なぜ形式手法か(B)

SEC形式手法人材育成作業部会(編)

このモジュールについて

品質の高いソフトウェアの効率よい開発へ向け、形式手法の有用性を理解した上で、形式手法の導入に前向きに検討する姿勢の獲得を意図したモジュール。特に正しい仕様の重要性に焦点

- 事前知識・経験
  - ソフトウェア開発の経験と現状のソフトウェア開発に対する問題意識
  - 形式手法の知識・スキルは不要
- 学習目標
  - 形式手法の効果について概略レベルの知識を得る
  - 教材中の例や自身の経験との関連付けにより、仕様の重要性と、正しい仕様の記述に対する形式手法の有用性を理解し導入の検討を開始できる
- 主な学習項目
  - 形式手法の効果と経験的根拠の概要
  - 現場向きの形式手法VDMの概要
  - 日本語や慣習的な非形式的記法の問題

形式手法の効果

- 直接的効果
  - 各種記述物
    - 記述物(要求、仕様、設計、コード、テスト、バグ、etc。)。
    - 証明結果、分析
- 間接的効果
  - 開発対象の認識と理解
  - 開発プロセス改善
  - コミュニケーション
  - 開発者の自信
  - 高品質ソフトウェアの効率的開発
- 「形式手法」とは「書くこと」
  - 書くために理解、認識、議論、分析、共有、etc。
  - 書いてあること V.S. 書いていないこと
    「決めること」

なぜ形式手法か？経験的根拠

- 日本語で正しい仕様を書くのは「不可能」である
  - 曖昧さの排除が困難
  - ツールによるチェックが、ほとんど不可能
  - 仕様の修正に弱い
  - その場で「文法」を考えられない
    - 日本語で仕様を書ききれなくなり、if-then-else など擬似コード的な仕様を書くことが多く、擬似コードの文法を考えている時間が馬鹿にならない
- 形式手法の方が技術移転や蓄積が容易
  - 日本語による意思疎通は難しい
  - 仕様記述言語を読むのは容易
    - UML1.* より容易、UML2.0 より遥かに簡単
  - フレームワークやライブラリの構築が容易
  - 業務知識の蓄積が容易
  - 仕様を「動かしてみる」ことができるので、理解しやすい

形式手法とVDMの有用性

- 理論的にも経験的にも形式手法が有効
- VDMToolsで使用するVDMは、現場寄りの形式手法
  - 証明やモデル検査は、長期的にはやるべきだが、すぐにはできない
- VDM導入は、さほど難しくない
  - 3ヶ月程度の教育とコンサルティング
  - 既存の役立つソフトウェア工学ツールと協調して、より効果が出る
- モデル化は、以下が重要
  - モデル化の範囲を決め、仕様を分割統治
  - 名詞から型、述語から関数または操作
  - 陰仕様を作成してから、静的検証
  - 陽仕様を作成してから、動的検証
- VDMは構造化日本語仕様として使うことができる

VDMとは？

- 数学をベースとした仕様記述と検証のための手法
  - 1960年代から1970年代にIBMウィーン研究所で開発
  - 1996年に、VDM-SLが世界初のISO標準(ISO/IEC 13817)仕様記述言語になった
- 特徴
  - 厳密に定義された仕様記述言語 VDM-SL, VDM++, VDM-RT を持つ
  - 制約条件(不変条件、事後条件、事前条件)の記述が可能
  - ツールによる証明課題の生成
  - 産業界の実用のために拡張された

構造化日本語仕様としてのVDMと日本語仕様

	構文を考える時間	構文チェック	関連チェック	実行テスト	証明問題生成
擬似コードによる仕様	かなりの時間が必要で記法も統一できない	レビューのみ	レビューのみ	具体的データを想定したコードインスペクションに相当するチェックのみ（通常行なわれない）	不可能
VDM仕様	言語マニュアルを参照すれば良いだけ	ツールでチェック	ツールの型チェック	ツールで実行	ツールで生成

