ドメイン駆動設計の実践

2024年7月25日

有限会社システム設計　増田 亨

自己紹介

増田 亨（ますだ とおる）

業務系アプリケーションソフトウェアの開発者

モデル駆動設計

Java/Spring Boot/IntelliJ IDEA/JIG

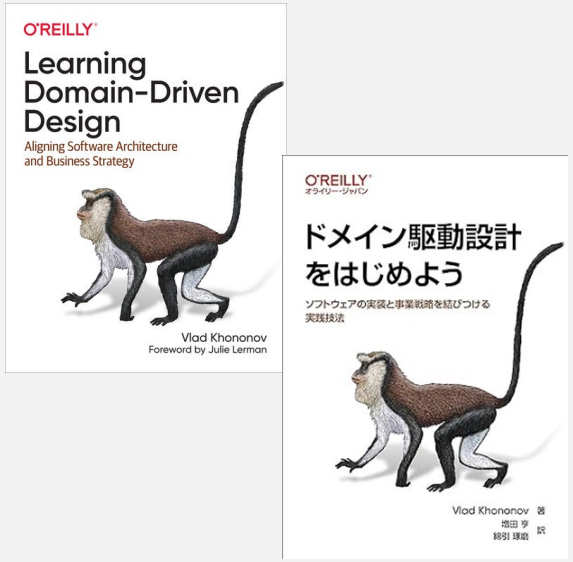
有限会社システム設計 代表
コミューン株式会社 技術アドバイザ

訳書

著書

『ドメイン駆動設計をはじめよう』

ソフトウェアの実装と事業戦略を結びつける実践技法



高評価“Learning Domain-Driven Design”の全訳

「ドメイン駆動設計を実践するために
最初に手にするべき1冊！」 by 出版社

ドメイン駆動設計抜きにしても、よいことが書いてある　by 訳者

今日の内容

① 何が書いてあるか？

② 事業活動の分析（１章）⇒**設計判断** ５章、６章、７章、８章、10章

③ 業務知識の発見（２章）

④ 事業活動の複雑さに立ち向かう（３章）

⑤ 区切られた文脈どうしの連係（４章）⇒実装方法 ９章

⑥ 事業の成長とソフトウェアの成長（１１章）⇒１章、５章、６章、７章

⑦ 開発チームの学習と成長（付録A）⇒１章、２章、３章、５章、６章、７章

⑧ ドメイン駆動設計の分散型アーキテクチャ（14章、15章、16章）⇒３章、４章


O'REILLY
オライリー・ジャパン
ドメイン駆動設計
をはじめよう
ソフトウェアの実装と事業戦略を結びつける
実践技法
Vlad Khononov 著
増田 亨
綾引 琢磨 訳

何が書いてあるか？

書いてあることは基本的には一つ

事業活動とソフトウェアを

一緒に発展させていくための方法

この本の主題であり、最初から最後まで一貫している

もう少し解像度をあげると

事業活動とソフトウェアを一緒に発展させていく

第Ⅰ部

基本となる考え方

第Ⅱ部

実装方法の選択

第Ⅲ部、付録A

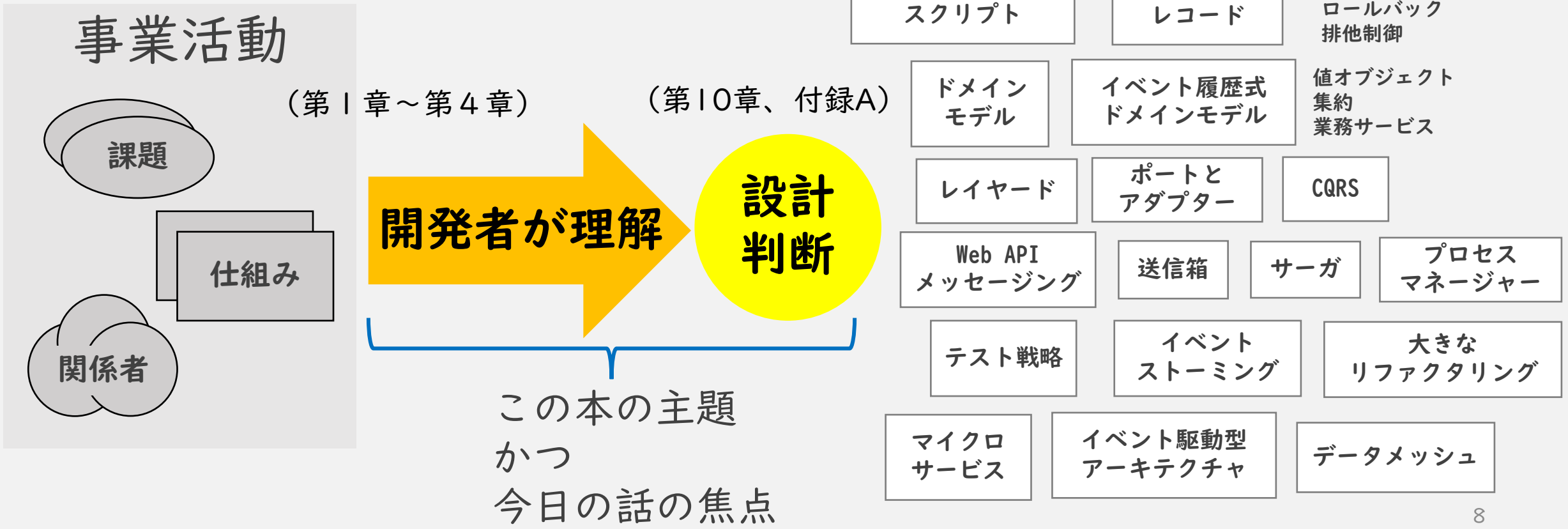
現場での取り組み方

第Ⅳ部

分散型システムへの挑戦

さらに解像度をあげると

開発者が事業活動を理解して、その理解を設計判断に活かす



この本の価値

次世代のドメイン駆動設計本

- 『ドメイン駆動設計』(原著2003年）、『実践ドメイン駆動設計』(原著2013年）が出版されて以降のさまざまな実践知を体系的に整理
- イベントソーシング、CQRS、イベントストーミング、マイクロサービス、データメッシュなど、最近の分散型アーキテクチャとの関係を具体的に説明

わかりやすく現場で取り組みやすいドメイン駆動設計本

- なぜそうするか（Why）の説明が一貫している
- 業務知識の具体例が豊富
- 多様な設計課題を相互に関係づけて解説している
- 要点を図、コード、表でわかりやすく説明している

誰が読むべき本か？

すべてのソフトウェア開発者の役に立つはず

しかし

- 開発者としての知識や経験の違いによって、**理解できる範囲が異なる**
- 現在の置かれている状況や当面の課題によって、**役に立つ個所が異なる**

だからこそ

- 現在の自分の知識、関心、必要性を**この本と対比してみる**
- チームで、お互いの理解の違い、関心の違いを**この本を題材に話し合ってみる**
- もっと経験を積んでから**読み直してみる**

開発者が事業活動を理解して

その理解を設計判断に活かす

事業活動を理解するための基本用語

原語	従来の訳語	**本書の訳語**
domain	ドメイン	**事業活動**
subdomain	サブドメイン	**業務領域**
domain logic	ドメインロジック	**業務ロジック**
core domain	コアドメイン	**中核の業務領域**
ubiquitous language	ユビキタス言語	**同じ言葉**
bounded context	境界づけられたコンテキスト	**区切られた文脈**
context map	コンテキストマップ	**文脈の地図**

技術者がドメインとサブドメインを理解し
境界づけられたコンテキストでユビキタス言語を使って
ソフトウェアを開発する

技術者が事業活動と業務領域を理解し
区切られた文脈で同じ言葉を使って
ソフトウェアを開発する

事業活動の分析（第１章）

よくある質問

出典：『ドメイン駆動設計をはじめよう』第１章の導入文

私が講師をしているドメイン駆動設計の講座でこの内容の説明を始めると、「私たちはソフトウェアを書くのが仕事です。事業経営をしているわけではありません。これを学ばないといけないんですか？」という質問が必ずでてきます。その答えははっきりしています。必要です。役に立つ解決策は課題を理解するところから生まれます。私たちソフトウェア開発者にとっての課題はどんなソフトウェアを作るかです。そしてどんなソフトウェアを作るべきかを理解するには、ソフトウェア開発の目的と背景を理解することが必要です。事業方針は何か、そしてソフトウェア開発によってどういう価値を手に入れようとしているかの理解です。

事業活動と業務領域

事業活動（ドメイン）

- 顧客にどんな価値を提供しているか
- どうやってその価値を提供しているか
- 競合他社とどうやって**差別化し競争優位**を生み出し維持するか

業務領域（サブドメイン）

- 事業活動の**領域全体を細分化**したもの
- すべての業務領域が一体となって顧客に価値を提供する
- 販売促進、販売、顧客サービス、出荷、在庫、研究開発、財務、人事

