第 32 年度
ソフトウェア品質管理（SQiP）研究会
演習コース II
形式手法と仕様記述

酒井 雄太（キヤノンアイテック株式会社）
宮本 陽子（株式会社メタテクノ）

2017 年 2 月 24 日

演習コースII
形式手法と仕様記述で

宮本は,

2011年:「良い仕様書」を作成するための規格,手法
（IEEE 830, USDM, VDM）の関係性をまとめる

2012年:VDM とUSDM を組み合わせた仕様記述方法を提案

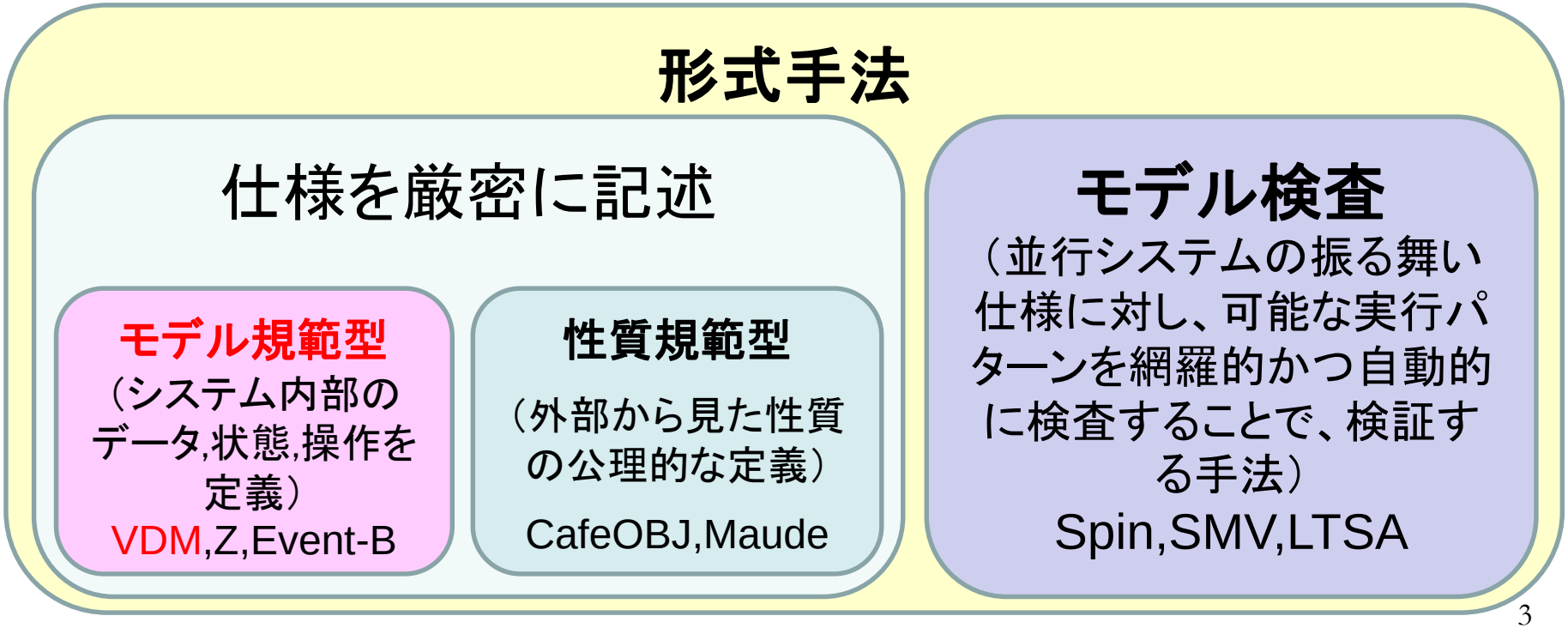
2013年:機能仕様書に記述されていない暗黙の仕様を考え,
書くためのステップ・質問パターンを紹介

