

見えないものに着目すると 上手くいく モデリングの勘所

2024/05/22(水)

合同会社DMM.com
ミノ駆動

プラットフォーム開発本部 第3開発部 DeveloperProductivityグループ

みなさんこんにちはー
ミノ駆動です

まずは
みんなで一緒に

#アーキテクチャ_findy
ハッシュタグ付きで

「設計なんもわからん」

と血涙を流しながら魂のシャウトを試みよう

せーの

設計なんもわからん

設計なんもわからああ
ああああああああああああ
ああああああああああああ
あああああああんなんな

自己紹介

ミノ駆動 (@MinoDriven)

合同会社DMM.com

プラットフォーム開発本部 第3開発部

DeveloperProductivityグループ

DMMプラットフォームの設計を改善し開
発生産性向上を図るのがミッション



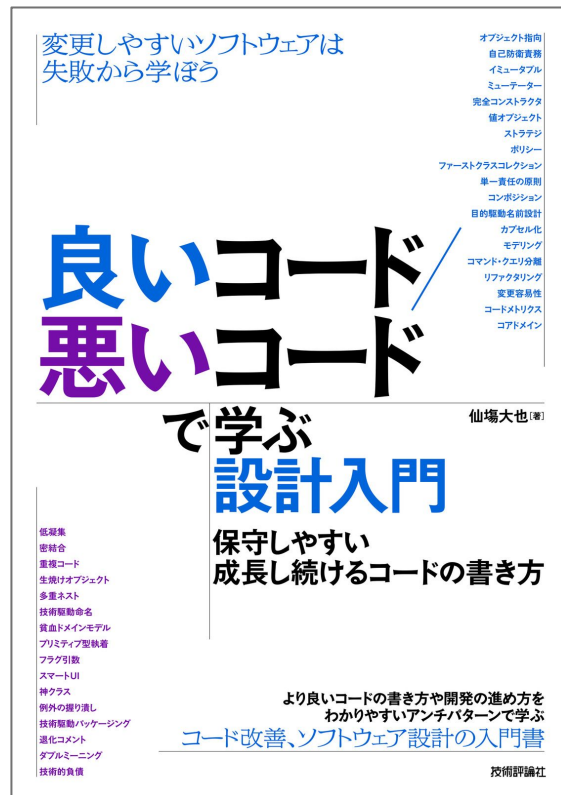
著書紹介

『良いコード／悪いコードで学ぶ設計入門』

変更容易性の高い設計を学ぶ、初級～中級
向け入門書

ITエンジニア本大賞2023技術書部門大賞
受賞

11刷重版



本セッションの理解目標

- 物理的に目に見えるものに囚われてモデリングすると、モデルの構造がイビツになりやすく、変更容易性が低下します。
- 目に見えないもの、認知困難なものが存在することを認め、目に見えないものに着目してモデリングすることで、変更容易性の高い構造を得られる。これが本セッションの理解目標です。

まずはありがちな
良くないパターンを見ていきます

商品

ID
商品名
説明
仕入単価
在庫数
安定在庫量
審査結果
審査所見
サイズ
重量
メーカー
取扱日時
予約開始日時
発売日
販売価格
電子書籍フラグ
...
.....