技術的な見方をすれば

業務領域とは関連したユースケースの集まり

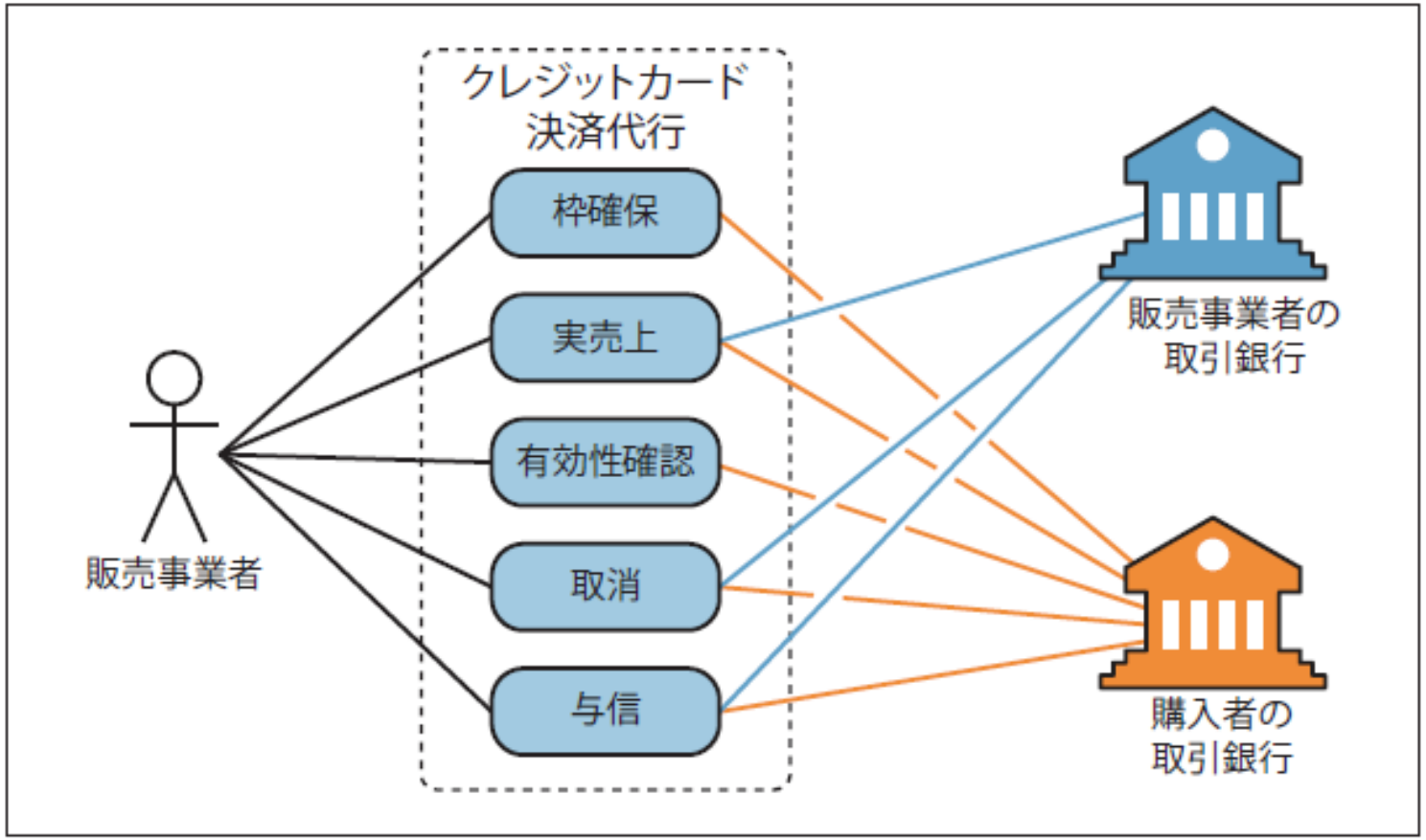


図1-3　クレジットカード決済のユースケース図

分析技法：業務領域のカテゴリー分け

中核の業務領域
- 競争優位の源泉
- 業務ロジックが複雑
- 変化を繰り返す

一般的な業務領域
- 他社と同じやり方でよい
- 業務ロジックは複雑
- あまり変化しない

補完的な業務領域
- 自社独自のやり方が必要
- 業務ロジックは単純
- あまり変化しない

業務領域の細分化と

三つのカテゴリーへの分類が

設計判断の基本枠組みとなる

対象の業務領域のカテゴリーを特定する

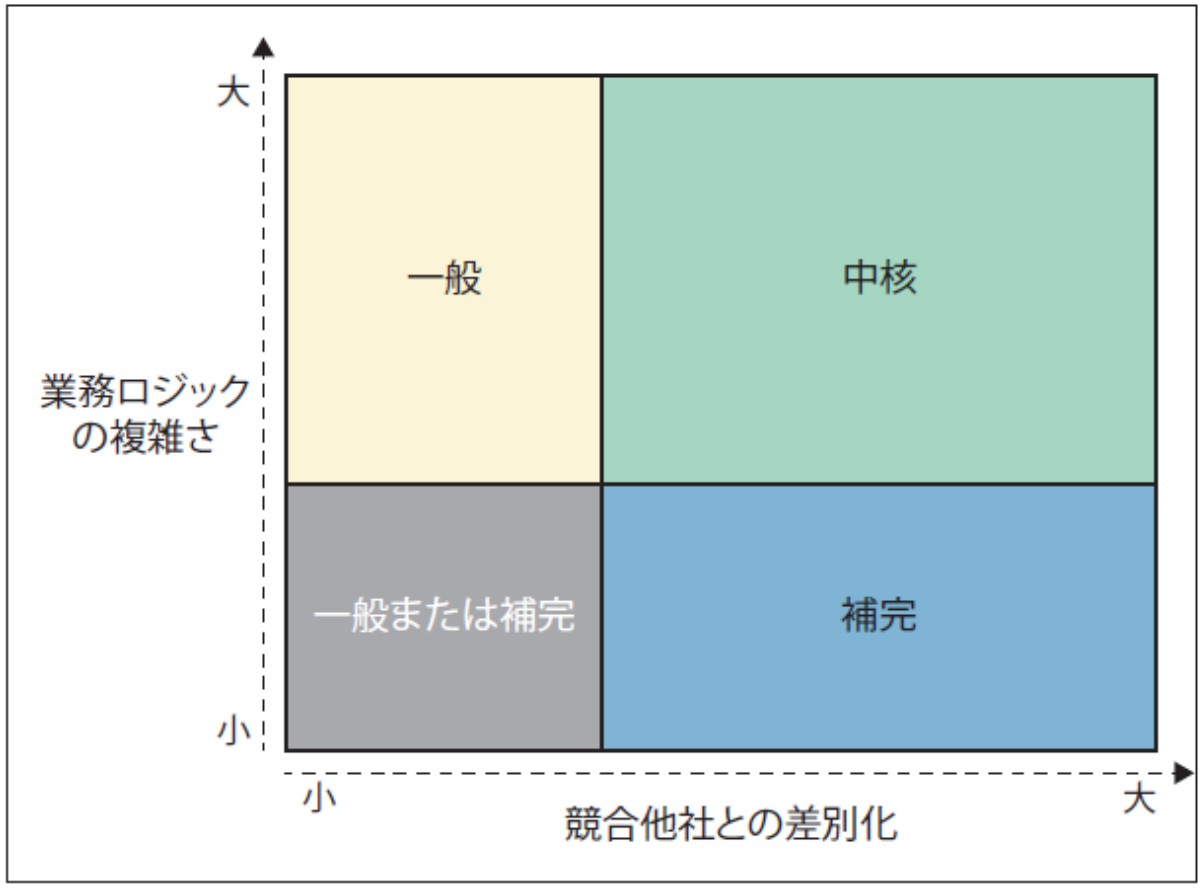


図1-1 差別化の度合いと業務ロジックの複雑さによる業務領域の分類

業務ロジックの複雑さと

競合他社との差別化の二軸で分類

業務ロジックの実装方法の判断に使う

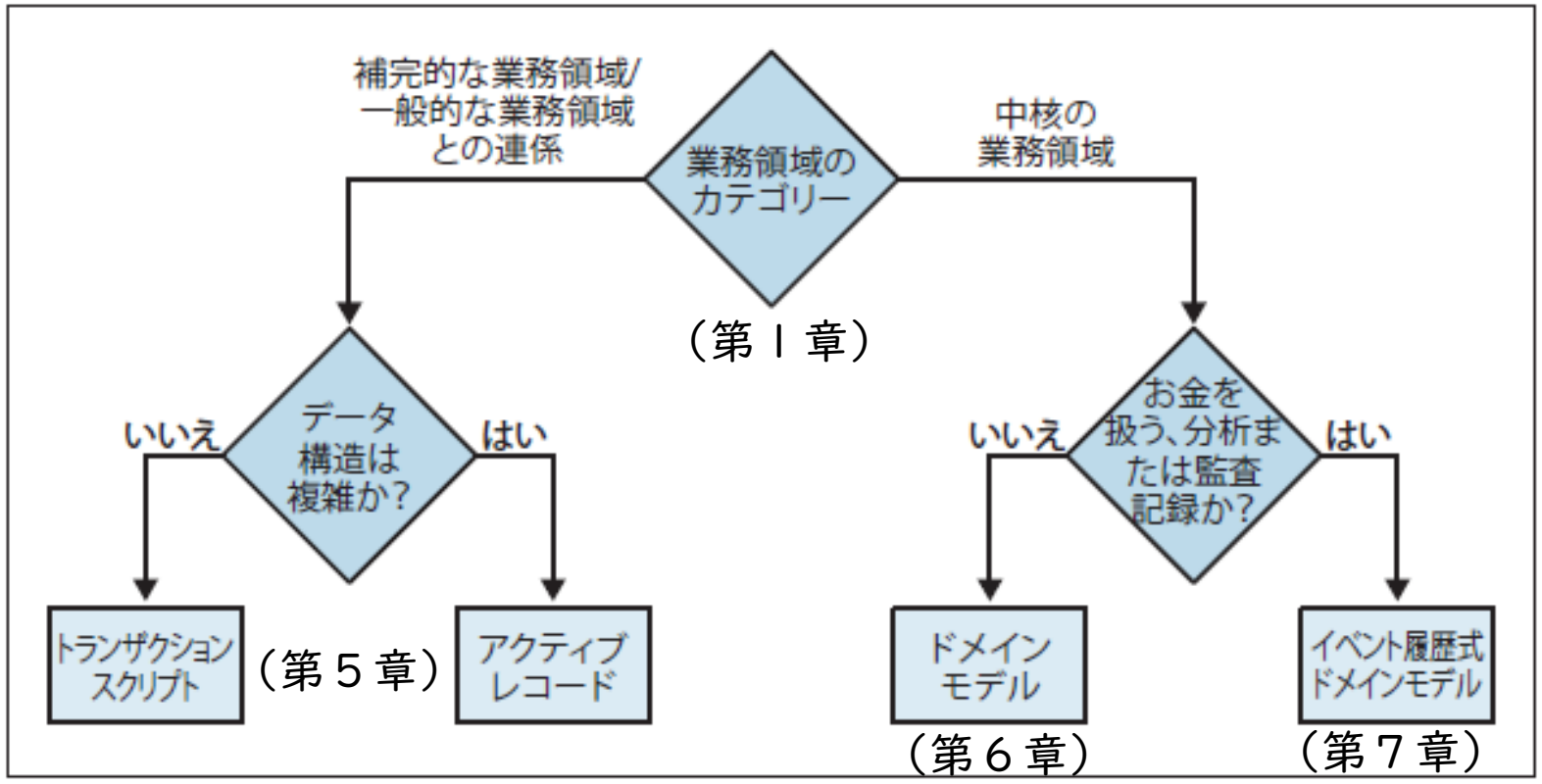


図10-3　業務ロジックの実装方法の判定方法

業務領域のカテゴリーから
業務ロジックの実装方法が決まると

アプリケーションの技術方式が決まる

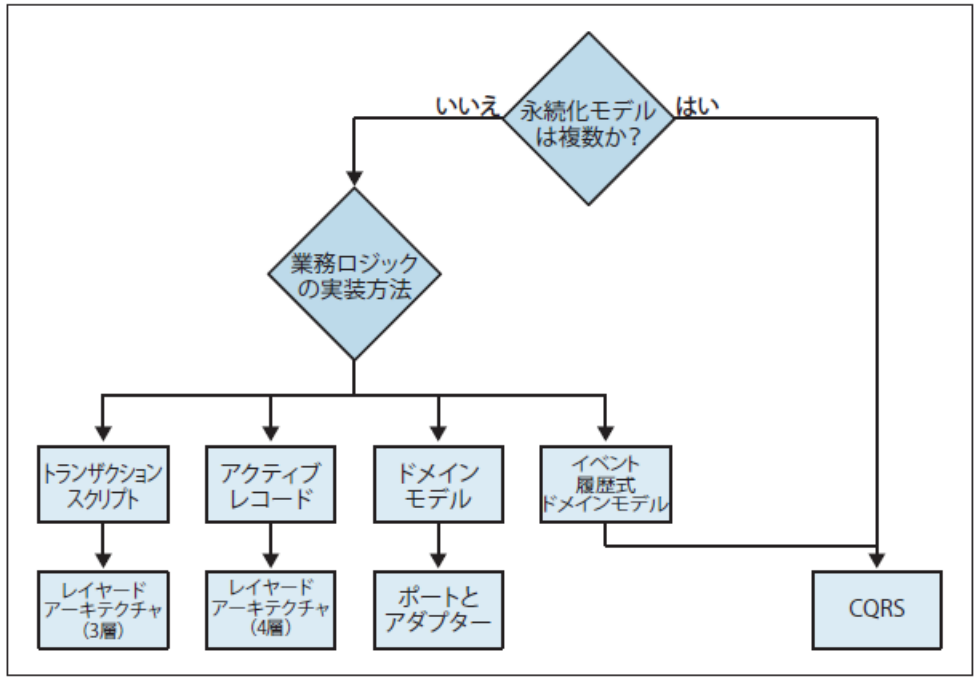


