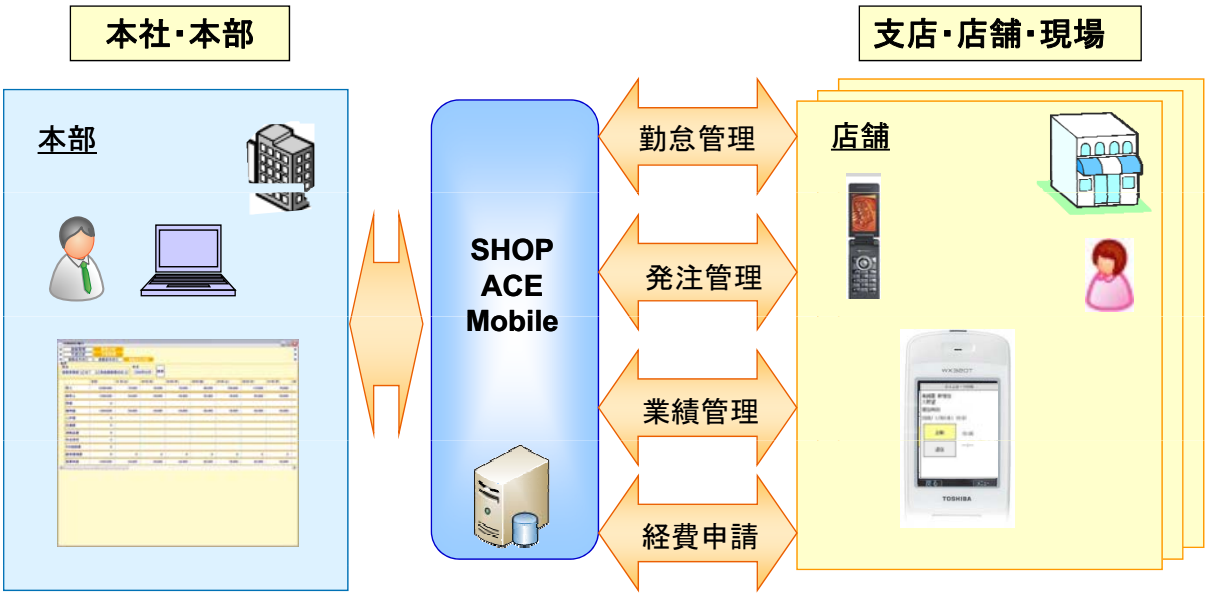
4. 開発時のUMLによるモデリングアプローチ

- BORGにおける工程別に制作するUMLのダイアグラムの関係は以下の通り。



5. 開発事例：SHOP ACEの紹介

- UMLを活用した開発として、弊社ソフトウェアパッケージのSHOP ACE Mobileの導入事例をもとに説明する。
- SHOP ACE Mobile：店舗に配置したモバイル（携帯電話、PHS、スマートフォン、ノートPC）から、勤怠管理、受発注管理、業績管理がリアルタイムでできるソフトウェア。



6. 開発事例：SHOP ACE機能一覧

- SHOP ACE では、以下の業務管理機能に対して、ユースケースの洗い出しをおこなった。

SHOP ACE WORKS(勤怠管理)

- タイムカード
- 出勤簿確認
- シフト確認
- 従業員情報確認
- 経費申請

勤怠管理

SHOP ACE ORDER(発注管理)

- 商品当日発注
- 商品予約発注
- 原材料発注
- 納期回答確認
- 商品棚卸報告

発注管理

SHOP ACE BI (業績管理)

- 売上報告
- 客数報告
- 店舗業績参照（日・週・月）
- 各種レポート申請

業績管理

SHOP ACE (基本機能)