巨大な商品クラス

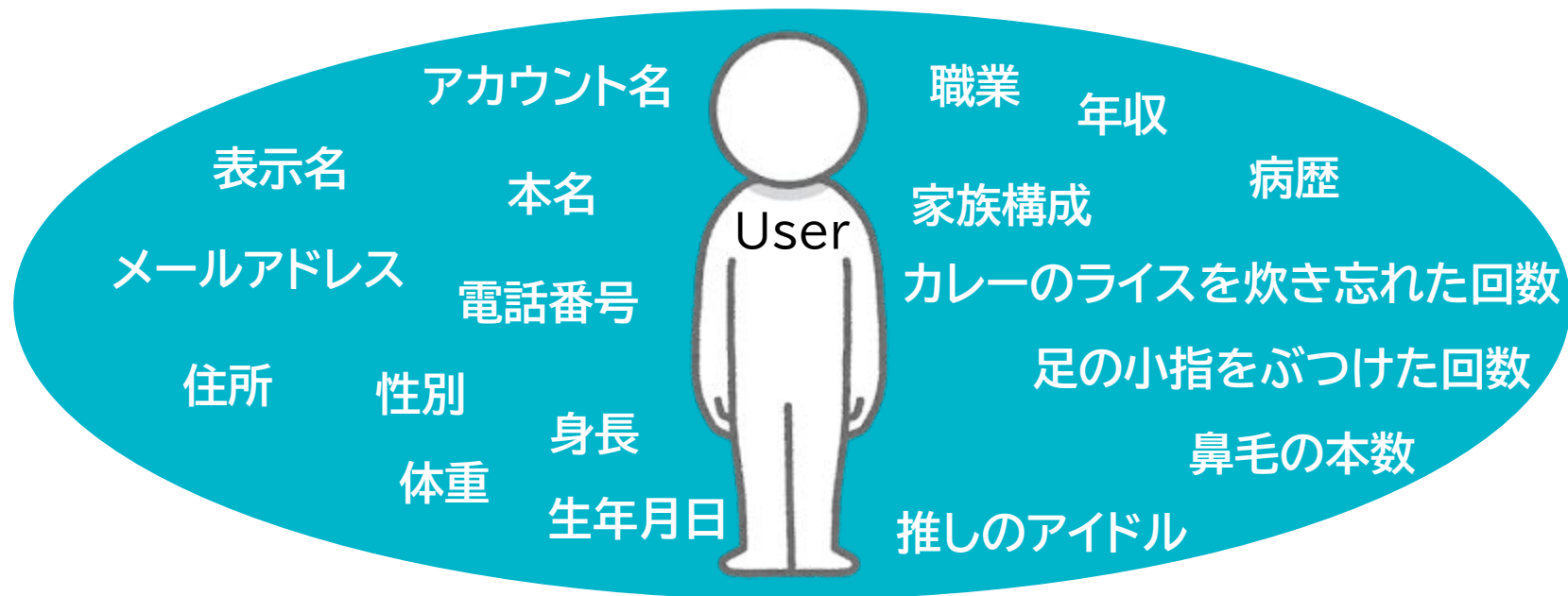
- 商品に関するありとあらゆるデータやロジックを持っている。
- 多くのさまざまなクラスと結合。
- 巨大かつ複雑で、保守や変更のコストがとにかく高くつく。
- 制約どうしがバッティングして不具合になることも多く、不具合回避のためのフラグや条件分岐が追加され、輪をかけて複雑化。
- 典型的かつ代表的な技術的負債。どこでも見かける。
- 商品だけでなく、サービスの「大通り」にいるクラスはたいていほとんどが巨大化する(例:Userクラス)