2014年:仕様の「認識のずれ」「曖昧さ」を無くすために,
自然言語・（業務フローなどの）図・VDMの組合せを提案

2016年:文章の記述力強化のために，文章作成時に思考の整
理と論理構成の検討を促すためのフレームワークの開発

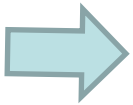
形式手法（Formal Methods）とは

システム開発，特にソフトウェア開発の手法で，数理論理学に基づく科学的な裏付けを持つ仕様記述言語を用いて設計対象の機能や性質を表現することにより，ある側面の仕様を厳密に記述し，開発工程で利用する手段の総称



VDM(Vienna Development Method)

- 仕様記述に特化した言語をもちいることで，仕様を厳密に記述できる．自然言語の仕様書に含まれる曖昧さを取り除くことができる
- （用語の統一，構文チェック，型チェックなどを）ツールがサポートしてくれる
- 仕様を動作させることや，テストができるため，開発者や評価者になじみやすい
- 日本語のツール，書籍，Web情報が，他の形式手法に比べ多い．
- オープンソースのツールが公開されている
- 文書やプログラムの記述力や論理的な思考力の向上に役立つ



VDMは形式手法の入門としておすすめ

誰でも休日が好きだ(VDM記述例)

形式仕様記述言語で書くと…

SEC Software Reliability Enhancement Center

- 誰でも全ての休日が好きだ
  - forall p in set 全員 & (forall d in 全休日 & 好き(p, d))
- 皆が好きな休日が少なくとも一つある
  - exists d in 全休日 & (forall p in set 全員 & 好き(p, d))
- 皆が好きな休日が丁度一つだけある
  - exists1 d in 全休日 & (forall p in set 全員 & 好き(p, d))
- 誰でも自分が好きな休日が少なくとも一つある
  - forall p in set 全員 & (exists d in 全休日 & 好き(p, d))
- 誰でも丁度一つだけ好きな休日がある
  - forall p in set 全員 & (exists1 d in 全休日 & 好き(p, d))

形式手法を利用することで

- 仕様や設計を「きちんと」書こうとする過程により
  - 曖昧さを解消することになる
  - 不正確さ，不整合も表面化する
  - システムに対する理解が深まる

- 仕様や設計を「きちんと」書いた結果により
  - 誤解なく情報を共有できる
  - 科学的・系統的な分析・検証ができる
  - 分析・検証をツールに支援させることができる

形式手法，形式仕様記述は

- 私たちソフトウェア開発技術者にとって，
  形式手法の知識を学び，かつ，
  実践しようとする（仕事に活かそうとする）こと
  は，あらためて「気づき」を促す．
  - 厳密な仕様記述の重要性（今までいかに曖昧なままで開発していたか）
  - システムや仕様についての，より深い理解
  - 発注者と開発者だけでなく，開発者間の認識の違い

