

UML Componentsにおける仕様策定プロセスの評価

富士通株式会社

上野 高史

takashi.ueno@jp.fujitsu.com

開発における問題点

手法・ツールの適用による解決

現在のソフトウェア開発では、社内の一定のスキル保持者(該当ドメインに熟知)によってアーキテクチャ設計が行われており、またその設計が個人の何らかの思想・経験に依存したものになっている。

問題点に挙げたようなスキル、属人性に関するような課題解決方法を、コンポーネントベース開発の中の一手法であるUML Componentsを適用することで評価した。

アプローチ

No	問題点	課題解決の方法
1	社内の一定のスキル保持者(該当ドメインに熟知)によってアーキテクチャ設計が行われている	属人性が排除された、設計プロセスの導入
2	アーキテクチャ設計が個人の何らかの思想・経験に依存したものになっている	同上



UML Componentsの特徴

- ◆ 簡潔なコンポーネント化プロセス
- ◆ 分析段階での早期コンポーネント化
- ◆ 設計段階での簡潔なアーキテクチャ
- ◆ 早期のデータ指向な切り分け
- ◆ ツールとしてUMLを使用



評価

観点	仕様策定プロセス		
	ビジネスインタフェースの識別	システムインタフェースの識別	コンポーネントの相互作用
①	○	○	○
②	× ※1	○	△ ※2

※1 コアタイプが検出できなかった場合のビジネスインタフェース識別指針がない
→ビジネスインタフェースなしで進めるのか、コアタイプが検出できるように再度モデル化するのかなどの指針が必要

※2 コンポーネント間の依存関係が循環した場合の指針が存在しない
→循環を回避する方法、循環した場合にどのフェーズまで戻り再設計するのかなどの指針が必要

UML Componentsの仕様策定プロセスを以下の観点で評価

- 観点①：経験に基づく設計スキルがなくてもアーキテクチャ設計ができるか
- 観点②：指針の例外ケースについての対処方法は示されているか

まとめ

- 総括
UML Componentsを適用することで課題解決を行えることの有効性を確認することができたと思う。
- 今後の課題
 - ・評価の際に発見した明確化されていない指針の明確化可否
 - ・「例外ケース」について網羅性の評価