

マイクロサービスによるシステム設計

ドメインモデルを中心とした設計(A1G4)

葛原卓 : 三菱電機ソフトウェア株式会社 山川幸紘 : 株式会社デンソー 津田一磨 : 株式会社 東芝
 平川康之 : 富士通株式会社 平山貢大 : キヤノンメディカルシステムズ株式会社

開発における問題点

近年、システムの複雑化に伴い、ドメイン駆動開発を活用したマイクロサービスアーキテクチャによるシステム開発が注目を集めている。

しかしながら、ドメイン知識を取り込んで適切にサービスを分割する難易度が高く、サービス分割が不適切だと、仕様変更や機能追加の際に、変更容易性が低いシステムとなる懸念がある。

手法・ツールの適用による解決

ビジネスサイドと開発者が協力してシステム化の対象ビジネスを分析し、境界づけられたコンテキストを抽出する「**イベントストーミング**」手法を活用し、ドメインモデルを作成した。

このドメインモデルを元にアーキテクチャを設計したことで、ビジネスが大幅に変わらない特定ドメインへの変更に対応できるアーキテクチャを実現した。

*コンテキスト：あるモデルを適用して問題を解決する領域のこと

アプローチ

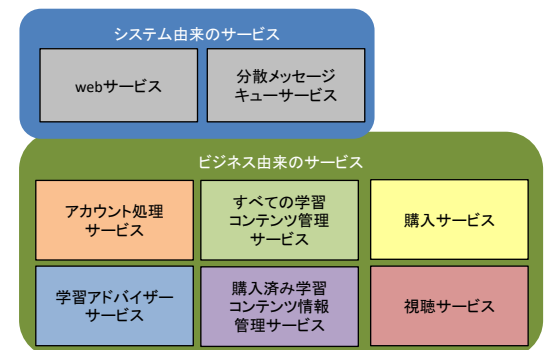
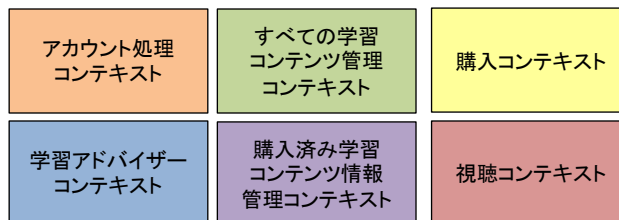
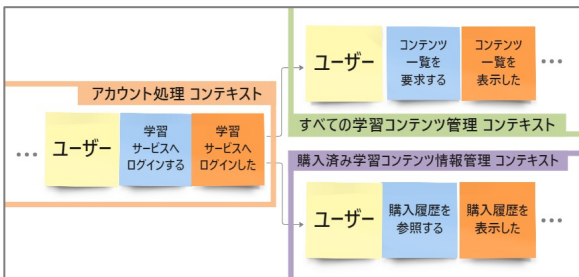
ドメイン分析

ドメインモデル作成

マイクロサービスアーキテクチャ設計

ドメインモデルまで戻って再設計

仕様変更発生



イベントストーミングを採用
 ビジネスサイドとドメイン知識を共有し、
境界づけられたコンテキストを抽出

イベントストーミングで抽出した
 境界づけられたコンテキストから
ドメインモデルを作成

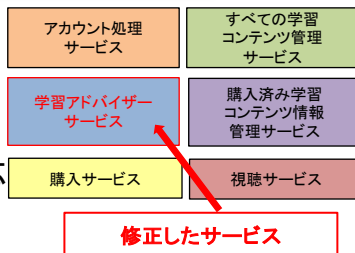
**ドメインモデルを元にした
 マイクロサービスアーキテクチャ設計**

評価と考察

① 設計結果の変更容易性 評価○

特定ドメインにおける仕様変更であれば、該当サービスの修正のみで対応可能。

▶ 学習アドバイザーサービスの修正で対応



② 設計結果の可用性

評価○：学習コンテンツ管理を2サービスに分割した

・すべての学習コンテンツ管理がダウンしても、購入済みコンテンツは使用可能

評価△：履歴データを購入済みコンテンツ情報管理経由で展開

・このサービスがダウンする検証時に、可用性の問題を検出
 ・イベント発生を起点にデータを展開するべきだった

③仕様変更時の再設計プロセス

評価○：ドメインモデルから仕様変更に対応

・イベントストーミングには戻らなかった

実際の業務では、立ち上げ開発よりも既存システムへの機能追加が主であることを想定し、再設計する工程を設定
 ・仕様変更に対応できた

ドメインモデルとアーキテクチャを対応付けて設計したこと、ビジネスが大きく変わる仕様変更でなかったことが理由に挙げられる

今後の課題

- ① 本当に変更に対して強かったのかの検証(実装)
- ② 可用性(購入履歴/学習履歴の持ち方)
- ③ 仕様変更時にどの工程まで戻って変更を加えるべきか