図10-4　技術方式の判定方法

（第８章）

テストの基本方針が決まる

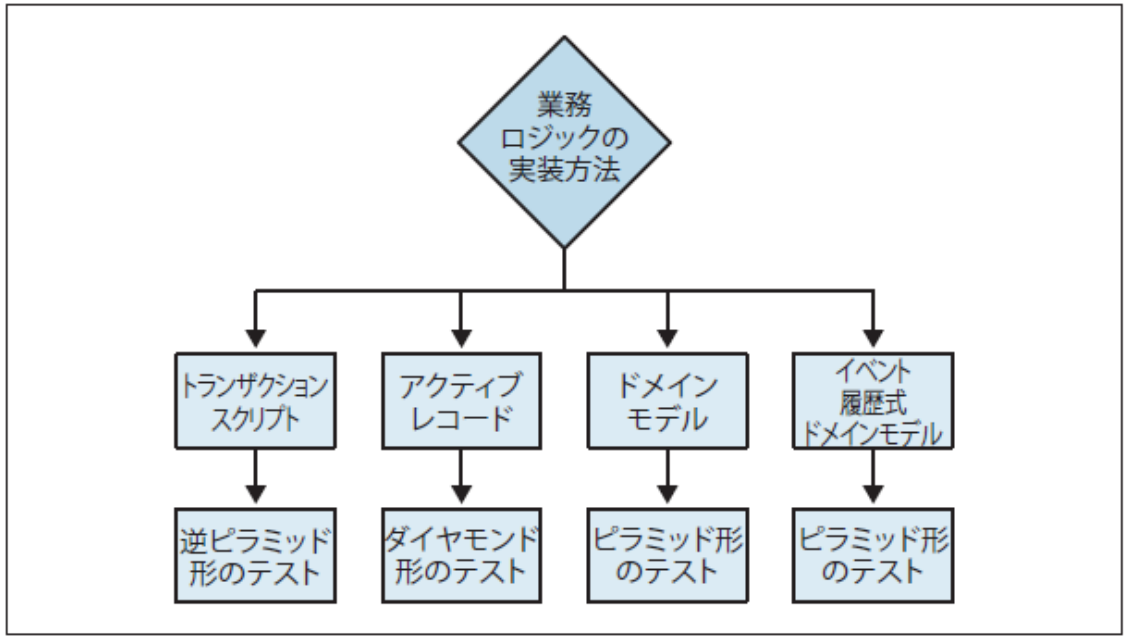


図10-6　テスト方針の判定方法

（第１０章）

業務領域の分類の具体例（第１章）

ライブチケットのオンライン販売事業

中核
- 推薦エンジン
- データの匿名化
- モバイルアプリ

一般
- 暗号化
- 会計
- 決済
- 認証と認可

補完
- 音楽ストリーミングサービスとの連係
- SNSとの連携
- ライブ参加履歴の管理

相乗りタクシー型ミニバスサービス

中核
- 運行経路の選択
- 利用者の行動分析
- モバイルアプリ
- 車両の管理

一般
- 交通状況
- 会計
- 請求
- 認証と認可

補完
- クーポン発行
- クーポンの有効性チェック

業務知識の発見（第２章）

伝言ゲーム式の開発

- 開発者が業務知識なしで設計し実装するための技法

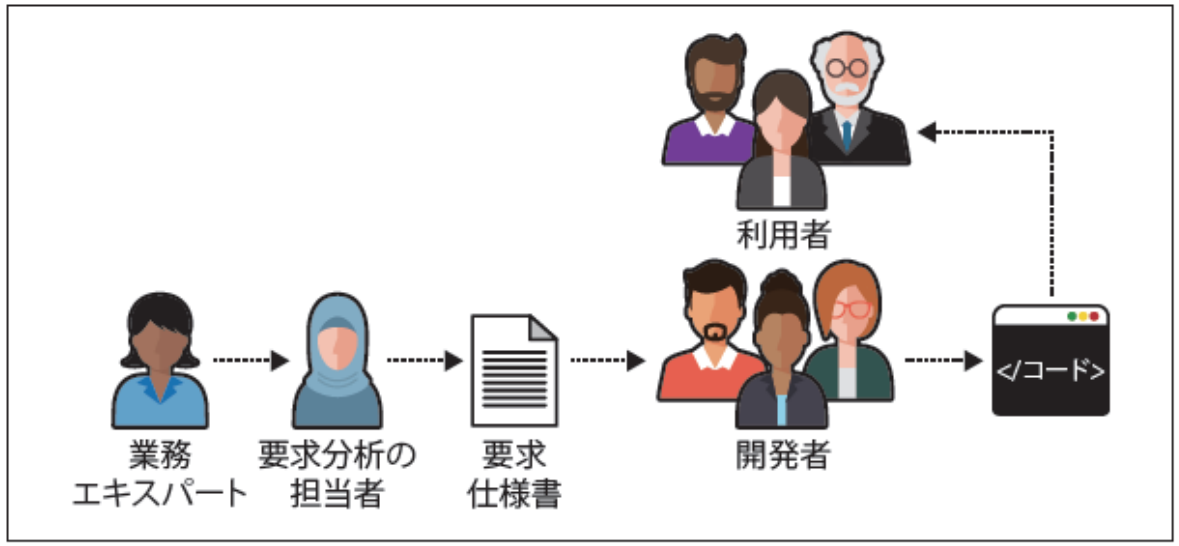


図2-1　ソフトウェア開発における典型的な知識の共有

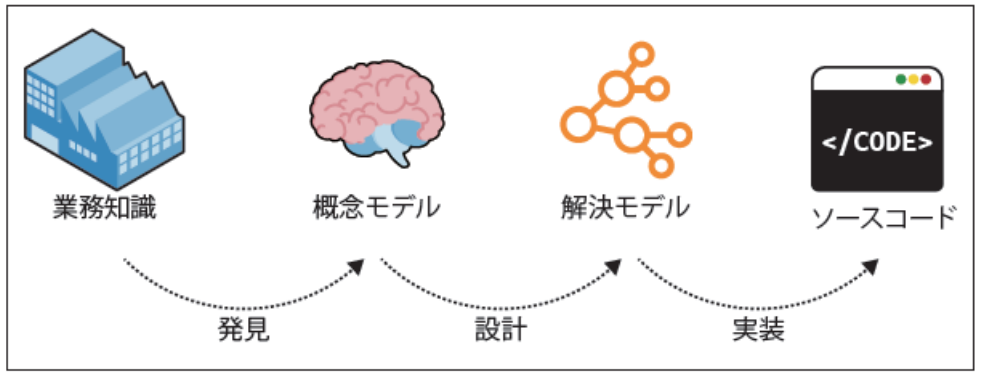
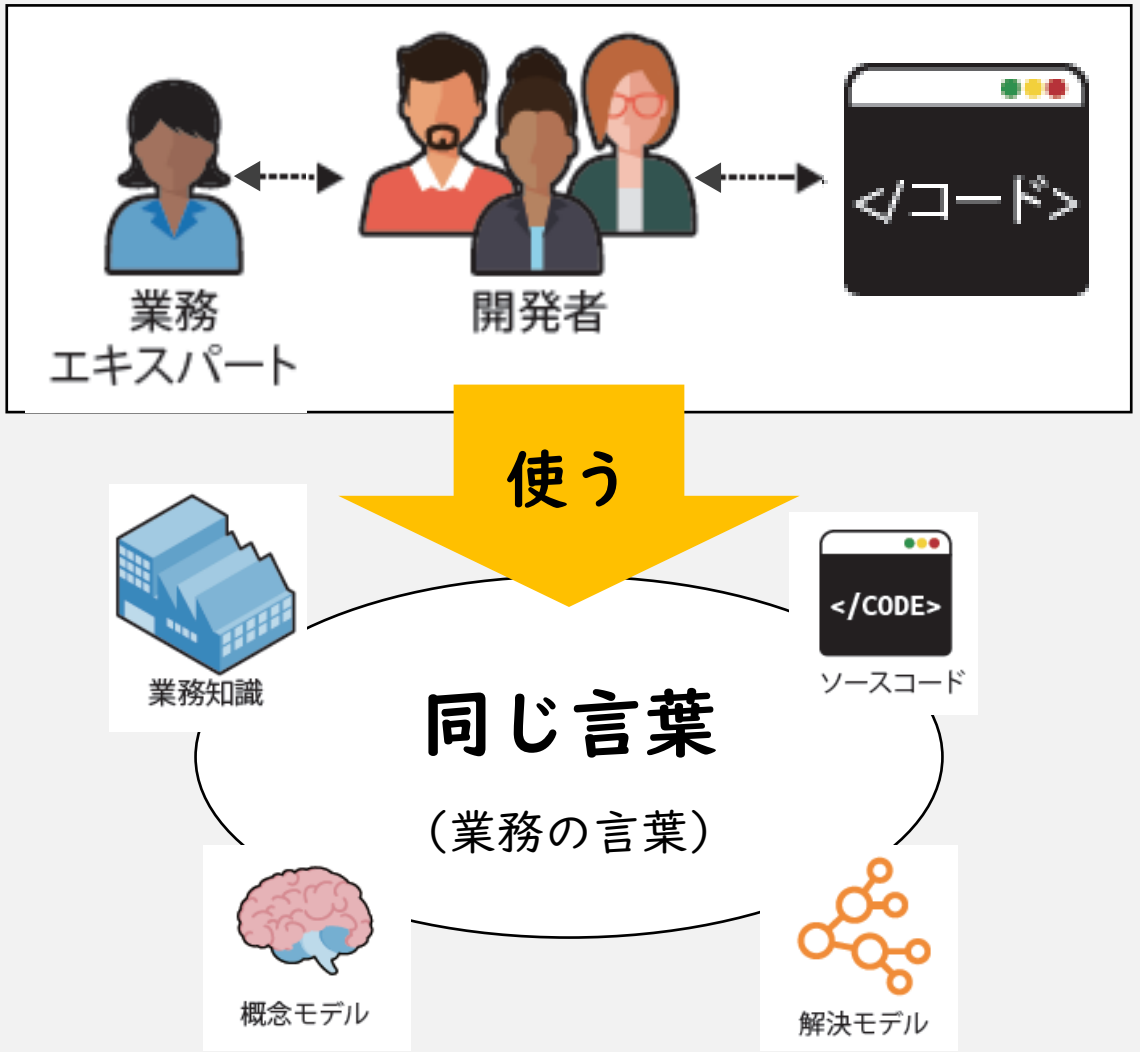


