アーキテクチャパターンを利用したWBS抽出支援手法の提案と検証

兪丹萍† 大木幹雄†
日本工業大学　工学研究科　情報工学専攻‡

1．はじめに

プロジェクト管理未経験者がプロジェクト管理を行う多くの場合，プロジェクト計画立案の時点で，プロジェクト達成に必要な作業や成果物を十分に洗い出すことができず，曖昧な計画を立ててしまう．その結果，手戻りや計画の大幅な変更が発生する．そこで，計画立案の前にプロジェクト達成のために必要な作業を，WBS(Work Breakdown Structure)として洗い出す作業が必要になる．

本研究は，プロジェクト管理未経験者のWBSの洗い出しに焦点を当て，任意の作業項目に関連する作業を「作業がもつ関連パターン」に従って洗い出す方法の有効性を検討し，その具体的な手順について考察することを目的とする．研究ではまず，「共通フレーム2007」で定義された作業間の関連構造をパターン化して，ソフトウェアアーキテクチャパターンと類比させて整理・体系化する．次いでパターン化した「共通フレーム2007」で定義された作業の関連構造に占める割合を明らかにする．最後に任意の作業を洗い出したとき，その作業の属するパターンに従って，次々と関連する作業を洗い出す方式の有効性について考察する．

2．共通フレーム2007とソフトウェアアーキテクチャパターン

2．1　共通フレーム2007

共通フレーム2007とは，ソフトウェアの企画から開発，運用，保守，廃棄に至るまでのライフサイクルを通じて作業項目，役割を産業界の総意として包括的に規定するものである．日本の産業界が抱える課題解決のため，ソフトウェアライフサイクルプロセス（SLCP）国際規格（ISO/IEC 12207: JIS X 0160）をベースに日本独自に強化・拡張したものである．作業関連を整理する基盤となる．

2．2　ソフトウェアアーキテクチャパターン

ソフトウェアパターンの一種で，特にソフトウェアのアーキテクチャに関するものをいう．主なパターンは以下のとおりである．

（1）Layers（レイヤ）

抽象度レベルを設け，処理機能を分担するパターンである．プロセス間の作業関連で言うと，分析→設計→プログラム→テストなどの作業の流れにあたる．

（2）Pipes and Filters（パイプ&フィルタ）

同じ抽象レベル機能を順序付け，組み合わせ，抽象度が同じの機能の流れを示すパターンである．機能間のコミュニケーションが重要になる．プロセス間の作業関連に対応付けるとシステム要件の定義と分析，システム要件定義の依頼と結果の承認とソフトウェア要件の定義と分析のような一連な作業に従う作業工程である．

（3）Blackboard（ブラックボード）

抽象度により階層化された処理機能が密接関連する場合，抽象度レベル間のコミュニケーション機能を重視したパターンである．画面設計とデータベース設計，画面デザインなどで抽象度が異なる作業が密接に関連するような作業工程に対応する．

（4）Broker（ブローカ）

処理機能の役割が異なる多数の機能を組み合わせるため，仲介的処理機能が必要で，役割間のコミュニケーション機能を重視したパターンである．データベース設計（スキーマ＝仲介的処理機能）等に出現する作業方法に対応する．

（5）Model-View-Controller（モデル－ビュー－コントローラ）

同等な立場での処理機能の役割分担をするような機能の関係で，M―格納，V－表示，C－監視の3つの機能をもつ役割分担を行うパターンである．オブジェクト指向アプリケーション分析設計作業工程で一般的に見られる作業工程と対応する．

（6）Reflection（リフレクション）

メタレベル（管理者の立場）を設けるような処理機能の役割分担で，抽象レベルごとの役割分担を示すパターンである．モジュール化した作業の関連に対応する．

3．研究概要

本研究は，プロジェクト管理未経験者による

A proposal and verification of WBS elicitation support technique based on software architecture patterns
†Danping Yu, Mikio Ohki
‡Nippon Institute of Technology

WBSの洗い出しの支援を目的にしたもので，主な研究内容は以下のとおりである．

（1）「共通フレーム 2007」の 578 の作業をソフトウェアアーキテクチャパターンを用いて整理・体系化

「共通フレーム 2007」の全体作業項目を基づいて，第2章で示したソフトウェアアーキテクチャパターンの特徴・関係により分類し，図 1 のようにどのパターンに適用するかを分析し，表1で各作業の関連を整理し，体系化した．

