

派生開発における要求分析工程での 影響箇所の検出手法

(株)東芝 ソフトウェア技術センター

大場聡司

satoshi.ooba@toshiba.co.jp

開発における問題点

手法・ツールの適用による解決

派生開発では、流用する既存ソフトウェア資産でデグレードが起きないように

- 顧客や企画が提案した要求・要求仕様を正確に分析した上で、
- それらを実現に伴う変更による影響を受ける機能を網羅的に特定できる必要がある

構造化設計においてシステム間で受け渡すデータの流れを可視化するDFD[1]を応用して、

- 要求・要求仕様レベルのDFDを新旧の要求・要求仕様ごとに生成し、
- 各DFDのデータストアの包含関係や、競合・排他関係を分析することで、正確な分析と影響箇所の網羅を実現する

提案手法のアプローチ

【派生開発における変更による影響箇所の検出を難しくしている要因】

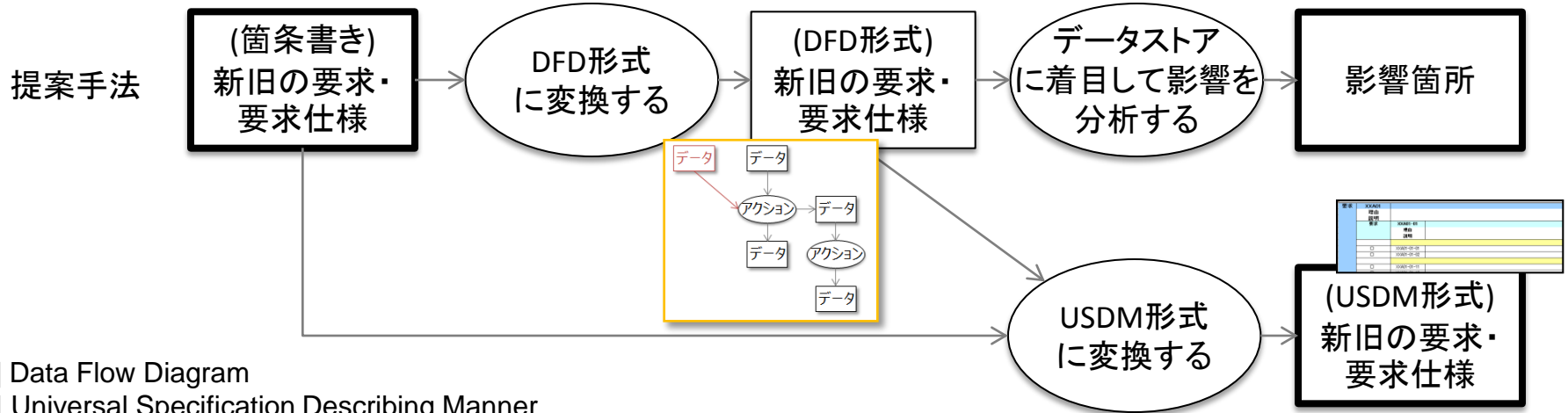
- ① 新しい要求・要求仕様の実現に必要な情報の未文書化
- ② 新旧の要求・要求仕様の文言の違い

文書形式の要求・要求仕様をより直感的に理解できる「構造」に分解することを検討

【提案手法のアプローチ】

- ① 新旧の要求・要求仕様それぞれをDFD変換し、必要なデータを網羅的に洗い出す
- ② 各DFDのデータストアの関係を分析し、影響を受ける要求・要求仕様を検出する

USDM[2]のような階層構造化に加え、データの流れを表現できるDFDが効果があるのではと推測



[1] Data Flow Diagram

[2] Universal Specification Describing Manner

適用実験

まとめ

	提案手法	従来手法(USDMのみ)
実施者	提案者(ドメイン知識あり)	中堅技術者(ドメイン知識あり)
分析結果	影響を受ける流用元の要求仕様を 5個全て※ を検出	影響を受ける流用元の要求仕様を 5個全て※ を検出
所要時間	4.5H (新旧の要求仕様のDFD変換と、比較検討分)	3.0H (新旧のUSDM形式の要求仕様のみを用いた比較検討分)

- 派生開発向け影響分析手法を提案
 - 構造化した新旧の要求仕様で影響箇所を洗い出すことで、変更仕様の見落とし防止できそう
 - 要求仕様のDFD変換など手作業が多いため、自動化手法の検討が現場展開では必須

※全部で約50項目中