図2-2　モデルの変換

ドメイン駆動設計：同じ言葉を使って開発する



同じ言葉を使って開発する

業務知識なしのソフトウェア開発

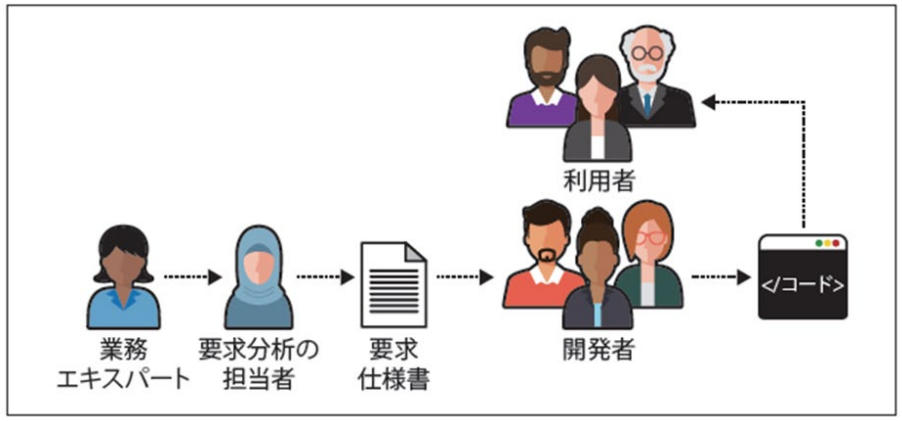
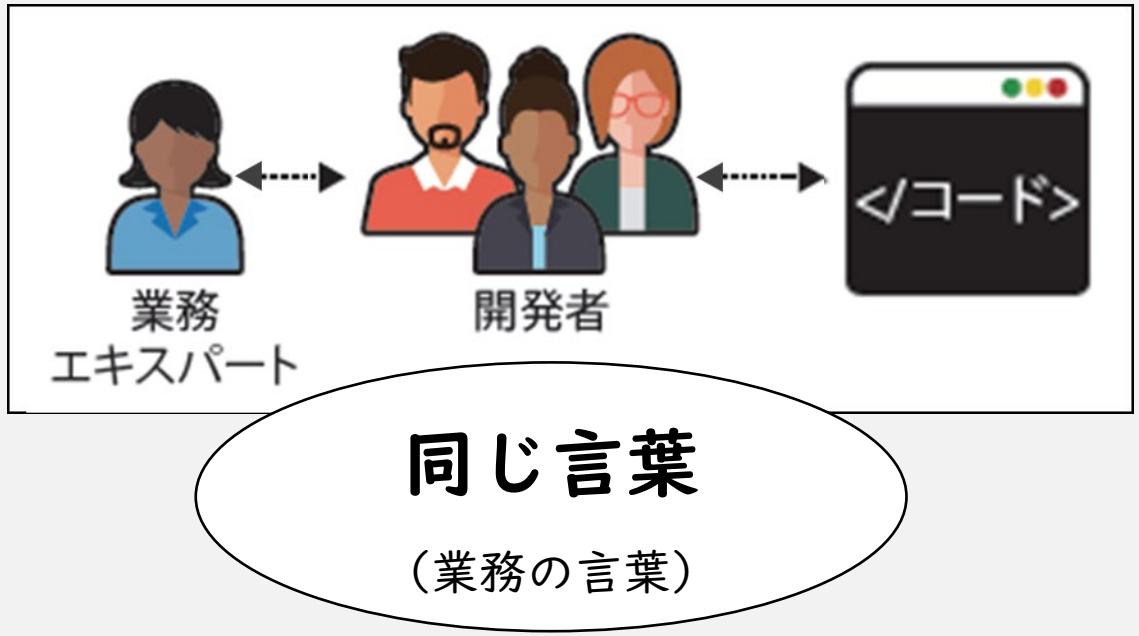


図2-1　ソフトウェア開発における典型的な知識の共有

業務知識が豊富なソフトウェア開発



事業活動の複雑さに立ち向かう

（第３章）

巨大な全体モデルで複雑さに立ち向かう

図3-1　事業活動の例：テレマーケティング

うまくいかない！

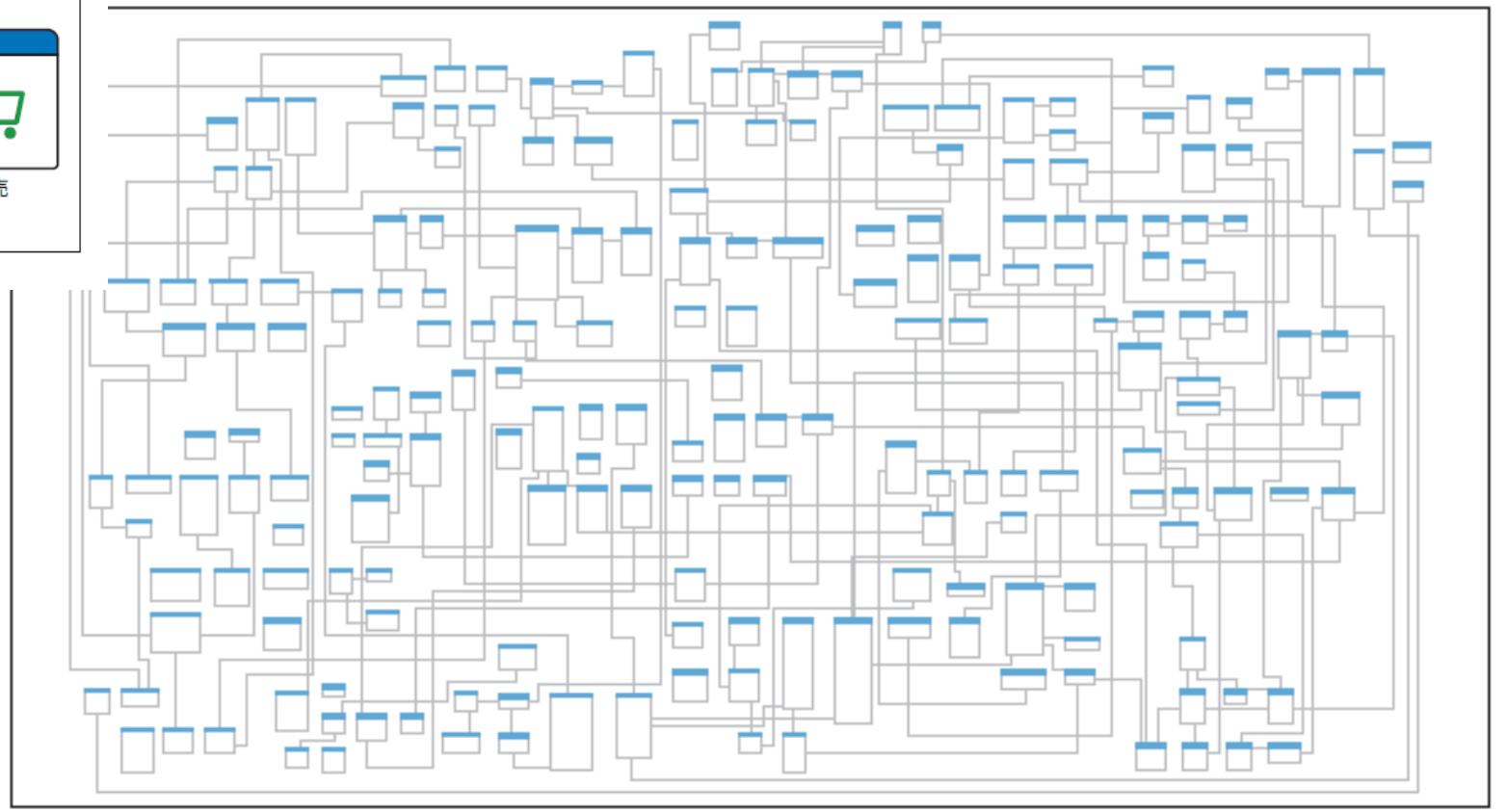


図3-2　事業全体を表現するデータモデル図

技法：文脈を区切って複雑さに立ち向かう

図3-1　事業活動の例：テレマーケティング

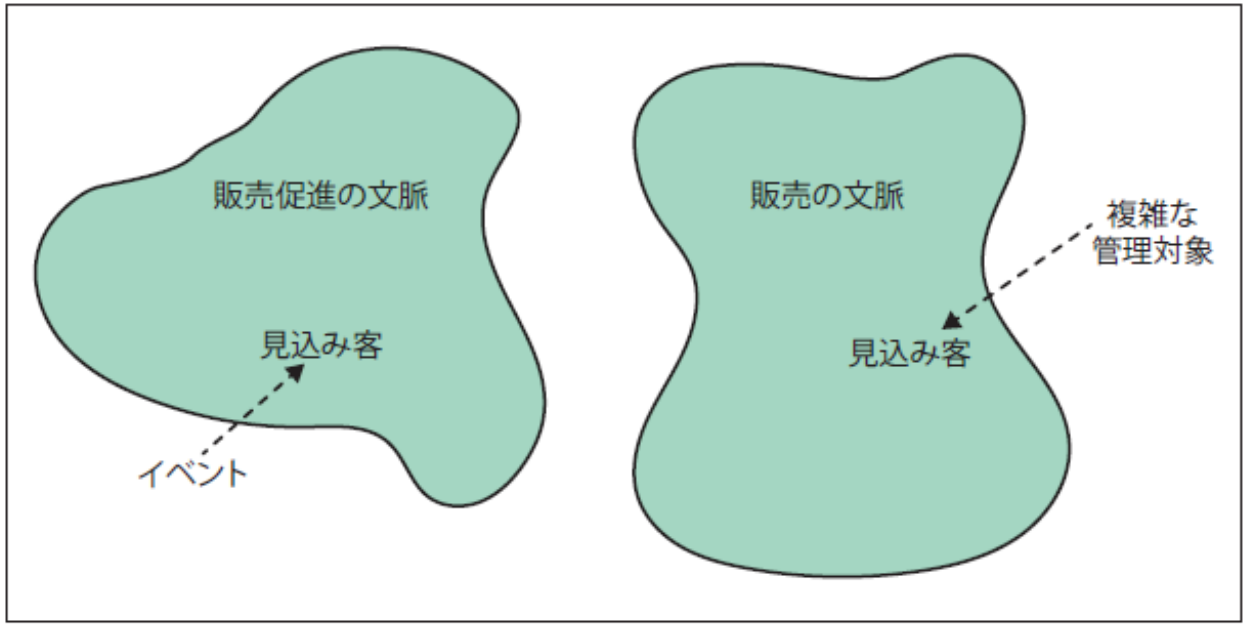


図3-3　文脈を区切って言葉の意味のあいまいさを取り除く

- 一つの言葉が業務エキスパートによって異なるとらえ方となる可能性がある
- 同じ言葉を複数の文脈に区切ることで、言葉の曖昧性や多義性を解消する
- 同じ言葉の通用範囲（関心の範囲）が全社に広がることはない