構造化日本語仕様としてのVDM

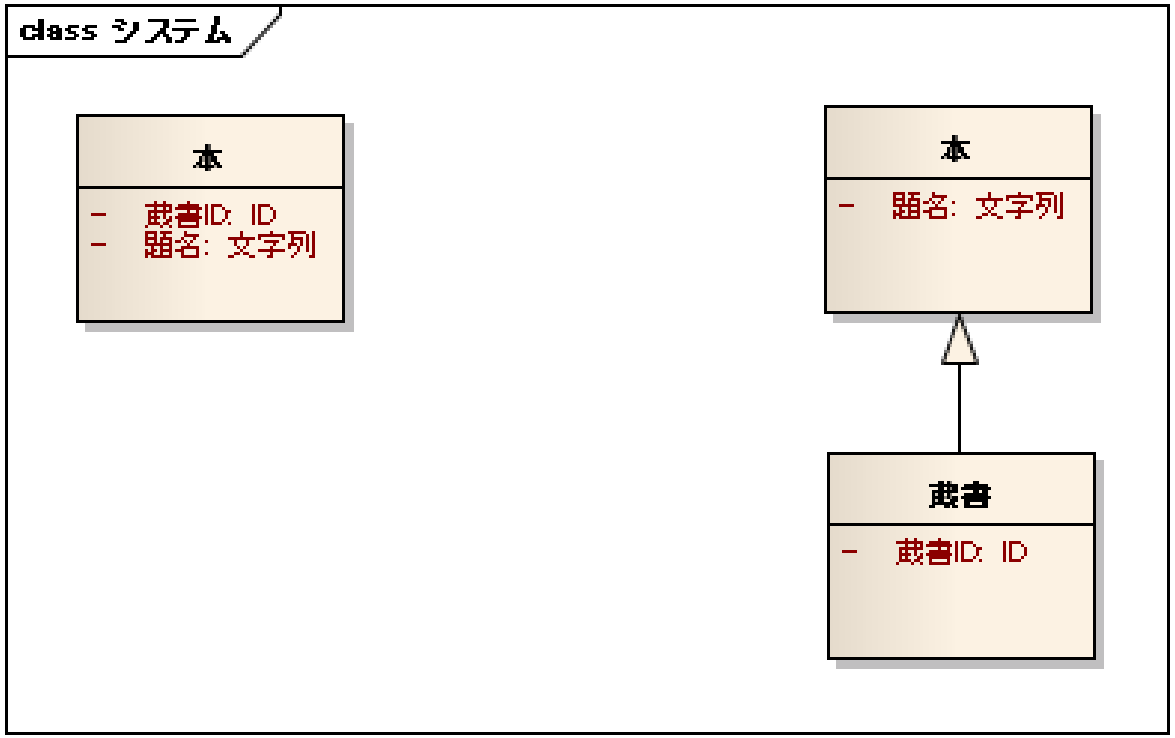
VDMは構造化日本語仕様として使える。日本語仕様は使えない。

- 構文を考える時間が不要である
- 構文・型チェック、証明課題レビューで静的に検証できる
- 証明課題で生成される条件式から、見落としていた不変条件や事前条件が見つかる
- 組合せテストにより、動的に正当性検証ができる
- 回帰テストにより、動的に妥当性検査ができる
- VDMソース自体が、要求辞書ともなる
- 日本語仕様より、記述と検証の工数が少ない
  - 特に、仕様修正に強い
- 擬似コード形式の日本語仕様に近い形で、かなりの部分を記述できる

日本語仕様の曖昧さの例

クイズ

- 以下の2つの「本」は同じか?
  - 本を図書館の蔵書として追加する
  - 題名で本を検索する



日本語仕様の曖昧さの例 エクスプレス予約

- エクスプレス予約サポートセンターとの問答
  - クレジットカードを変更したら、予約できないんですが?
    - カードを変更したら、エクスプレス予約を新規に契約して下さい
  - クレジットカード変更前に予約したe特急券に引き替えようとしたらできないのですが?
    - カードで引き替えできるようになったので、それで引換えて下さい。暗証番号はカードのものを使って下さい
  - カードの暗証番号って、何ですか?
    - クレジットカードの暗証番号です
- 東京駅での問答
  - クレジットカードでe特急券に引換えできないんですが
    - 会員証でやってみて下さい
  - 会員証でもできないんですが
    - 変ですねー、こちら(駅員)でやってみましょう。駄目ですねー
    - ひょっとして、古いクレジットカードではどうですか?
    - あ、できました。はい、切符です

その後判明したエクスプレス予約の用語

- カードの種類
  - エクスプレス予約会員証（変更前、変更後）、クレジットカード(変更前、変更後）
  - その後、今回の件とは無関係と判明したカード
    - EX-ICカード、エクスプレスカード

曖昧さを修正したエクスプレス予約の問答

- エクスプレス予約サポートセンターとの問答
  - クレジットカードを変更したら、予約できないんですが?
    - クレジットカードを変更したら、エクスプレス予約を新規に契約して下さい
  - クレジットカード変更前に予約したe特急券に引き替えようとしたらできないのですが?
    - エクスプレス予約会員証で引き替えできるようになったので、それで引換えて下さい。暗証番号はクレジットカードのものを使って下さい
- 東京駅での問答
  - 変更後のクレジットカードでe特急券に引換えできないんですが
    - エクスプレス予約会員証でやってみて下さい
  - エクスプレス予約会員証でもできないんですが
    - 変ですねー、こちら(駅員)でやってみましょう。駄目ですねー
    - ひょっとして、変更前のクレジットカードではどうですか?
    - あ、できました。はい、切符です

エクスプレス予約の問題は何だったのか?

