

マイクロサービスによるシステム設計

富士通株式会社

澤村大輝

日本ユニシス株式会社

磯崎紘

日本ユニシス株式会社

廣田大理

リコーITソリューションズ株式会社

吉村僚祐

課題、取り組み

システムが肥大化すると、軽微な修正に対して甚大なコストがかかることがある。その問題に対応するために、本演習では商社の受発注システムの例題に対してドメイン駆動設計(DDD)とマイクロサービスアーキテクチャ(MSA)を適用することでシステムの変更容易性が高まることを確認する。

実践結果

ドメイン駆動設計とマイクロサービスシステム設計を適用することにより、**影響範囲調査と改修作業が容易になり変更容易性が高まる**ことがわかった。一方で、実現するにあたっては**データ連携方式などの考慮点があり、十分な検討が必要である**と考察した。

分割方針の検討

基本的なDDDでのドメイン分割方針

境界づけられたコンテキストで分割

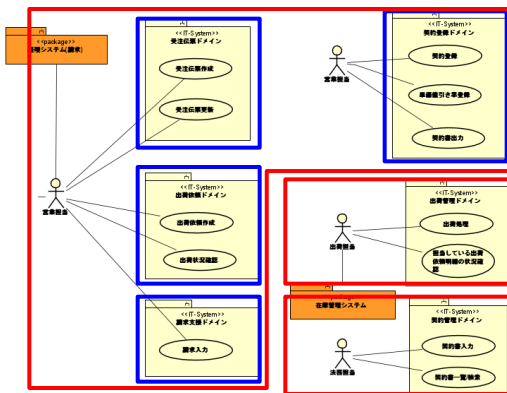
具体化

適用した分割方針

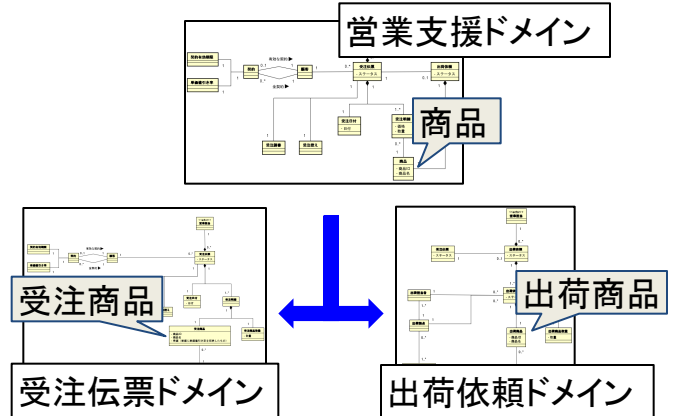
- ・1つのドメインを取り扱う**部門／アクター**が複数いるのなら分割する
- ・その上で、1つのドメインの中に依然として多くのユースケースが紐づいている場合は、**ユースケースのまとまり**を考慮して分割する

ドメイン、エンティティの分割

初期のビジネス概念モデル、ユースケース図を「**部門／アクター**」、「**ユースケースのまとまり**」の二軸で分割。
 ドメインの分割に合わせてエンティティについても分割。
 ex. 商品エンティティ ⇒ 受注商品エンティティ / 出荷商品エンティティ



ユースケース図



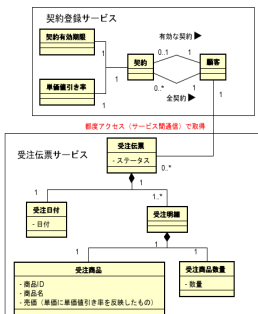
ビジネス概念モデル

サービス間のデータ連携方式

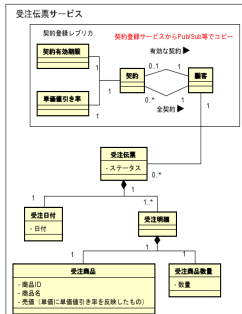
サービス間のデータ連携方式として、他サービスに「都度アクセス」するか、自サービス内に「レプリカを保持」するか、業務要件に合わせて各サービス連携箇所適切に選択する必要がある

考察

MSAにしたことで**自サービスに関係ない情報**がサービス内に存在しない
 ⇒ **モノリシックアーキテクチャに比べて影響調査や改修が容易になる**

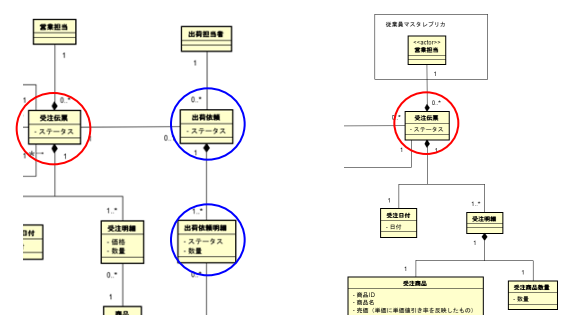


都度アクセス



レプリカを保持

	都度アクセス	レプリカを保持
整合性	○ 常に最新のデータ	△ 更新されると不整合
可用性	✖ 他サービスに依存	○ 他サービスに依存しない
変更時の影響範囲	○ 2箇所	△ 3箇所以上



モノリシック

マイクロサービス