さまざまな視点に合わせた仕様書の作成・維持の支援手法

演習コースⅡ 形式手法と仕様記述

2017/2/24

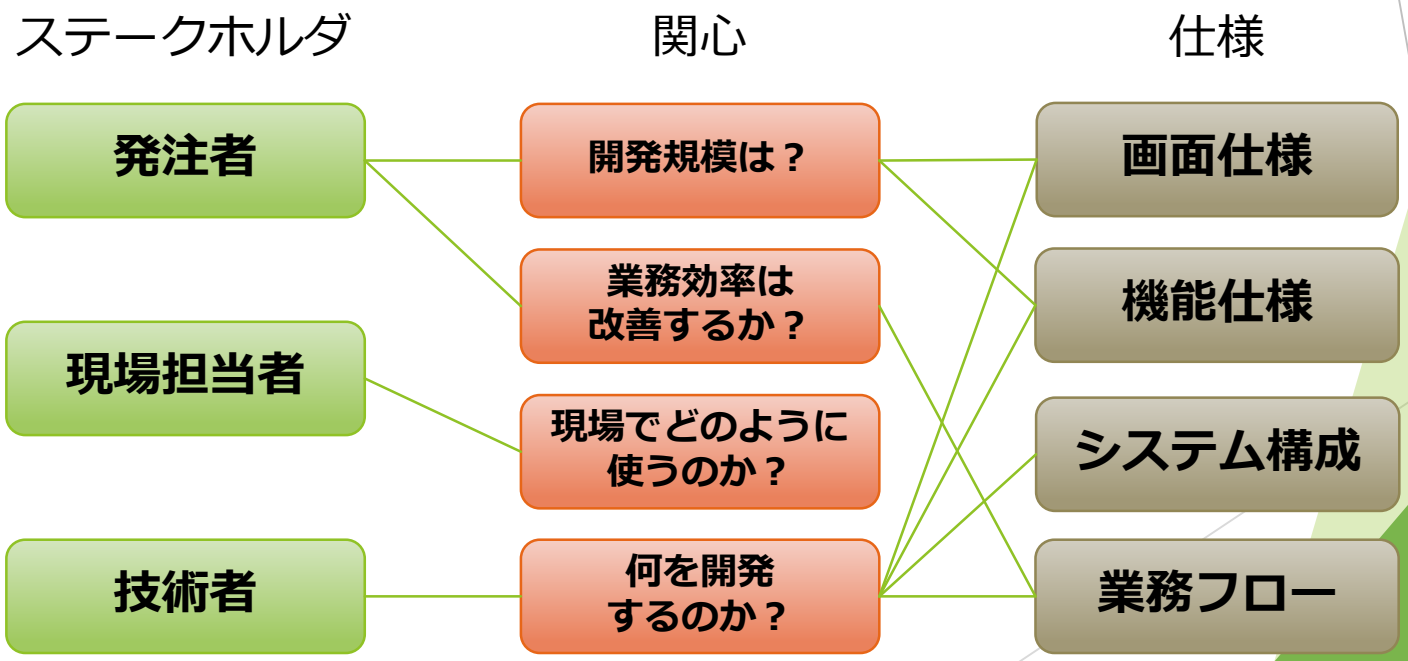
キヤノンアイテック株式会社　酒井 雄太

概要

- 仕様書の課題
  - 仕様書の記述があいまいである
  - ステークホルダ毎に関心事は異なる
  - 多くの説明資料を用意すると不整合が生じる
- 課題解決のアイディア
  - 一つの厳密な仕様書から，さまざまな視点の仕様書を自動的に生成する
- アイディアの検証
  - 一つの仕様書・さまざまな仕様書の自動生成
  - 一定の効果と拡張性の感触を得た