要求仕様モデルの考慮不足

- モバイル SUICA のクレジットカードを変更すると、エクスプレス予約を新規に契約せねばならず、エクスプレス予約会員証も新規に作成
  - 変更という概念が無い
- 個々の予約が、クレジットカードにリンク
  - クレジットカード変更に弱い
- エクスプレス予約会員証からクレジットカードにリンク
  - クレジットカード変更に弱い

日本語仕様の曖昧さの例　応当日

- 信用取引の決済日(期日)を得る

  弁済期限とは、信用建玉に対して当社がお客様に信用を供与する期限をいいます。弁済期限は、現在のところ6ヶ月のみを取扱っています

  弁済期限が6ヶ月であるということは、信用建玉の建日(信用建玉が約定した日)の6ヶ月目応当日が信用期日となり、この日を超えて建玉を保有することは法律で禁じられています。信用期日が休日の場合には、直近の前営業日が信用期日となります

日本語仕様の曖昧さの例　応当日

- 弁済期限と決済日と信用期日との関係は?
- 応当日とは?
- 応当日が月末日を超えたらどうなるのか?
- 直近の前営業日が5ヶ月後になった場合はどうするのか?

仕様書？

- 口頭
- ホワイトボード
- 「パワーポイント」独自スタイル
- 表
- UML
- 自然言語
- 「VDM」、形式仕様記述言語

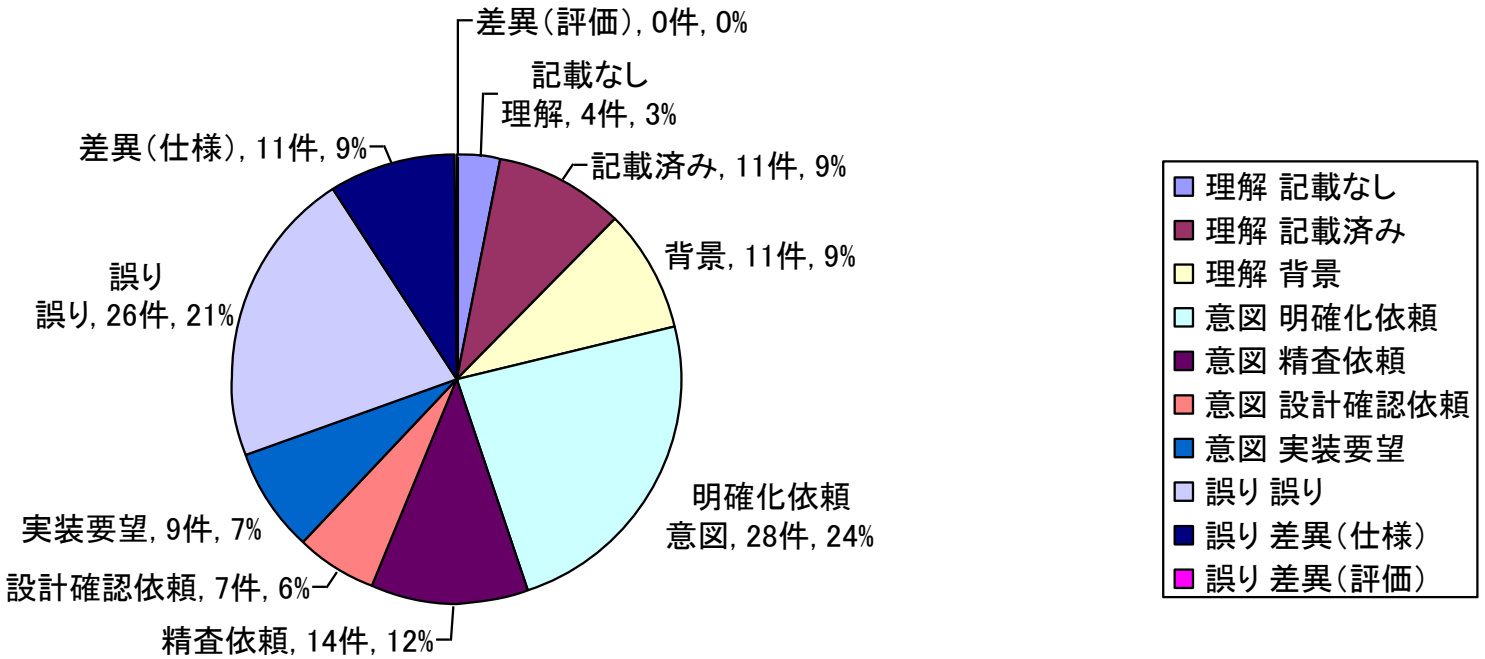