区切られた文脈と業務領域の関係

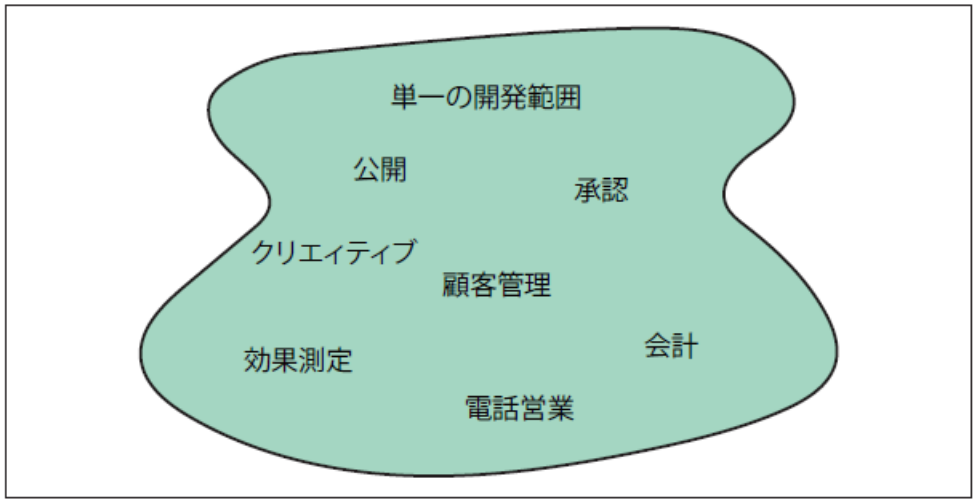


図3-5　単一の文脈で開発する

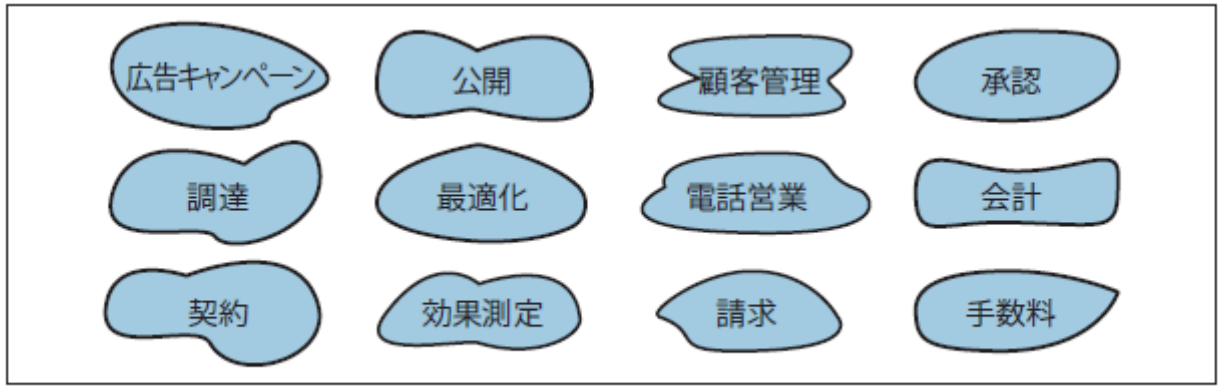


図3-7　業務領域に合わせて文脈を区切る

- 区切られた文脈が一つの開発単位となる
- 一つの区切られた文脈に複数の業務領域を含めることがある
- 業務領域ごとに文脈を区切る場合もある

事業活動から業務領域を発見し、ソフトウェアを開発するために区切られた文脈を設計する

区切られた文脈ごとに一つのチームが担当する

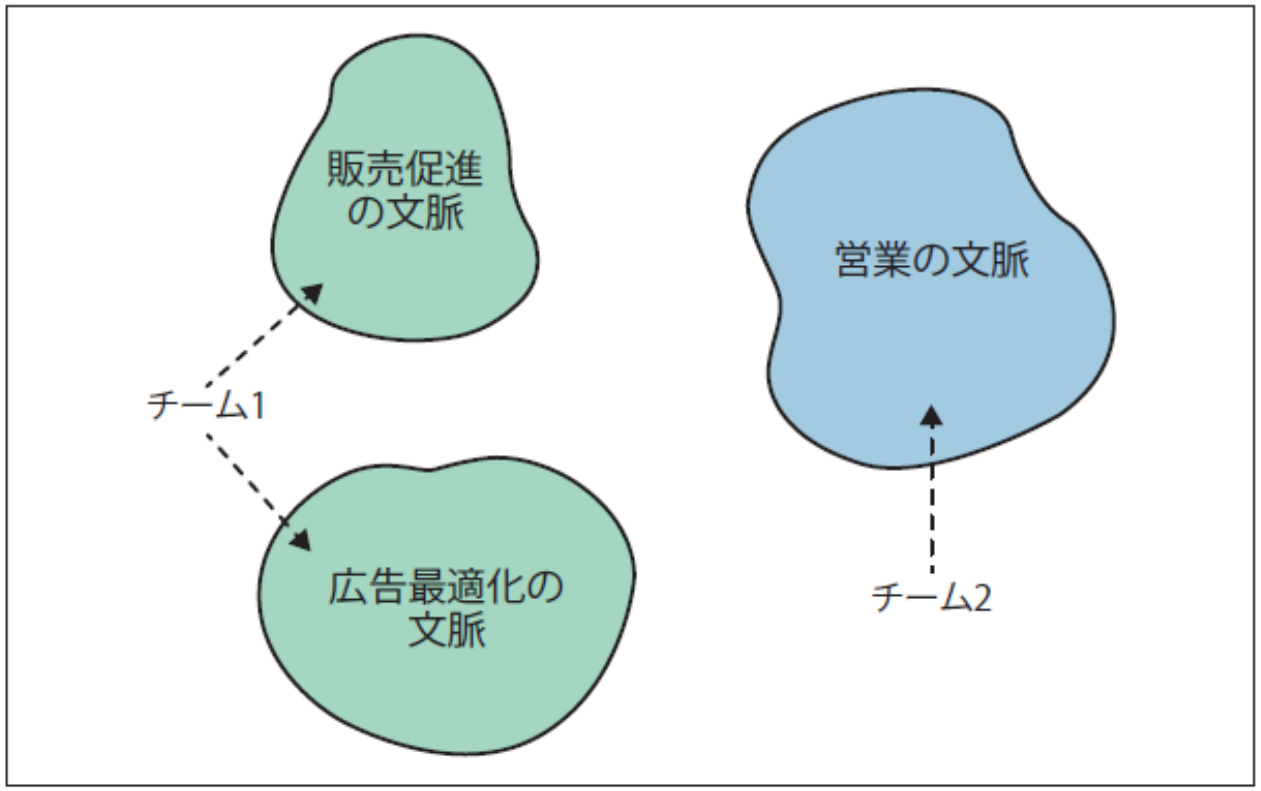


図3-8　チーム１が販売促進と広告最適化を担当し、チーム２は営業活動を担当する

区切られた文脈

- 同じ言葉の通用する範囲
- モデルの境界
- 開発単位の境界
- チームの責任範囲の境界

区切られた文脈どうしの連係

（第４章）

課題

- 多くのアプリケーションを**うまく連係させる**必要がある

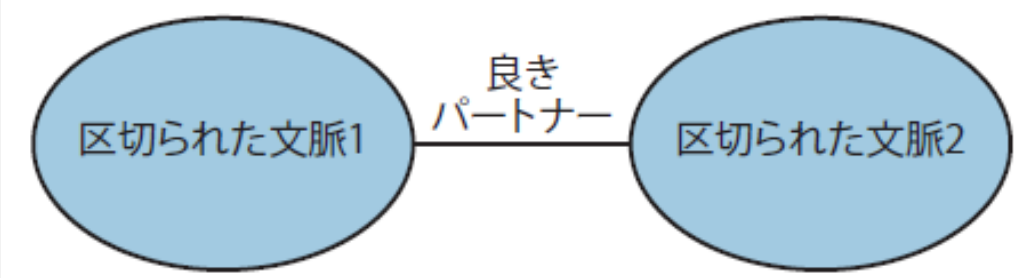
しかし

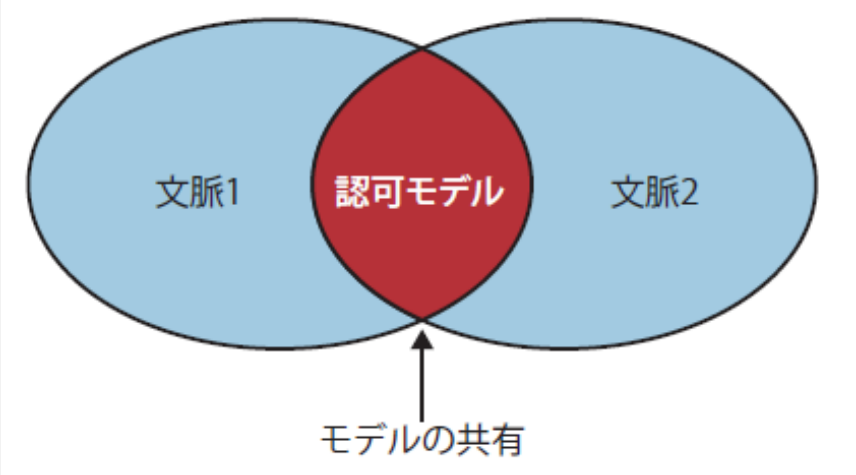
- それぞれの区切られた文脈ごとに、**独自のモデル**で開発
- それぞれの区切られた文脈ごとに、**異なるチーム**が開発