ちゃんとモデリングしたつもりなのに
上手くいかない…

結局分割すればいいんだろうけど
どう分割すればいいかわからない…

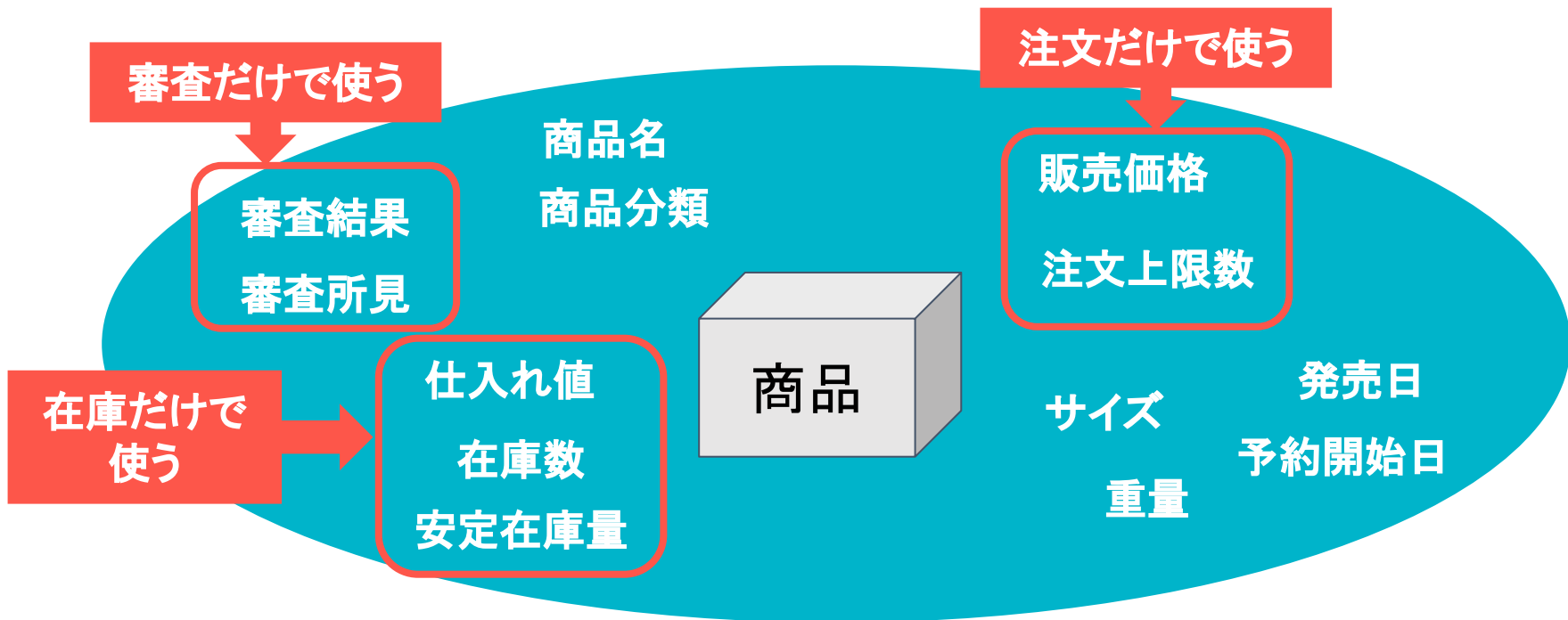
なぜ上手くいかないのか
私はこう考えます

物理的存在に
とらわれているから



商品でも人間でも、挙げたらきりが無いほど多くの属性を持っています。したがって物理的単位をモデリングの境界にすると、ありとあらゆる属性が詰め込まれてしまいます。

物理的単位とは別の見方で
モデリングする必要があります



必要とする属性は状況により異なります

「状況により異なる」とは
一体どういうことか

私はこう考えます

目的が異なる

目的が異なると解決が必要な問題が異なる

問題解決に必要な手段が異なる

目的	問題	解決手段
		
		