- お知らせ機能
- データ取得（同期）
- 端末ユーザ登録
- アプリケーションロック

→ユースケースモデリングによるSHOP ACEの全体モデル検討から勤怠管理のユースケース図、アクティビティ図、クラス図、シーケンス図の作成までを説明

7. SHOP ACE開発でのモデリングステップ

1. サブジェクト(場)の定義
業務の場(ロケーション)をサブジェクトとして抽出

2. サブジェクトに関連するアクターの抽出
業務ユーザをアクターとして定義

3. サブジェクト内の機能定義
業務の場での業務遂行のための機能を洗い出し

4. アクターの役割(ROLL)定義
業務機能遂行上のアクターの役割を定義

5. ユースケース図作成
1～4を繰り返して、サブジェクトごとにユースケース図をまとめる

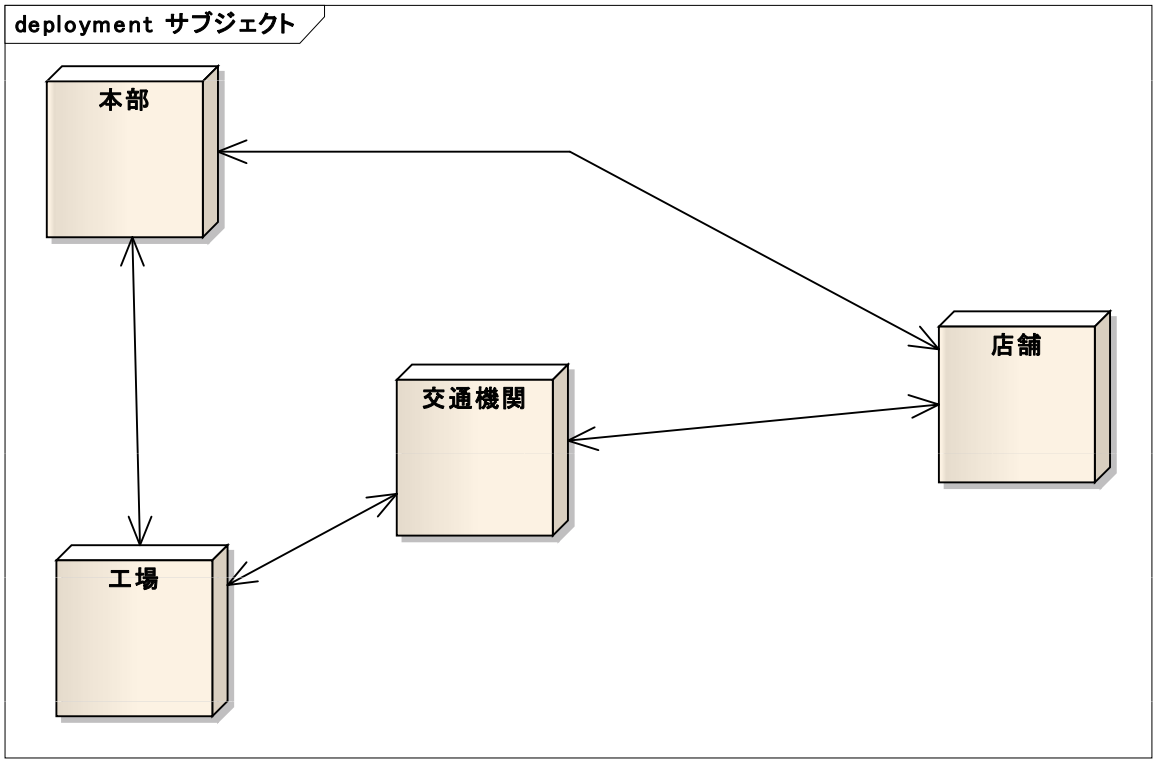
6. アクティビティ図作成
アクター＊サブジェクトでパーティションを作成して機能フローを定義