独自のモデルと異なるチームという状況で、
どうすれば区切られた文脈を**うまく連係**できるか？

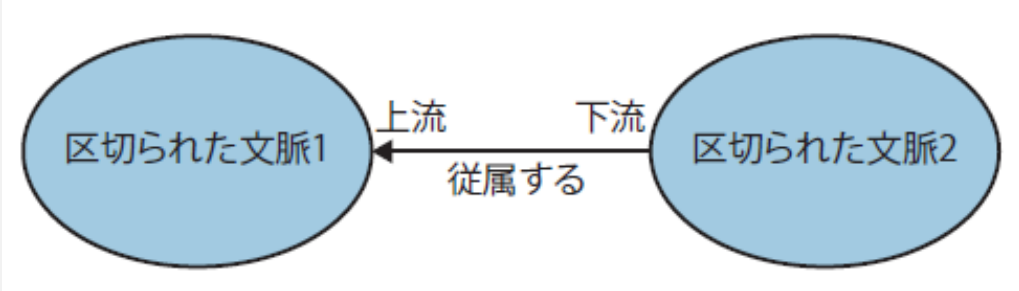
連係方法の選択肢

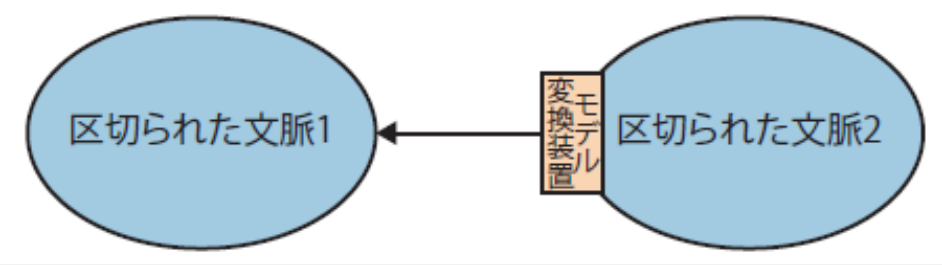
対等の関係

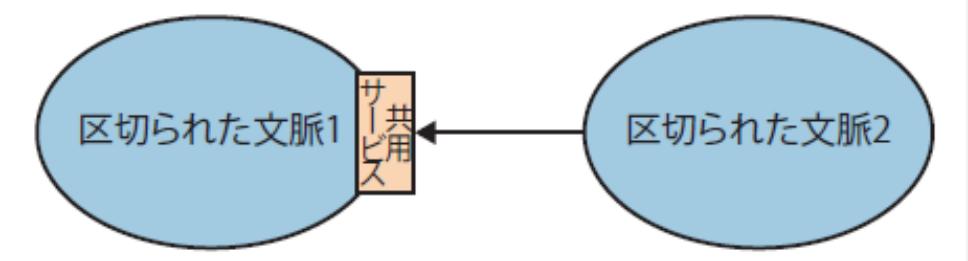




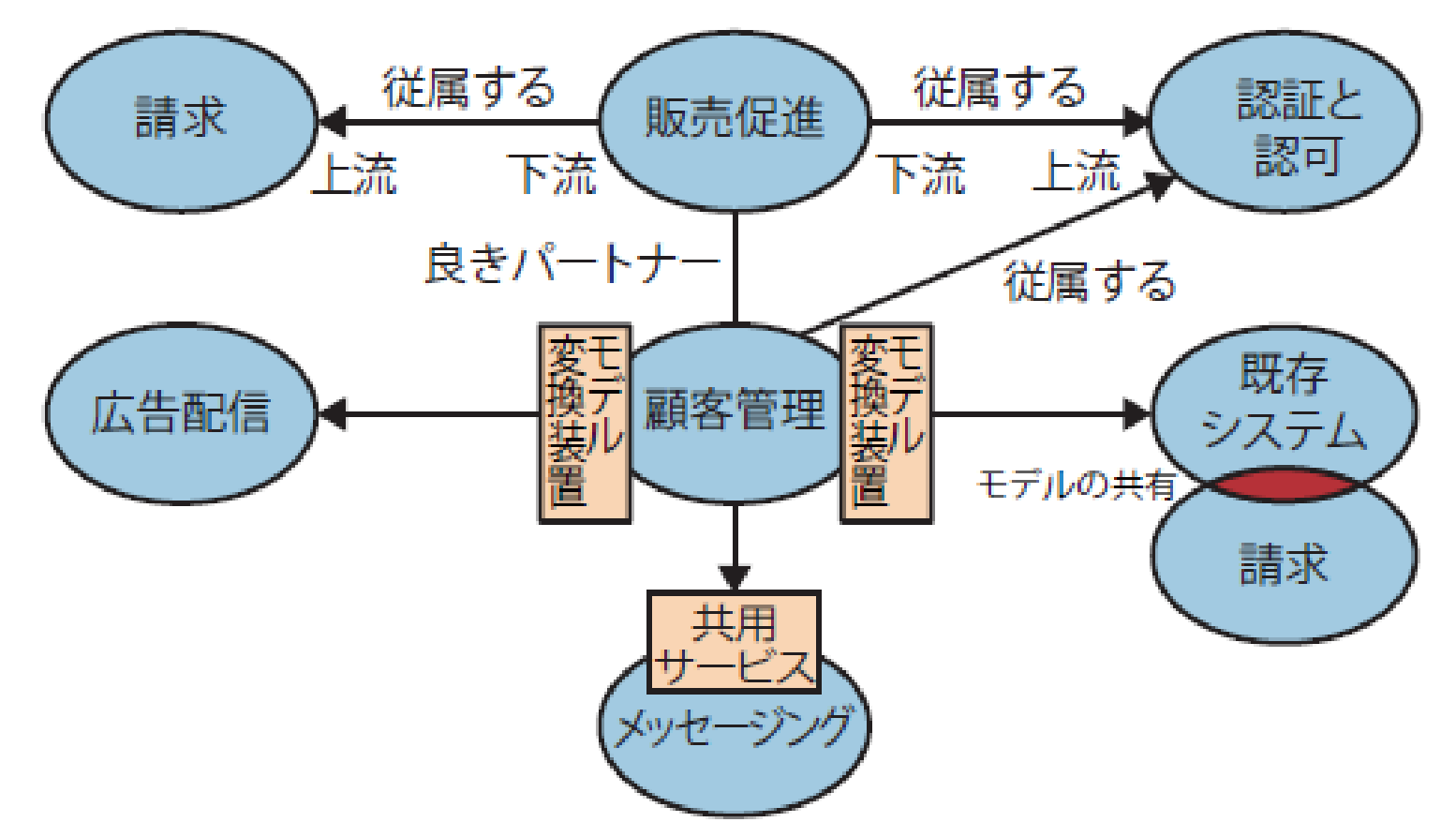
力関係が片寄った関係







文脈の地図（連係の全体像）



この地図からわかること

・顧客管理が中核の業務領域
・それ以外は一般または補完

中核モデルの独自性を保護するための設計判断

区切られた文脈どうしを連係する技術

（第９章）

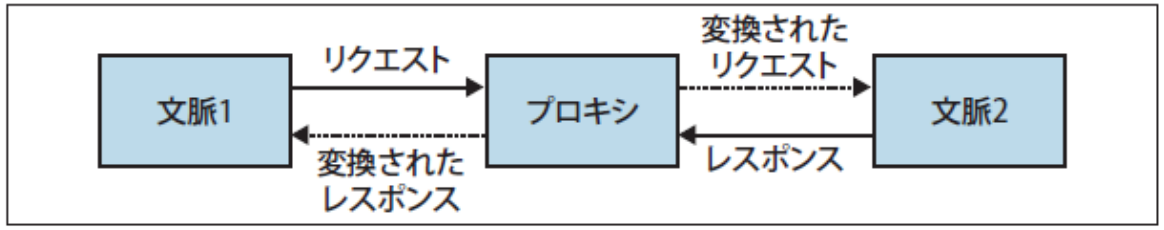


図9-2　同期通信

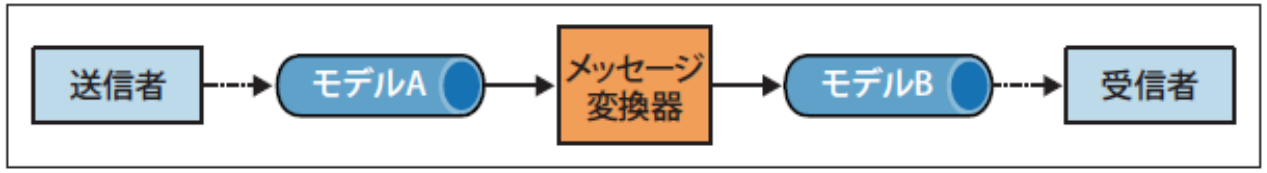


図9-5　非同期通信での変換

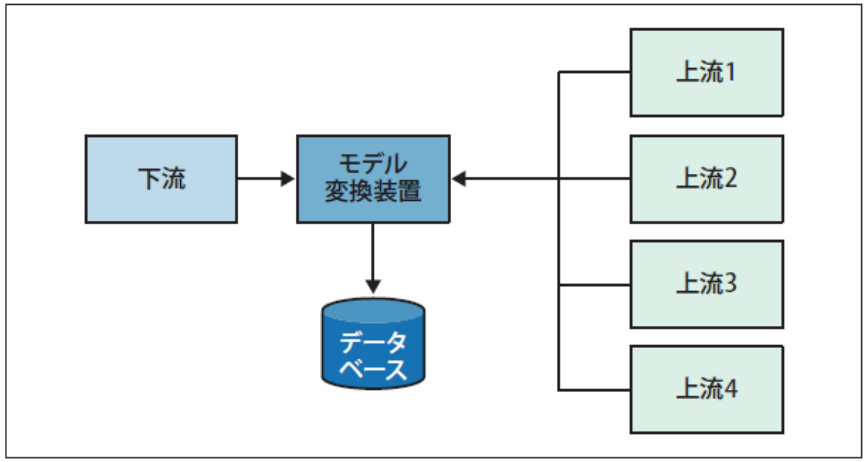
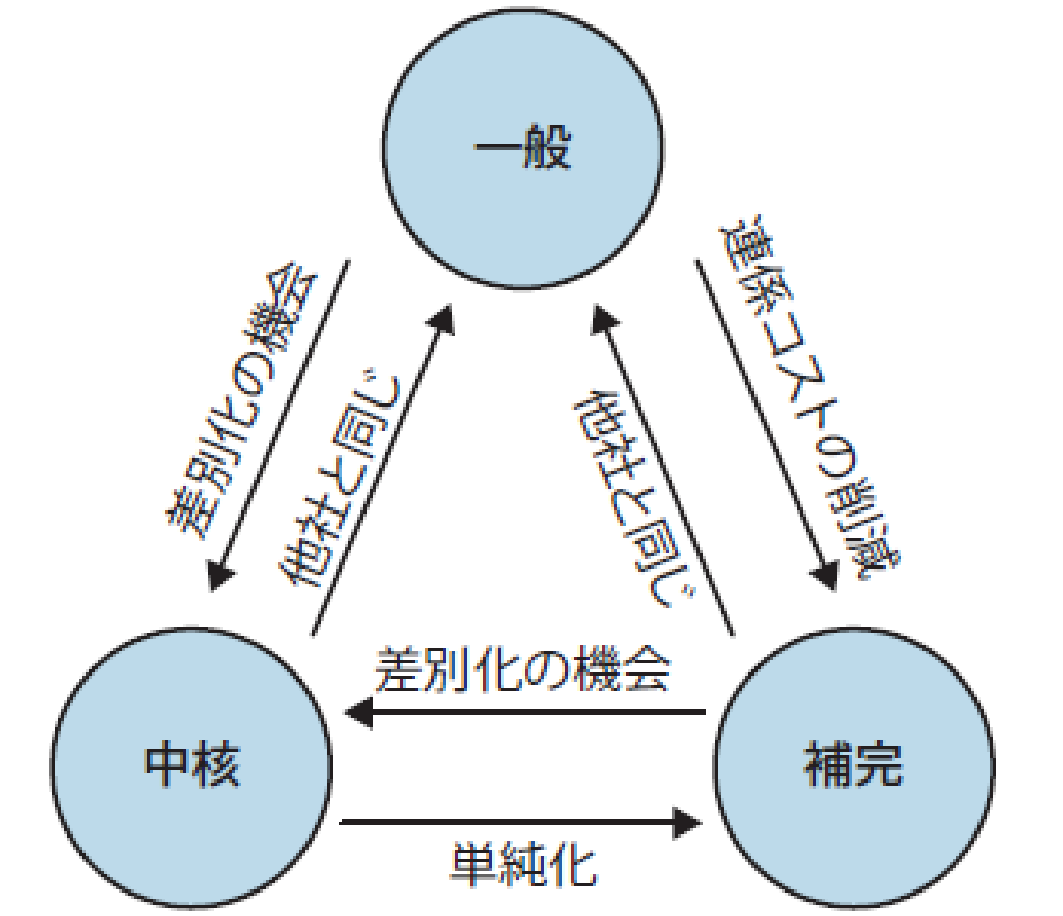