システムは目的達成手段

目的	手段
目的地にはやく移動したい	自動車、飛行機
食べ物を温めたい	電子レンジ
商品を売買したい	ECサイト
遠くの人と意思疎通したい	電話、SNS

手段たるシステムは、目的達成上の問題やその解決に必要な技術が、それぞれ異なります。

モデルも
「目的を達成するためのミクロなシステム」
と考えることができます

審査目的の
「審査品目」として
モデリング

注文目的の
「注文品」として
モデリング

審査結果
審査所見

商品名
商品分類

販売価格
注文上限数

在庫目的の
「在庫品」として
モデリング

仕入れ値
在庫数
安定在庫量

商品

サイズ
重量
発売日
予約開始日

目的特化のモデリングにより目的外の要素が混ざらなくなる
構造がシンプルになる



逆に巨大な一枚岩モデルは
目的が異なる家電が悪魔合体したような存在と言えます
目的単位で分離隔離しましょう

ここまでは
私のこれまでの登壇で
よく話していた内容

ところでなぜ物理的存在に
とらわれてしまうのだろう

私はこう考えます

人間は入力情報のうち9割を
視覚情報に頼っているから

アンコンシャスバイアス

- 無意識の思い込みや偏見のこと。
- 自分が知っている属性や枠組み、パターンに当てはめて考える脳の機能。
- この脳機能のおかげで人間は高速に思考できる(だから必ずしも悪いことではない)。
- しかし当てはめる枠組みが不適格だと判断が誤ってしまう。
- 人間は視覚情報に頼りすぎているため、どうしても物理的存在に引っ張られた思考に陥りがち。



例えば街を歩いていて
すれ違う人それぞれが何を考えているのか
あなたは見た目で見えますか？



ほぼ100%分らないですよ
目的は何か、何に悩んでいるのかは
見た目じゃわかりません
聞いてみないとわからないですね

ソフトウェアで達成したい目的や
それに付随する問題はもっと複雑で
見えない、わからない、認知困難

だからこそユースケース分析や
イベントストーミングといった分析により
見えないものを顕在化する活動が
設計上重要なのです

一旦ここまでのまとめ

- 目に見える物理的存在の単位でモデリングすると、あらゆる付帯要素が詰め込まれて巨大化複雑化する
- 目的が異なると解決を要する問題や解決手段が違ってくる
- システムは目的達成手段であり、モデルはミクロなシステム
- 目的単位でモデリングすると目的外の要素が混入せず、構造がシンプルになる
- 人間はほとんどを視覚情報に頼っているために、物理的存在に引っ張られた思考に陥りがち
- ソフトウェア開発で必要なのは目的やそれに付随する問題といった目に見えないものばかり、だからこそ分析し顕在化する活動が重要