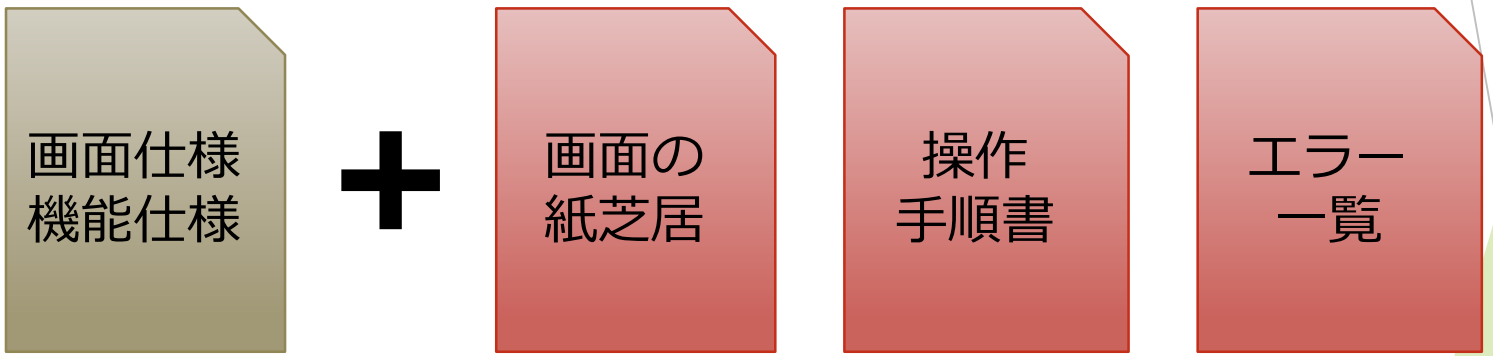
ステークホルダ毎に関心事は異なる

- ステークホルダ間のコミュニケーションのために仕様書が用いられるが・・・



多くの説明資料を用意すると不整合が生じる

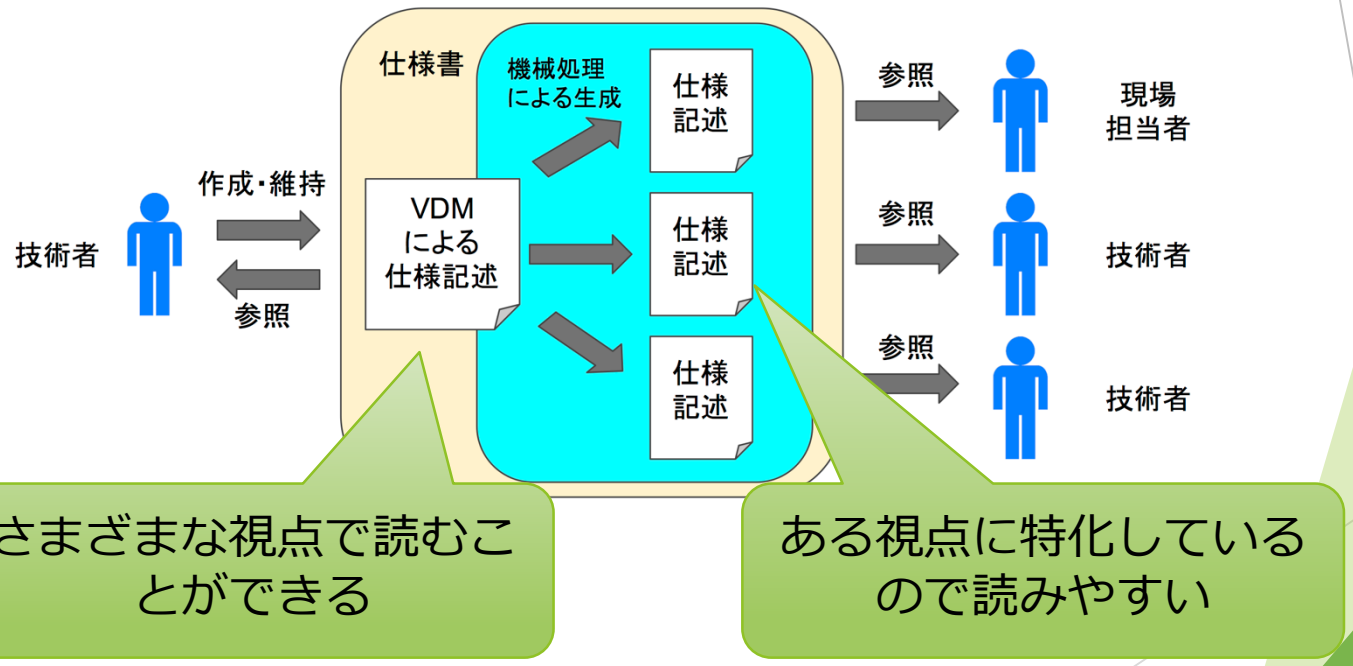
- 読み手に合わせた説明資料を用意することが多い



曖昧な仕様をもとに作成された資料は
不整合が生じる

さまざまな視点の仕様書を自動的に生成する

- 一つの仕様書を作成・維持するだけで，複数の仕様書を生成する



<u>さまざまな視点に合わせた仕様書の作成・維持の支援手法</u>

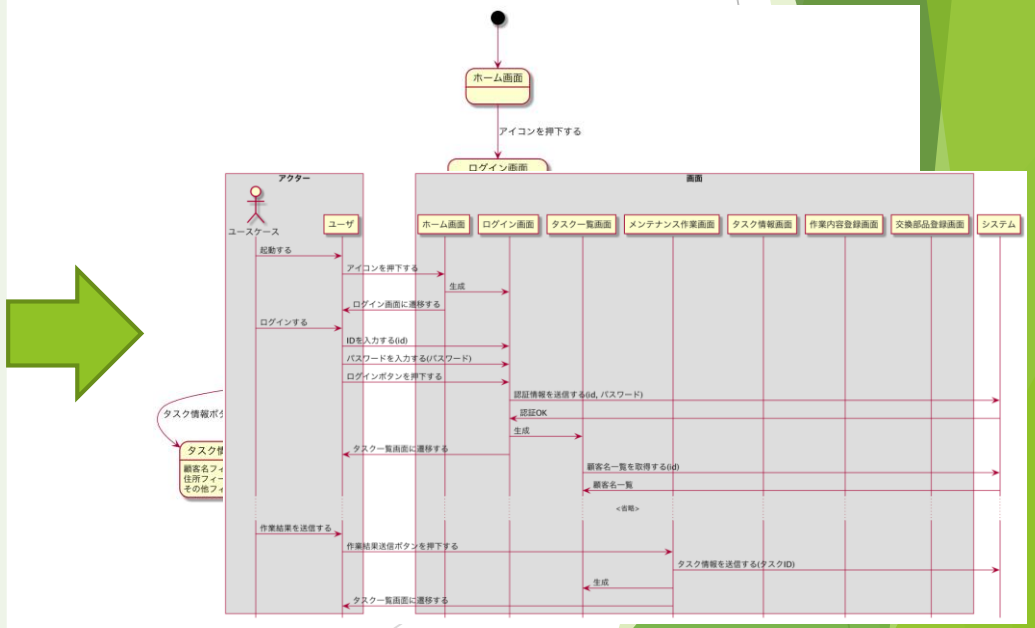
VDMによる仕様記述の方法

- VDMによる仕様書は，視点の異なる仕様書を生成することを狙った書き方をする
  - 各画面を1つのVDMクラスとして記述することで，画面毎の仕様をまとまった形で抽出しやすくする
  - 画面に対する一連の操作の具体例を記述することで，基本的な使い方の視点の仕様を生成できるようにする
  - 全ての操作に対して，ログ出力を記述することで，内部の振る舞いについての仕様を生成できるようにする

自動的な仕様書の生成

- 画面遷移図(画面仕様の全体像を理解する)
- シーケンス図(基本的な使い方を理解する)

```
-- このクラスは ログイン画面 についての仕様記述である
class ログイン画面 is subclass of 画面
instance variables
-- UI要素をインスタンス変数として表現する
public IDフィールド : token := mk_token("NULL");