人手によるチェックの限界

- 「これから配布する文書の、ある文字の数をかぞえる【7 分間】

自然言語のコミュニケーション

- 自然言語によるマニュアル

→ 明確化依頼が多い

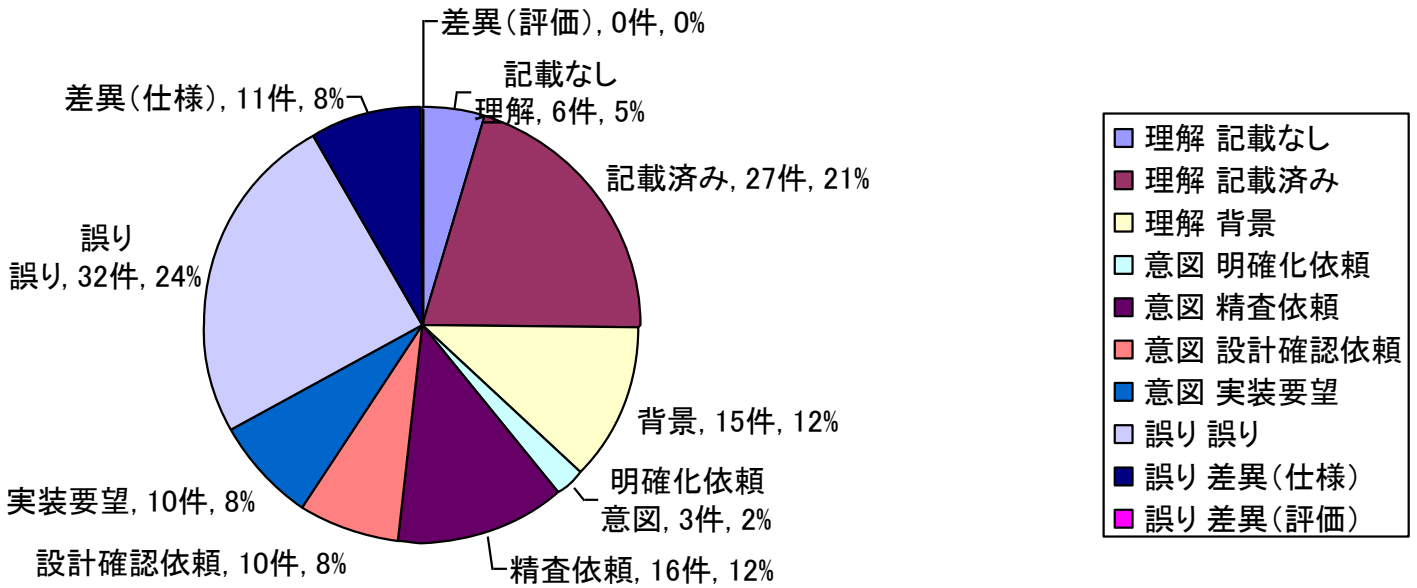


形式手法によるコミュニケーション

- 形式仕様記述言語による外部仕様書

→「理解」に関する質問が多い

→ 仕様策定背景・経緯はコメントに記述する



コミュニケーションのまとめ

- 自然言語・形式仕様記述言語の違い

  → 「理解」と「明確化依頼」以外は似通っている
- 自然言語

  → まずよくわからない… 「明確化依頼」 = 「わかった」でとどまってしまう
- 形式仕様記述言語

  → 中途半端に「理解」できない。背景までつきつめて「理解」したい、納得したい

- 日本語での議論はつらい?

  → 記述から人格を消して「問題対私たち」とする

サマリー

品質の高いソフトウェアの効率よい開発へ向け、形式手法の有用性を理解した上で、形式手法の導入に前向きに検討する姿勢の獲得を目標に、主に以下を学習

- 形式手法の定義と成果例
- 現状のソフトウェア開発の課題と形式手法
  - 信頼性・安全性への期待、国際標準・規格、など
  - 慣習的他手法の限界と、その解決に有用な形式手法の特性