ここから先は
前述の考え方にもとづき
ミノ駆動が実践している
「見えないものに着目する設計」を紹介します

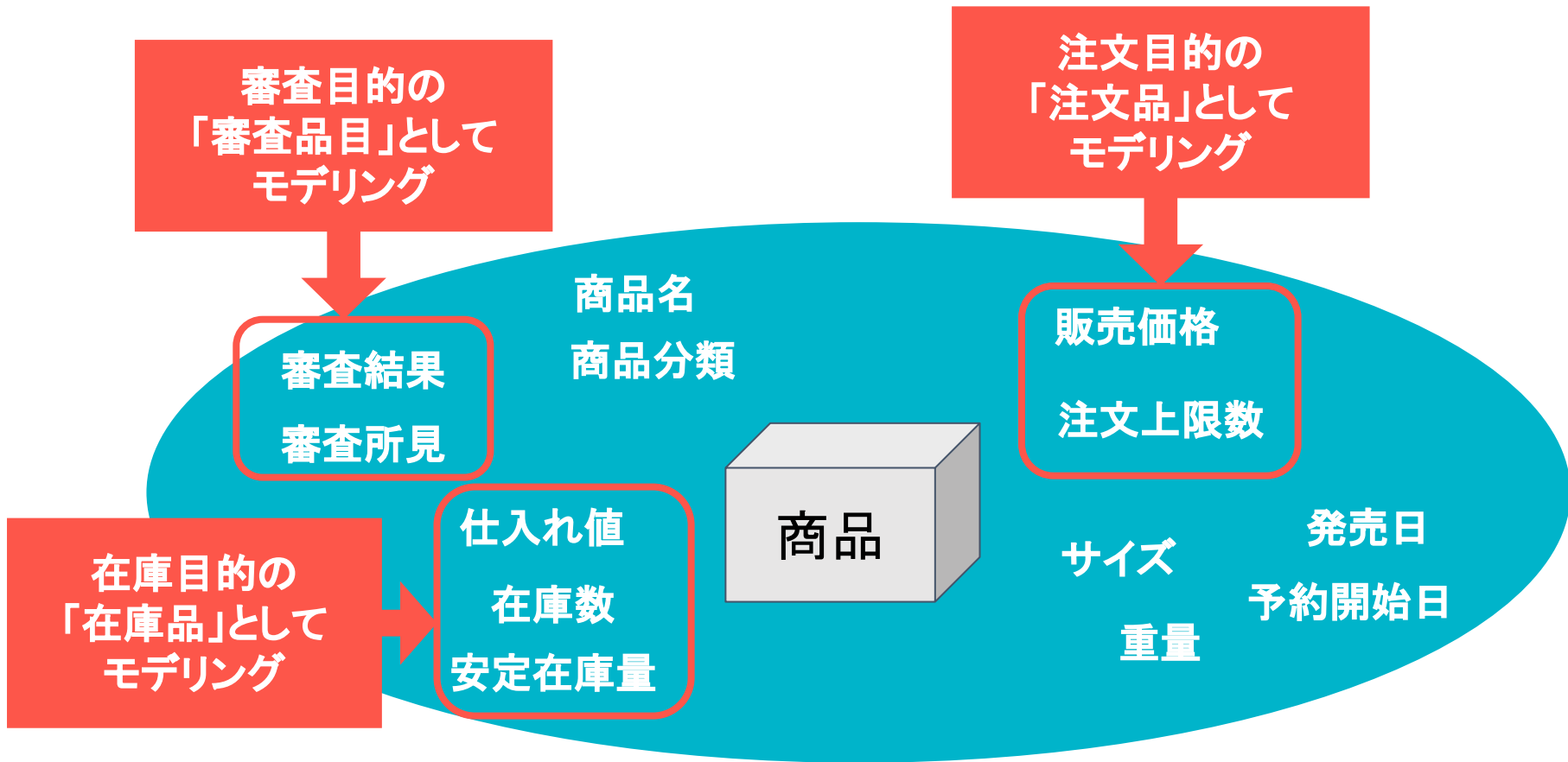
【実践ノウハウ1:超基本】 目的ベースの抽象化

抽象化の定義

対象物に付随する様々な特徴のうち、ある目的に合致した特徴のみを抜き出すこと。

(『「具体⇔抽象」トレーニング 思考力が飛躍的にアップする 29問』著:細谷功, PHPビジネス新書, p.97)

抽象化の定義にはいろいろありますが、システムは目的達成手段なので、モデリングなどではこの定義に則った抽象化を考えた基本としています



先程の商品の目的特化のモデリングでも
それぞれ目的に沿った要素のみを抽出していますね

【実践ノウハウ2】
物理的存在ではない
目的特化の概念として解釈する

「商品の売買」とは
いったいどういうこと？

どのような解釈かご存知ですか？

「商品の売買」とは所有権の移転

- 「商品の売買」とは、商法では売買契約の締結であり、契約締結をもって商品の所有権が移転します。
- 所有権に伴うルールを把握した上で売買機能を実装することが肝要です。
- 例えば「いつ」「誰から誰に」所有権が移転したかをシステム上で記録することで、売買や配送で万が一なんらかのトラブルが発生した際、法的に問題を解決しやすくなると考えられます。

書籍ドメイン駆動設計の例

貨物の物理的移動は、その貨物に対する法的責任の移動に主役の座を譲った。「船荷証券(bill of lading)」のような、一見すると明らかではないオブジェクトが前面に来たのだ。

(『エリック・エヴァンスのドメイン駆動設計』 著:エリック・エヴァンス, 翔泳社, p.192)

DDDの貨物輸送アプリケーションの例でも、船舶輸送用コンテナといった物理的存在ではなく、船荷証券といった、輸送の目的について回る権利や責任を表現するモデルが重要になった旨の記述があります。