図9-10　モデル変換装置を使ってデータ連係を単純化する

事業の成長と
ソフトウェアの成長
（第１１章）

業務領域のカテゴリーの変化



事業が成長すれば、業務領域のカテゴリーは変化する

業務領域のカテゴリーが変化すれば設計方針が変化する

業務領域のカテゴリーと設計判断

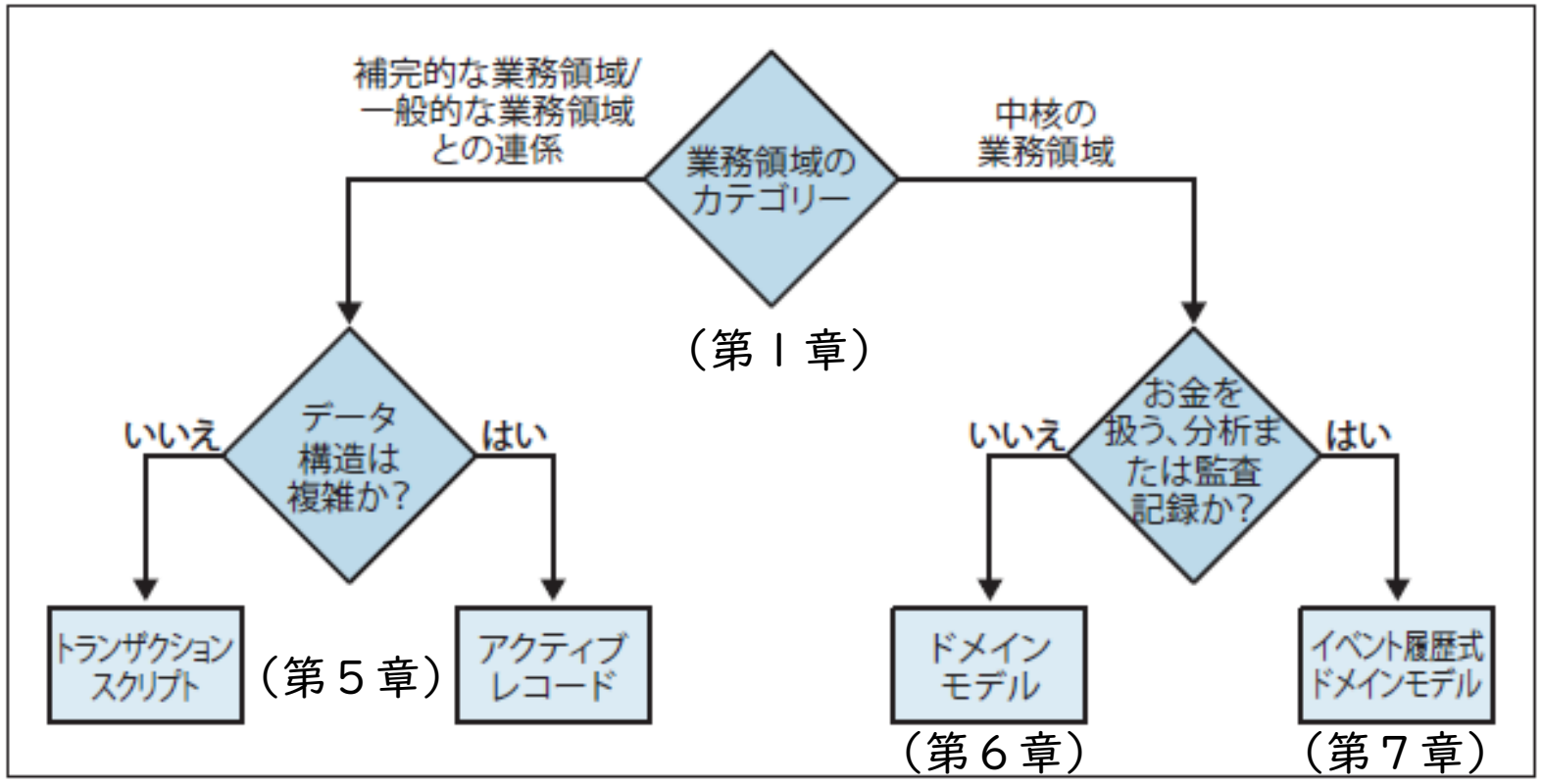


図10-3　業務ロジックの実装方法の判定方法

開発チームの学習と成長（付録A）

原著者の経験談（失敗談）
そこからの学び

現実世界のドメイン駆動設計

こういうことはありえないし、必要もない（第13章）

- チーム全員がドメイン駆動設計を熟知している
- 最初から全員が役に立つモデルの探求に全力をつくす
- 全ての関係者が同じ言葉を忠実に使う
- 既存システムや外部サービスを考慮しなくてよい制約の少ない新規案件