7. クラス図作成
それぞれのユースケースに対して、クラスを設計

8. シーケンス図作成
それぞれのアクティビティ図のフローごとに、シーケンス図を作成

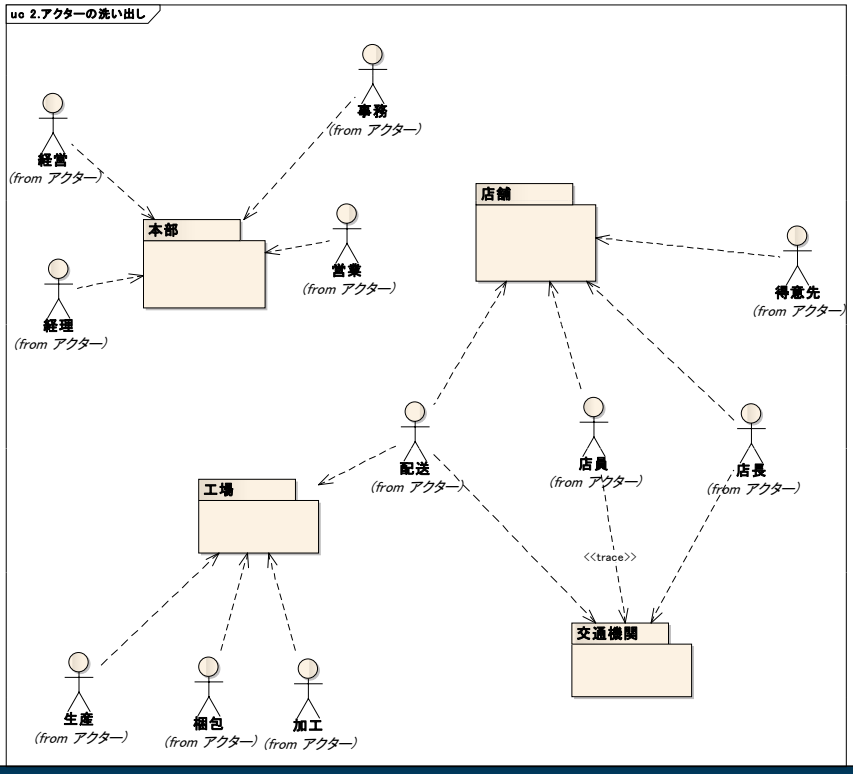
7-1. サブジェクト(場)の定義

業務の場(ロケーション)をサブジェクトとして抽出する



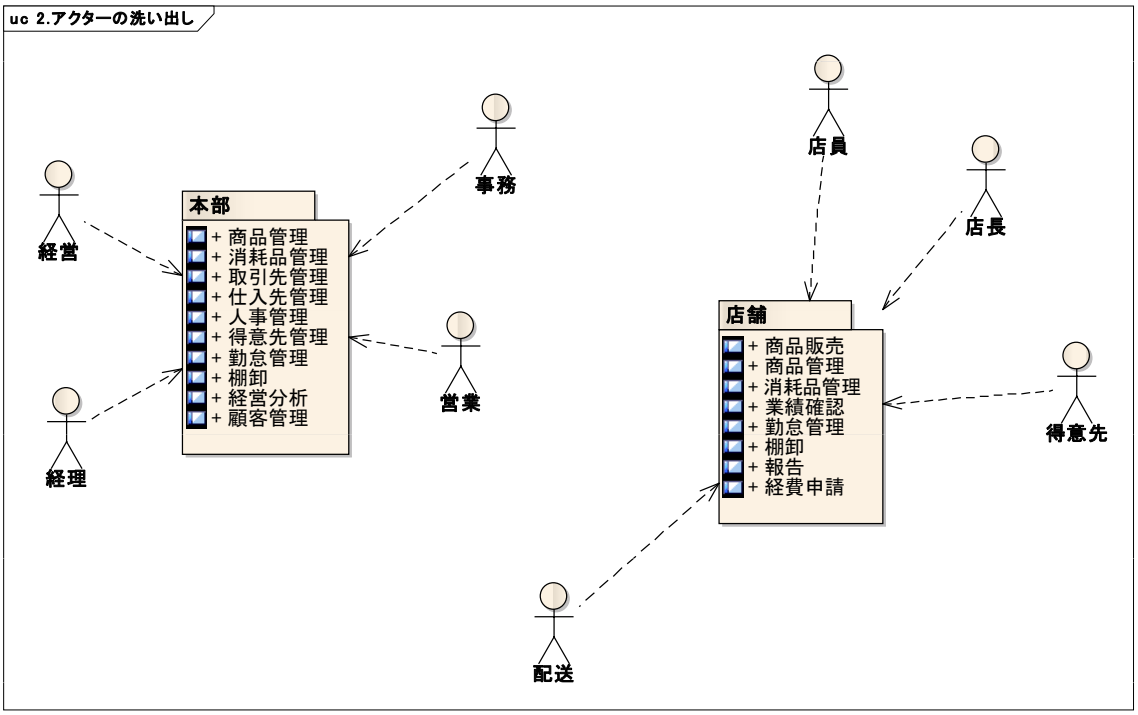
7-2. サブジェクトに関連するアクターの抽出