目的によって解釈は異なる

目的	商品の解釈	ついて回る概念
売買	所有権	売買成立日時、所有権の移転先、支払方法
配送	貨物	荷送人、配送先、到着予定日時、追跡ID
会計	棚卸資産	仕入単価、在庫評価、棚卸評価損

商品ひとつとっても、目的それぞれで解釈やついて回る概念が大幅に異なります。目的ごとの解釈で概念を整理し、目的に特化したモデルをそれぞれ設計する考えが重要です。

目を閉じ心眼を開け

- 物理的存在はモデリング上本当に厄介で、私でも意識していないと思
考が引っ張られてしまいます。それほど強力なバイアスが働きます。
- 目を閉じ、実態が何であるのか心の眼で見破る姿勢が重要です。
- 「お前は○○以外の何かだな」「目的は何だ？一体何をしようとしてい
る？」「正体は何だ？」「本当の姿は何だ？」と問いかけましょう。

【実践ノウハウ3】
分析活動では
物理的に目に見えないものや
知識がないと認知困難な概念に着目する

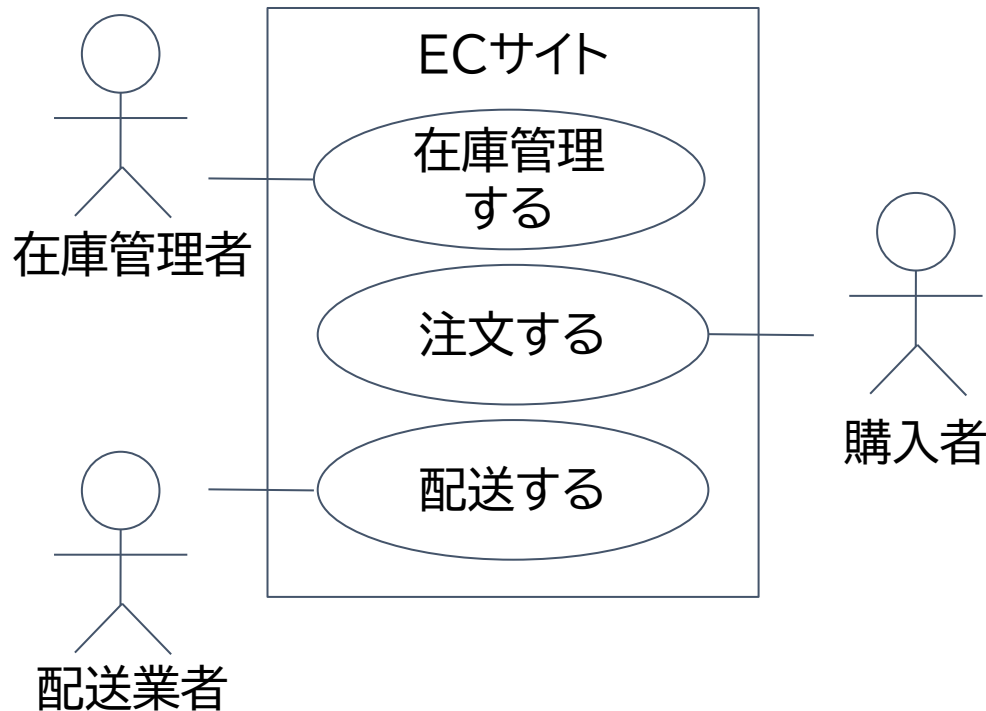
目に見えないものを見える化する分析活動

- ユースケース分析やイベントストーリーミング、ドメインエキスパートとの議論など、モデリングの役に立つさまざまな分析活動があります。
- ただ漫然と形式に則って分析や話し合いをするのではなく、目的などの目に見えないもの、知識がないと認知困難な概念を顕在化ことを意識しましょう。
- 特に、例として前述した「所有権」「棚卸資産」は、専門知識がないとそもそも認識できません。ドメインエキスパートから専門知識を引き出す、関連する文献に目を通して意味を理解する姿勢が大事です。