public パスワードフィールド : token := mk_token("NULL");
  <省略>
--画面遷移を伴う操作の戻り値は 次の画面のインスタンスである
public ログインボタンを押下する : () ==> 画面
ログインボタンを押下する() ==
  (
    --操作に対して，VDM++を実行した際に出力されるログが埋め込まれている
    io.print("ログイン画面.ログインボタンを押下する¥n");
    let
      sys = システム`システムインスタンス,
      認証情報確認結果 =
      sys.認証情報を確認する(IDフィールド, パスワードフィールド)
    in
      if 認証情報確認結果 = true
      then
        (
          io.print("ログイン画面.タスク一覧画面に遷移する¥n");
          return new タスク一覧画面();
        )
      else return self
  )
  <省略>
end ログイン画面
```



結果と今後の展望

- VDMによる仕様記述から自動生成により，さまざまな仕様記述が生成できる
- 仕様記述間の不整合を防ぐことができる
- より多くの形式の仕様書を生成する手法の考案
  - 画面の紙芝居(静的プロトタイプ)
  - 業務フロー図

まとめ

- 関心事の異なるステークホルダーに合わせて多くの文書を作成・維持することは大変だが，形式手法はそのような課題に対しても有効である
- 今後は，技術者ではない読み手による評価などを行い，提案内容をより良いものにして，要求獲得の道具として使えるものに育てていきたい

皆さま，ありがとうございました!