業務ユーザをアクターとして定義する



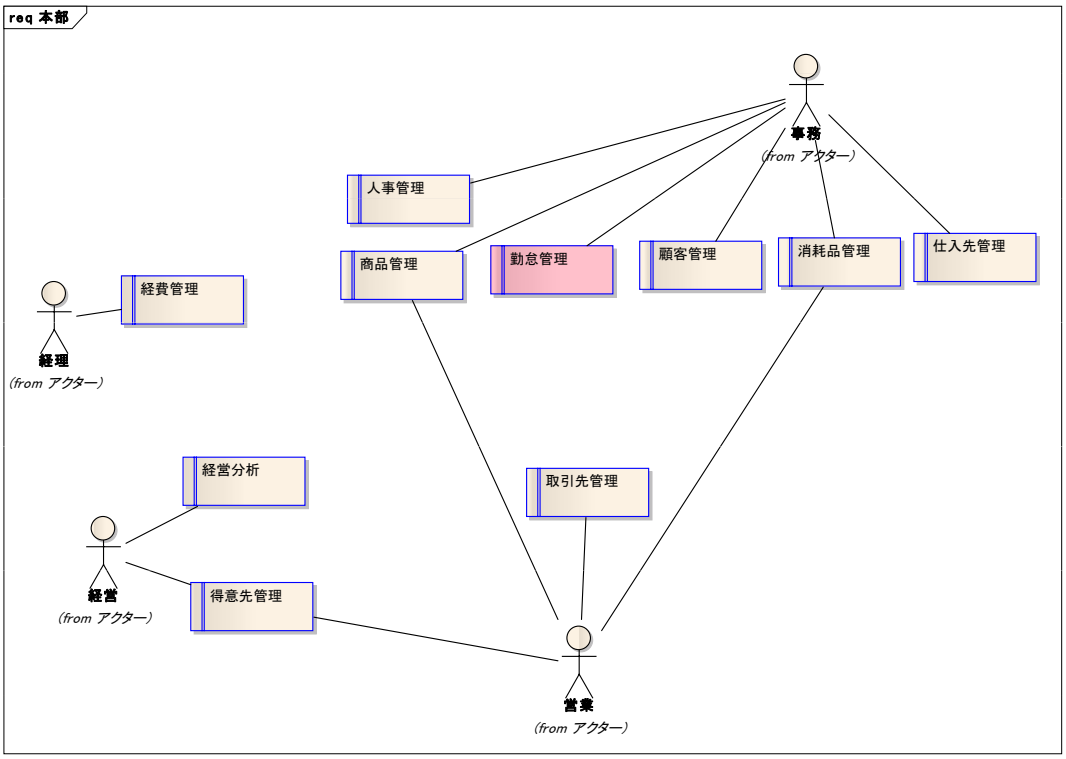
7-3. サブジェクト内の機能定義

業務の場での業務遂行のための機能を洗い出す



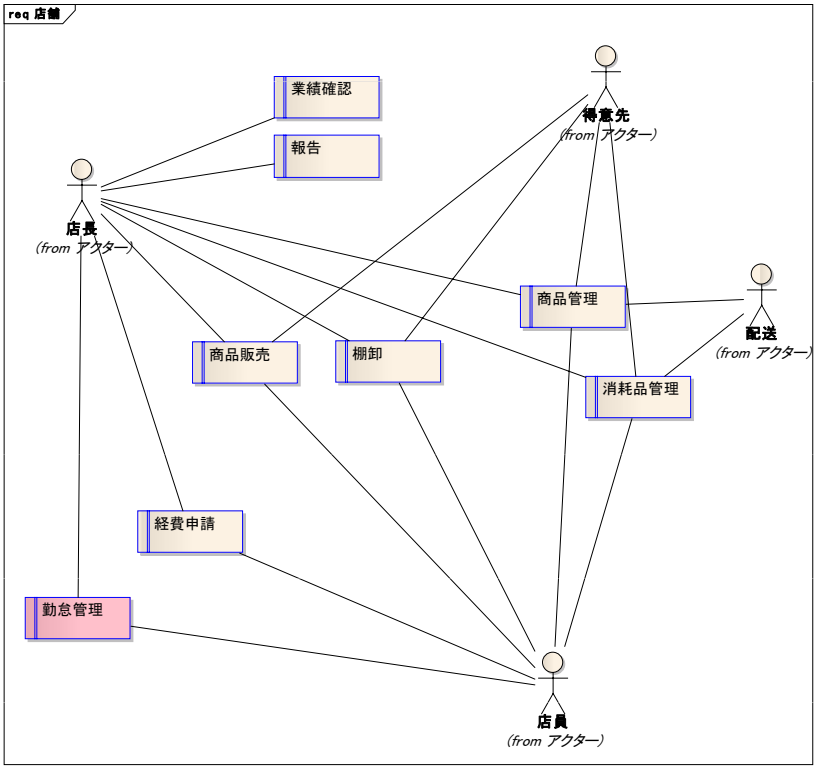
7-4. アクターの役割(ROLL)定義①～本部～

業務機能遂行上のアクターの役割を定義



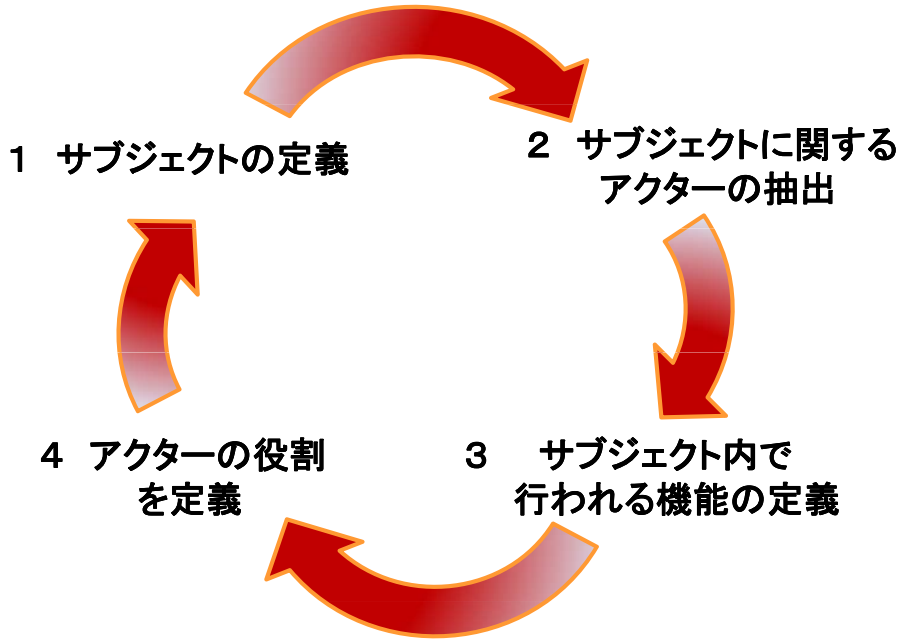