【実践ノウハウ4】
ユースケース図のアクターから
目的を洗い出す

目的は人が決めている

- 「～したい」という目的は天から降ってくるものではありません。
- 目的は人が決めています。
- ユースケース図ではシステムに参与するアクターを図式化します。
- どんなアクターがいるか、それぞれがどんな目的を持っているのか分析するのが良いでしょう。



【実践ノウハウ5】 些細な目的の違いを見破る

- 例えば注文には、通常の注文の他、予約注文やサブスク注文、抽選注文、法人注文などさまざまな種類があります。
- 良くない設計では、これらをたったひとつの注文モデルにまとめ、フラグで種類を表現する実装になりがちです。
- フラグでの制御切り替えを必要とするため、ロジックが非常に複雑になります。
- 「予約したい人」「サブスク注文したい人」など、アクターの目的はそれぞれ違います。目的の違いを見破りましょう。
- 「予約注文」「サブスク注文」など、目的ごとのモデルに分離隔離しましょう。

注文

ID
注文金額
注文日時
注文数
予約注文フラグ
予約順
サブスク注文フラグ
抽選注文フラグ
当選フラグ
オークション注文フラグ
落札フラグ
法人注文フラグ
.....
...

【実践ノウハウ6】 異常系ファーストで検討する



道路交通法は何のために存在するのでしょうか？

道路交通法 第一条

この法律は、道路における危険を防止し、その他交通の安全と円滑を図ることを目的とする。



交通事故といった異常系を防ぐための
法律(ルール)と言えますね

ところが

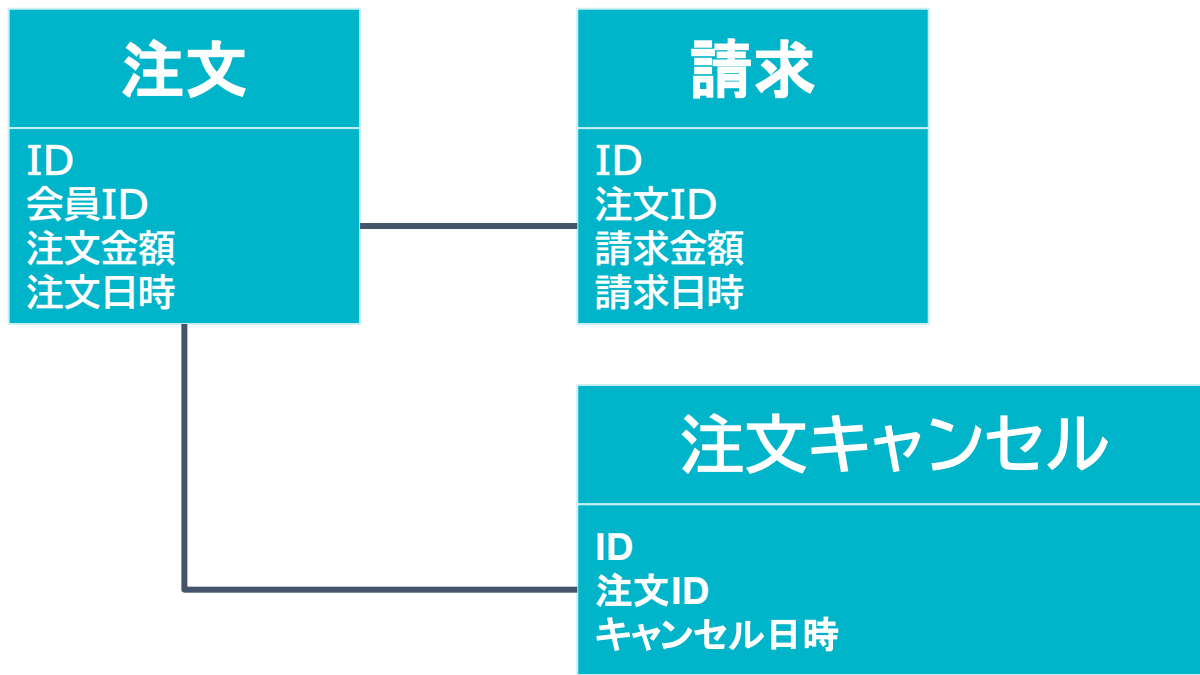
異常系は検討の後手に回りがち

- 機能仕様は正常系ばかりが検討され、異常系は後手に回りがち。
- リリース後に障害発生し、そのとき異常系検討の抜け漏れが判明することがよくあります。
- さらに悪いことに、急いで障害対応するため、付け焼き刃的に粗悪なコードが実装されがちです。変更容易性が低下する主要因のひとつといっても過言ではありません。
- 「とりあえず動くものを早く作る」の精神で開発すると「起こってほしくないこと」の想像が困難になります。こうした点から、異常系は認知困難で「見えない概念」とも考えられます。

異常系は問題解決ルール宝库

- ところでなぜ我々は条件分岐を実装するのでしょうか？それは正しく目的を達成するためです。異常を起こさず正しく達成するためのルールが条件分岐として実装されるのです。
- 「どうやったらシステムをバグらせられるか」「どうやったら目的が達成できなくなるか」といった異常系を検討すると、異常を起こさないためのより良い解決ルールが見つかります。