➢ **ドメイン駆動設計のすべての技法を使う必要はない**

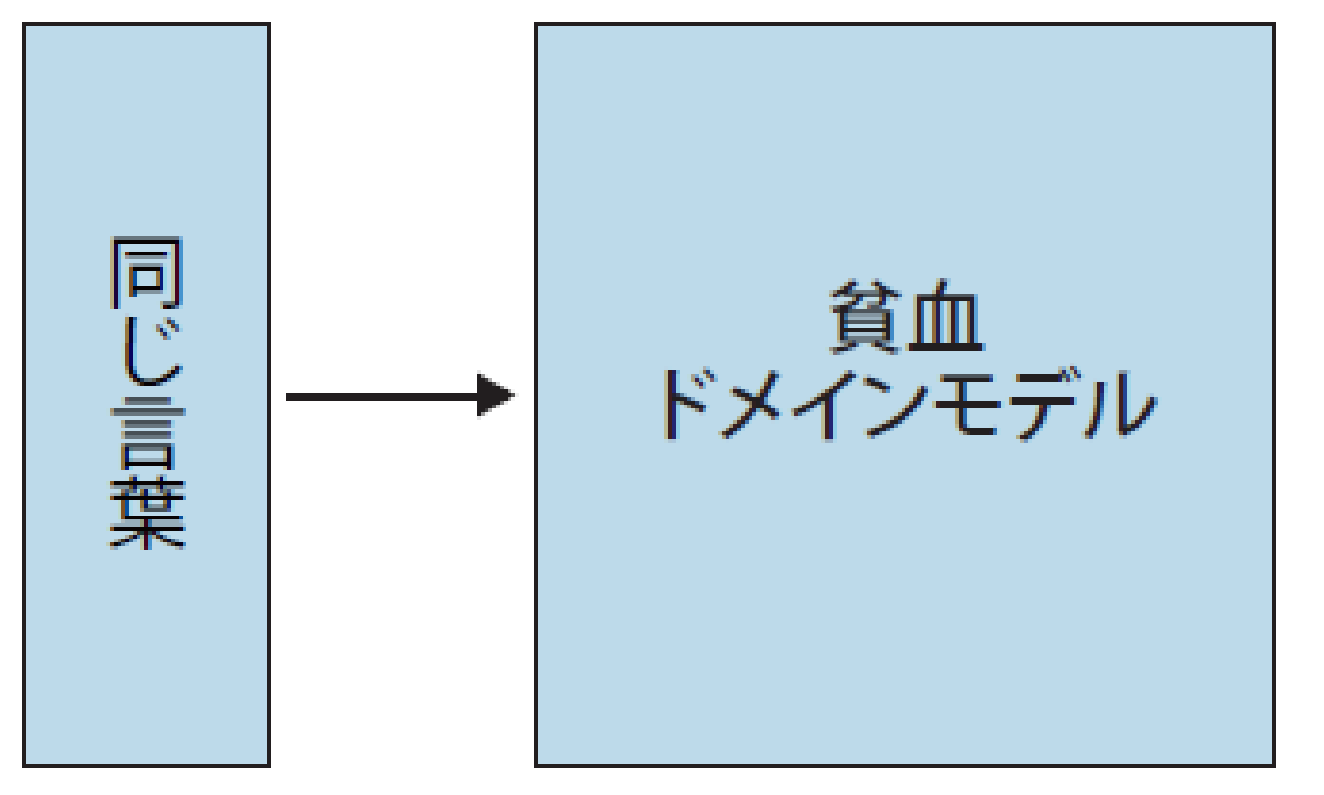
➢ **ドメイン駆動設計が組織として受け入れられていない状況でも実践は可能**

✓適切な道具を必要に応じて**選択的**に使う

✓それぞれのやり方の背景にある**考え方と原則**を意識して使う

✓組織の変化とソフトウェアの成長に**忍耐強く**取り組む

ドメイン駆動設計：最初の理解

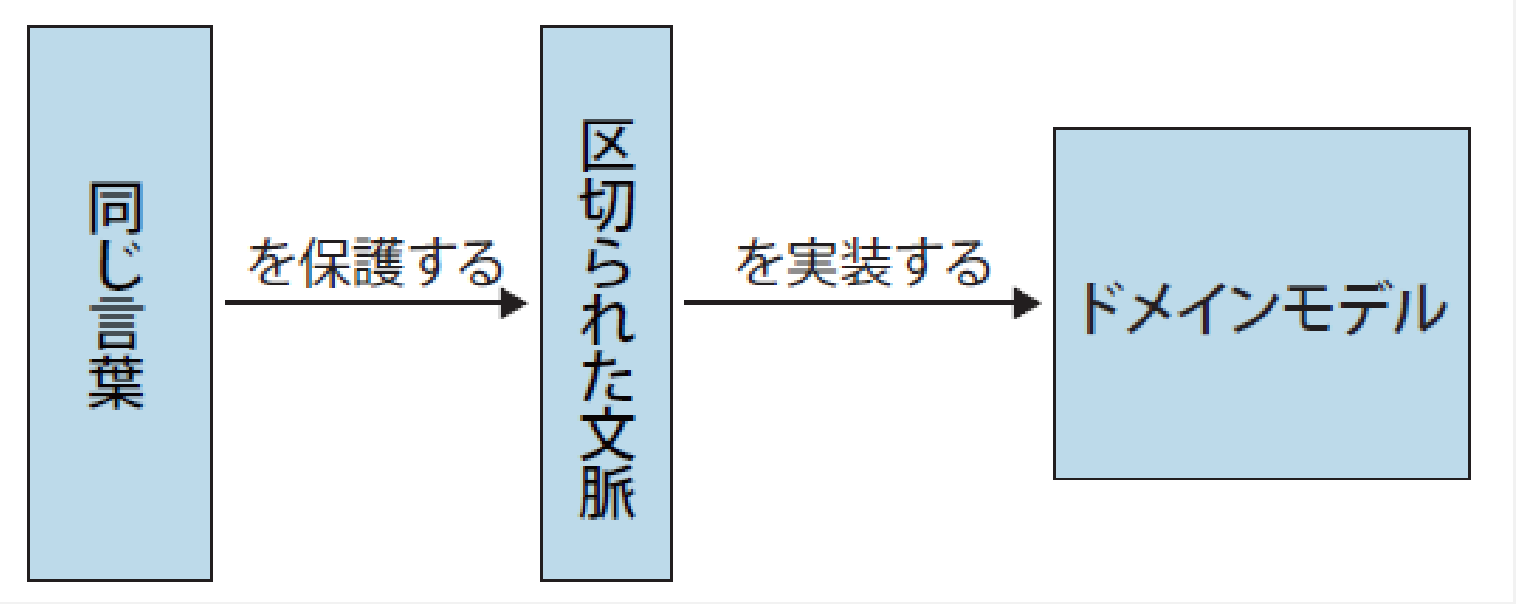


ドメイン駆動設計の表面的、かつ、
断片的な理解で取り組んだ段階

とにかく集約！

ドメイン駆動設計：失敗から学ぶ

モデルの複雑化→納期遅れ→誤った分業体制



同じ言葉の通用範囲の確立

同じ言葉の完全性の保護

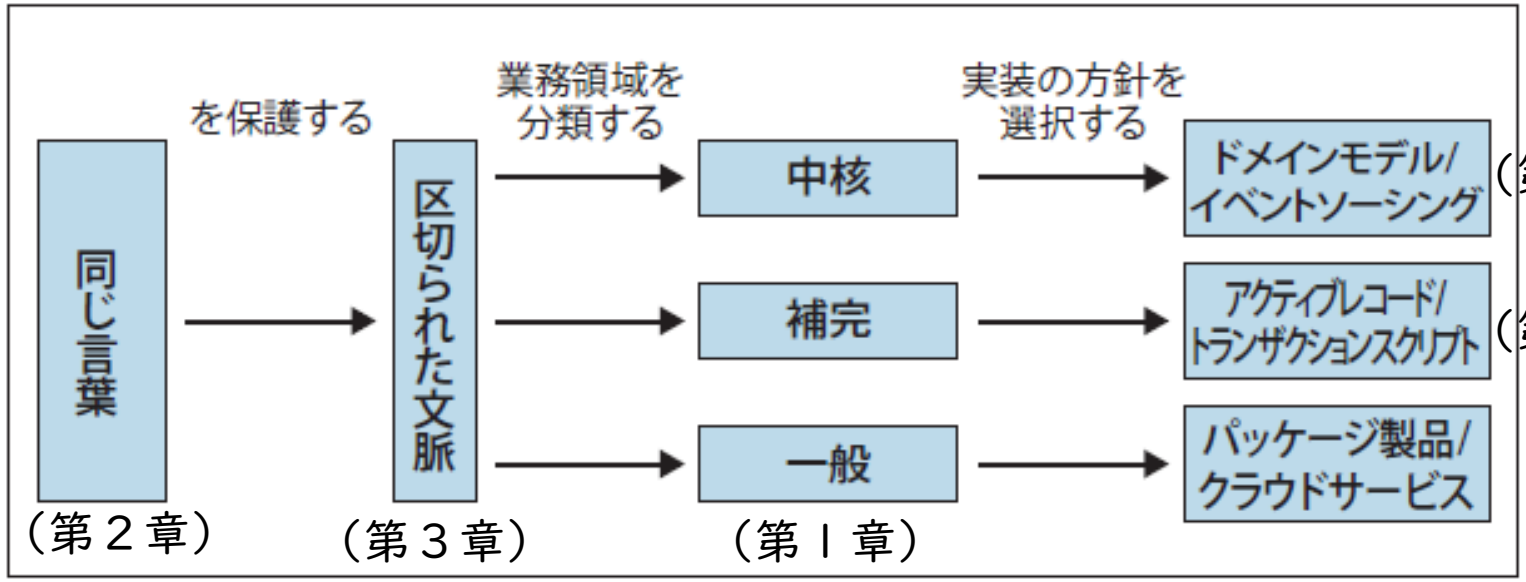
貧血ドメインモデルから

知識豊かなドメインモデルへ

完全に理解した！

ドメイン駆動設計：本格的な理解

事業戦略と整合した設計判断ができるようになった！



（第６章、第7章）

（第５章）

図A-6　ドメイン駆動設計の本格的な理解

ドメイン駆動設計と分散型アーキテクチャ（14章、15章、16章）

ドメイン駆動設計と分散型アーキテクチャ

マイクロサービスアーキテクチャ（第14章）

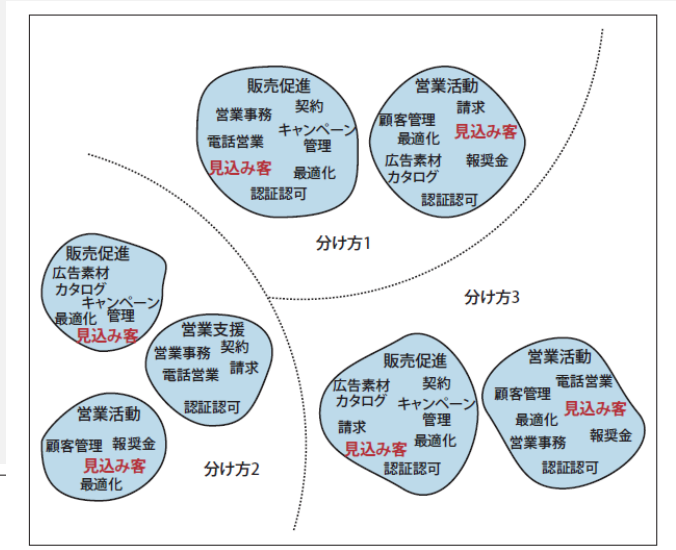


図14-9　区切られた文脈のさまざまな分け方

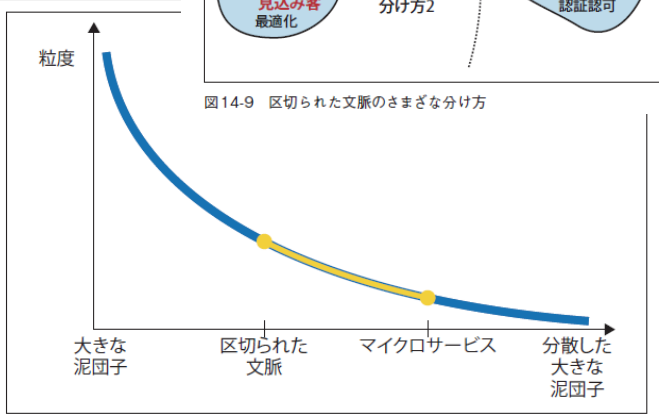
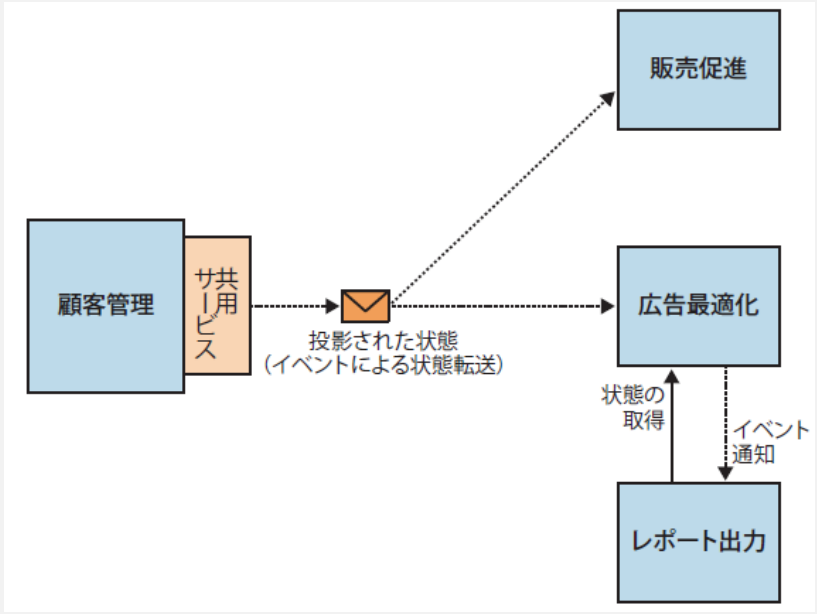
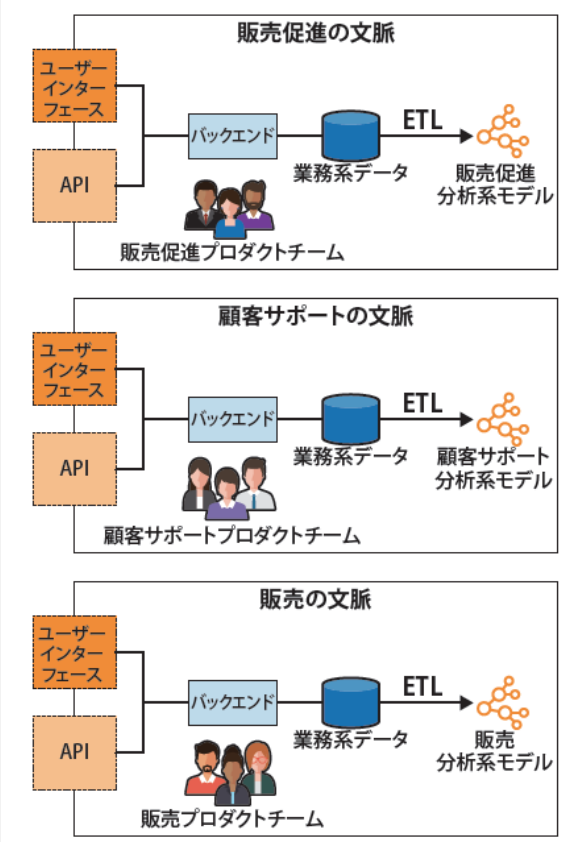


図14-10　粒度とモジュール性

イベント駆動型アーキテクチャ（第15章）



データメッシュ（第16章）



区切られた文脈の考え方で取り組む

まとめ

書いてあることは基本的には一つ

事業活動とソフトウェアを

一緒に発展させていくための方法

この本の主題であり、最初から最後まで一貫している

開発者が事業活動を理解して、その理解を設計判断に活かす