7-4. アクターの役割(ROLL)定義②～店舗～

業務機能遂行上のアクターの役割を定義



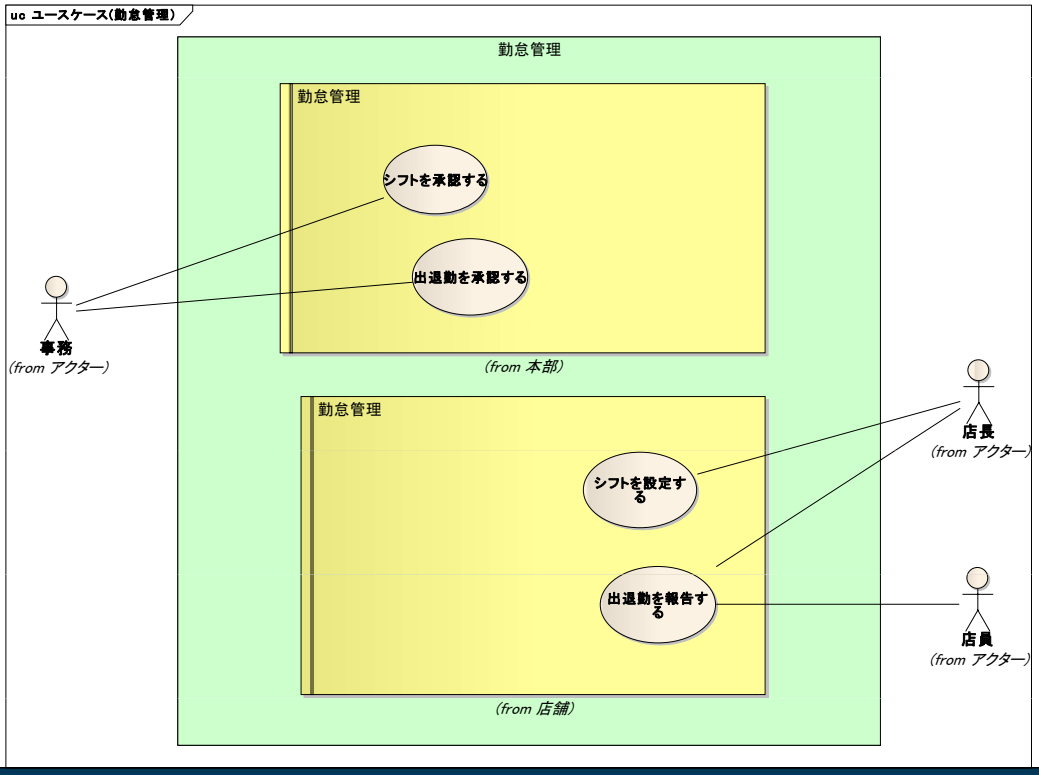
【参考】 モデリングステップの繰り返しによる精緻化

サブジェクトの定義からアクターの役割定義までのプロセスを繰り返してユースケースの内容について精度や確度を高め、アクター、サブジェクトだけでなく、その要求、制約、役割、ルールについても定義していく