- そうしたルールをモデルに組み込むことで、不具合に強く、変更容易性の高い構造となります。
- 注文キャンセルなどのキャンセル操作はさまざまな巻き戻しが生じるため、ロジックが非常に複雑になります。異常系検討の格好のネタです。

【実践ノウハウ7】 出来事をモデリングする



ひとつの出来事はひとつのモデルとして設計しましょう。「注文してないのに注文キャンセルできてしまう」といった歴史の前後関係が矛盾する類の不具合を防止できます。

(詳しくは「イミュータブルデータモデリング」「T字型ER法」)

【実践ノウハウ8】 関係性をモデリングする



「実は彼らは共犯関係だった」のように、関係性も見えない、認知困難な概念です。

良くない設計では社員モデルがnull許容な部門IDを持っているなどイビツな構造です。

上記「部門所属モデル」のように関係性を交差エンティティとして設計することで、null許容カラムを実装せずに済む、複数の部門への所属を表現できるといったメリットを得られます。

(これも詳しくは「イミュータブルデータモデリング」「T字型ER法」を参照)

【おまけ】 犬猫クラスが勘違いを生む

【ダメ】犬う、猫おおおあああ！！【絶対】

- プログラミング入門書や入門記事で頻繁に登場するクリーチャー。
- 「よくわかんないけど、クラスは物理的な存在を実装するものなんだな」と初学者に先入観を植え付けてしまう、序盤のステージにおける超の付く強敵。
- もし犬猫を持ち出すにしても、まずはサービスの目的を明らかにして(MUSTですウ！！)、その目的に特化した概念として抽象化すべき。たとえばペット販売サービスならペットモデル、血統書管理サービスなら血統書モデル、など。
- 物理的存在をモデリングするという悪しき手法から我々は脱却しなければならない。それは初学者が学ぶ際も同様である。

以上が発表内容です

採用募集



©大久保篤・講談社／特殊消防隊動画広報課

AIから地方創生まで、
16の領域で60以上の事業を運営。

なんでも やってる DMM

規模の大小やジャンルに関係なく
未来を感じるビジネスに投資。
ありとあらゆる領域に参入、挑戦し続ける。

一緒にDMM.comで働いてみたい方は
こちらからどうぞ！！

採用案内資料

<https://speakerdeck.com/pf-businessdivision/platform-business-division-recruitment>

エントリー

<https://dmm-corp.com/recruit/engineer/822/>

Findy上の募集ページ

<https://findy-code.io/companies/177/jobs/giPkGMm76gnBa>

ご清聴ありがとうございました