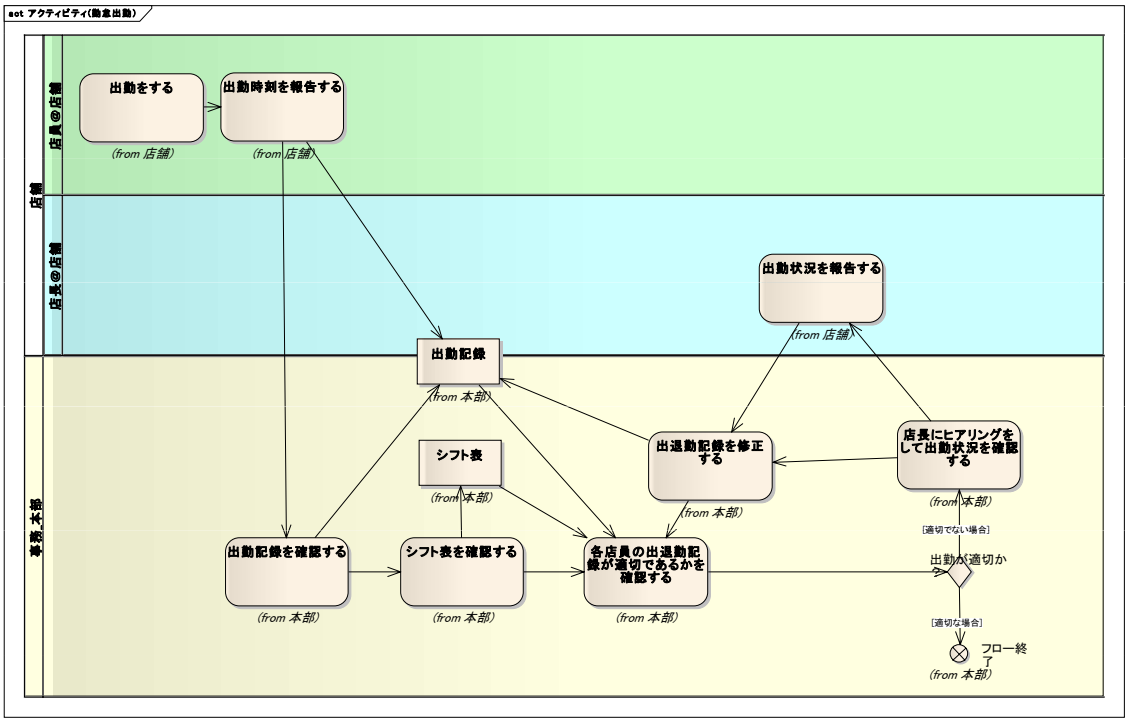
7-5. ユースケース図作成(勤怠管理)

サブジェクトごとにユースケース図をまとめる



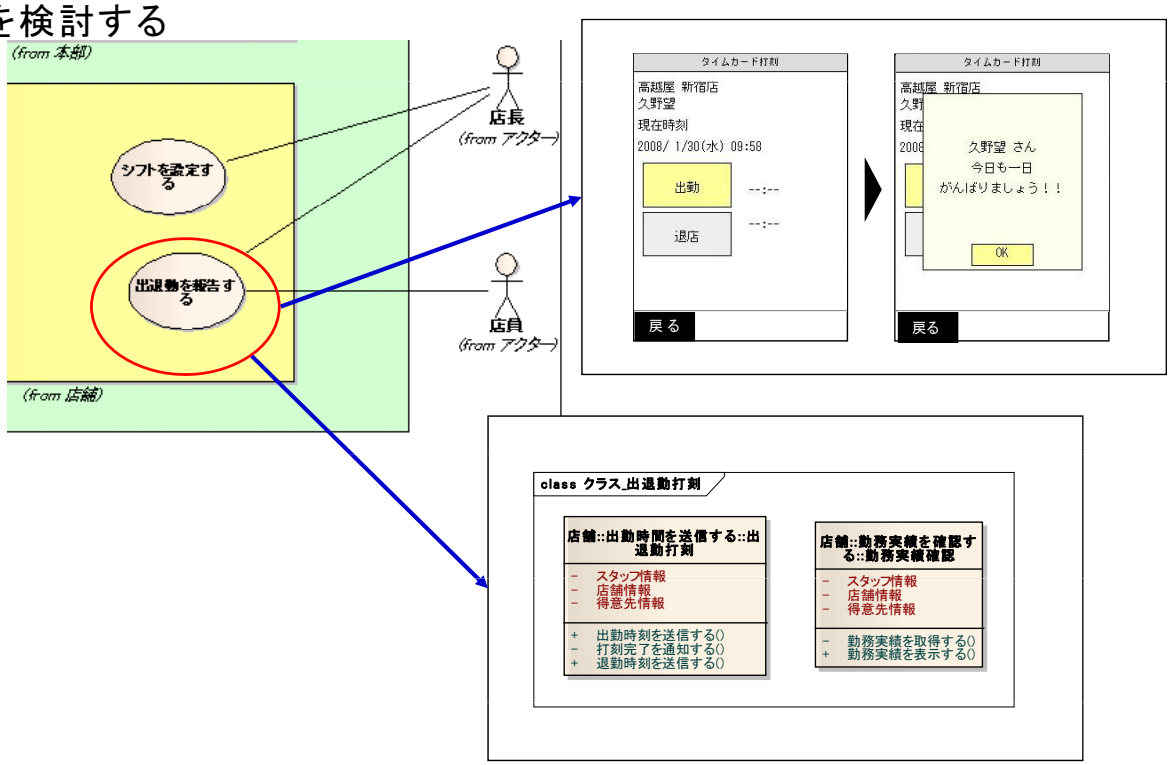
7-6. アクティビティ図作成(勤怠管理)

サブジェクト内のアクターごとにパーティションを作成して機能フローを定義する



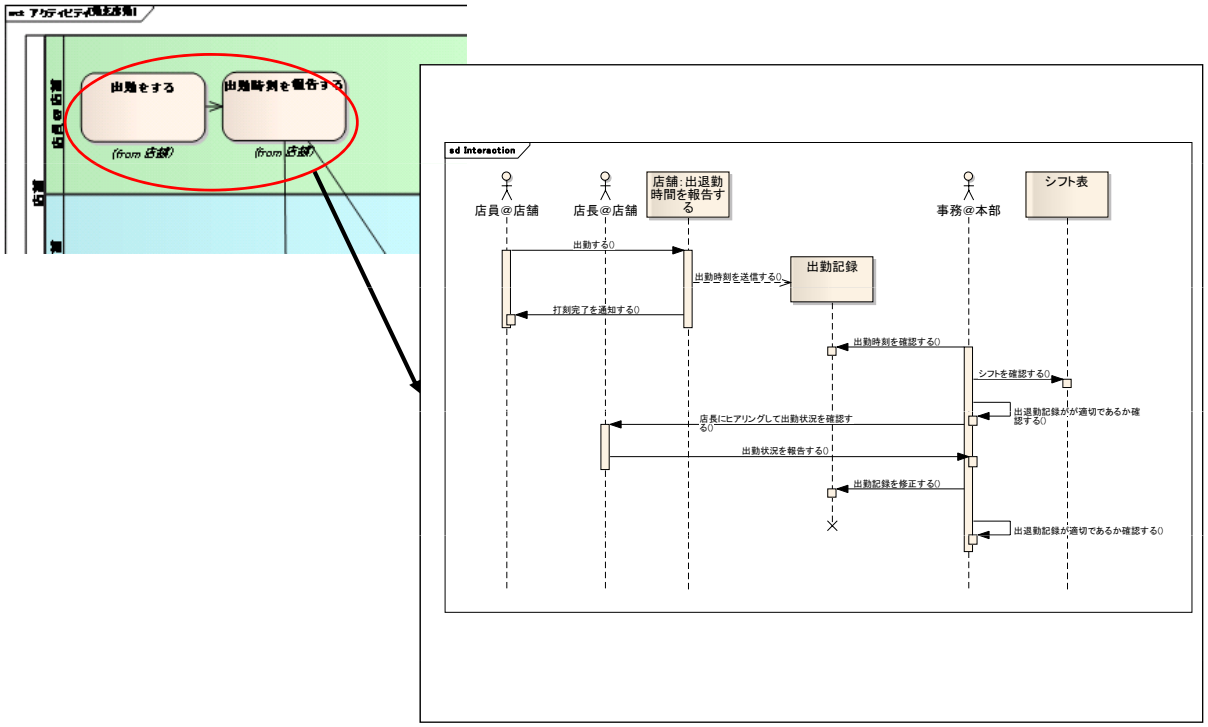
7-7. クラス図作成(勤怠管理)

それぞれのユースケースに対して、クラスを設計し、画面インタフェースを検討する



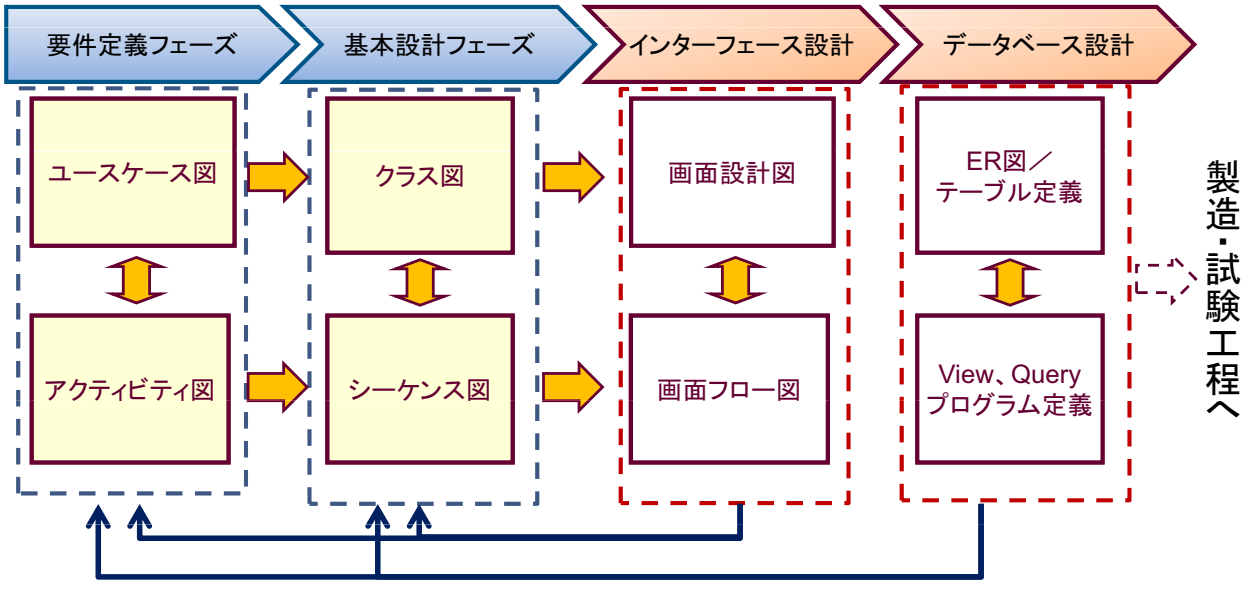
7-8. シーケンス図作成（勤怠管理）

それぞれのアクティビティ図のパーティション内のフローごとに、シーケンス図を作成する



8. UMLモデリング以降の設計・開発

設計したダイアグラムをもとに、ユーザインターフェースとデータベースの設計をおこなう。



UMLダイアグラムの内容をフィードバック検証し、
要件や設計内容が画面やデータ処理に反映されているかを確認

9. ユースケースモデリング活用によるメリット

- ユースケース図を初めに作成することで、他のダイアグラムを作成する上での要求・制約が明らかになるだけでなく、設計工程での他のダイアグラムや設計書の整合性をチェックすることができる。
- 試験工程でのテストケースやテストシナリオを作っていく上でユースケース図やアクティビティ図を参考にすることができる。
- モデリングにあたって、サブジェクト＝業務の場、アクター＝ユーザとして活用していることで、システム化されていない業務に対して現場ユーザへのヒアリングや現地調査の情報共有ツールとして役立てることができる。
- サブジェクトやアクターがわかりづらい場合、プロジェクトメンバー内でブレーンストーミングをおこない、機能やユーザを洗い出すことなどによって、ユースケース図を作成するためのノウハウが蓄積されていった。
- ユースケース図によって「業務ユーザや業務機能の関係性」、アクティビティ図によって「業務遂行におけるユーザや機能の関与タイミング（業務スケジュール）」を洗い出すことができ、チーム開発における要件定義作業のレベルを高めることができる。

10. UMLモデリング活用に向けた課題

- **ユースケース図に限らず、UMLのダイアグラムを成果物や納品物として求めるユーザ企業は少ない。**

  →未だ画面フロー図や画面設計、ER図、テーブル定義書を重視する傾向が強いが、組織的な開発やプロジェクトの引き継ぎにとってUMLの活用は有効

- **ユースケース図の網羅性・精緻さに対して、どの程度まで作り込み、稼働を割いてよいかが難しい。**

  →ユースケースモデリングは後工程の仕様変更の影響分析や試験工数にとっても重要であるため、システム化対象のアクターとサブジェクトを全て洗い出すまで時間をかけることは重要。精緻さは、ユースケース図の数を業務のスケジュールとイベントの組み合わせ数と合わせるなどのアプローチがある。

- **ユースケース図において、アクターやサブジェクトの定義が難しい。**

  →連携・参照する情報システムや、コンピュータ、プリンタや各種帳票などの業務機能遂行に必要なアイテムをユースケース図上に盛り込むと、解釈が異なる場合や、システム境界が不明確になることがある。ヒトの形をしたアクターに対して情報システムを擬人化してアクターとして定義する場合には注意が必要。