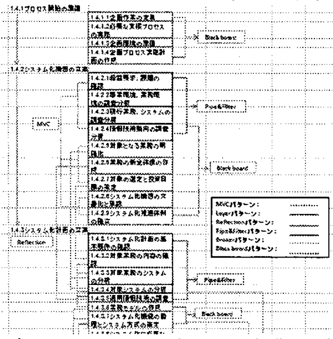

図1. 各パターンで分析した作業関連の一部

表1．各パターンの体系図の一部

MVC	Layer	Reflection	Pipe&Filter	Broker	Blackboard
機能ごとの役割分担支援	階層レベルごとの機能	階層レベルごとの役割分担支援	機能間のコミュニケーション支援	役割間のコミュニケーション	階層レベルごとのコミュニケーション支援
1.4.2.1経営要求，課題の確認 1.4.2.2事業環境，業務環境の調査分析 1.4.2.3現行業務，システムの調査分析 1.4.2.4情報技術動向の調査分析 1.4.2.5対象となる業務明確化 1.4.2.6業務の新全体像の作成 1.4.2.7対象の選定と投資目標策定 1.4.2.8システム化機能の文書化と承認 1.4.2.9システム化推進体制の確立		1.4.2.5対象となる業務明確化 1.4.3.3対象業務のシステムの分析 1.4.3.4対象システムの分析	1.4.2.1経営要求，課題の確認 1.4.2.2事業環境，業務環境の調査分析 1.4.2.3現行業務，システムの調査分析 1.4.2.4情報技術動向の調査分析	1.4.3.8システム化に必要な付帯機能，付帯設備に関する基本方針の明確化 1.4.3.9サービスレベルと品質に対する基本方針の明確化 1.4.3.10プロジェクトの目標設定	1.4.1.1企画作業の定義 1.4.1.2必要な支援プロセスの実施 1.4.1.3企画環境の準備 1.4.1.4企画プロセス実施計画の作成
1.4.3.1システム化計画の基本要件の確認 1.4.3.2対象業務の内容の確認 1.4.3.3対象業務のシステムの分析 1.4.3.4対象システムの分析 1.4.3.5適用情報技術の調査 1.4.3.6業務モデルの作成 1.4.3.7システム化機能の整理とシステム方式の策定 1.4.3.8システム化に必要な付帯機能，付帯設備に関する基本方針の明確化 1.4.3.9サービスレベルと品質に対する基本方針の明確化 1.4.3.10プロジェクトの目標設定 1.4.3.11実現可能性の検討 1.4.3.12全体開発スケジュールの作成 1.4.3.13システム選定方針の策定 1.4.3.14費用とシステム投資効果の予測 1.4.3.15プロジェクト推進体制の策定 [illegible]		1.4.2.6業務の新全体像の作成 1.4.3.5適用情報技術の調査	1.4.3.1システム化計画の基本要件の確認 1.4.3.2対象業務の内容の確認 1.4.3.3対象業務のシステムの分析 1.4.3.4対象システムの分析 1.4.3.5適用情報技術の調査	1.5.2.3新規業務及び業務環境条件の具体化 1.5.2.4機能要件の定義 1.5.2.5非機能要件の定義 1.5.2.6スケジュールに関する要件の定義	1.4.2.5対象となる業務明確化 1.4.2.6業務の新全体像の作成 1.4.2.7対象の選定と投資目標策定 1.4.2.8システム化機能の文書化と承認 1.4.2.9システム化推進体制の確立

（2）パターン化した「共通フレーム 2007」で定義された作業の関連構造に占める割合の調査

「共通フレーム 2007」でソフトウェアアーキテクチャパターンに適用できる作業工程の分布状態を調査した．WBS を作成するときに，あらかじめ，並行作業と前後作業，影響を及ぼし合う作業などが明らかになっているため，次々と作業候補として，提示することができる．

（3）新たな作業間の関連パターン（Overlap パターン）の定義と追加

図 2 のように，ガントチャートから見ると，作業項目が重なっている時もある．プロセス間の作業関連で言うと，設計→プログラムの作業の流れの時，設計がある程度進んだ時，プログラムは次の設計と重なって進んでいくことである．それを新たなパターン（Overlap パターン）に定義した．パターン化できなかった作業項目について，Overlap パターンに属する作業関連を洗い出し，ソフトウェアアーキテクチャパターン化した作業関連に加えると，パターンによって特定できる作業関連の割合がアップした．

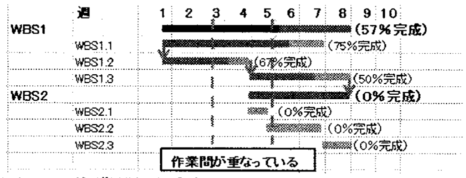

図2. 作業間が重なっているガントチャート

4．検証結果

4．1 ソフトアーキテクチャパターンで分析したデータの集計図

「共通フレーム 2007」の 578 の作業を分析した結果とそれぞれのソフトアーキテクチャパターンが適用できた作業割合を表 2 および図 3 に示す．

表2. 各パターンの作業数

MVC	187	32.4%
Layer	28	4.8%
Reflectio	27	4.7%
Pipe&Filte	83	14.4%
Broker	48	8.3%
Blackboa	158	27.3%
Overlap	24	4.2%
Other	23	4.0%
Total	578	1

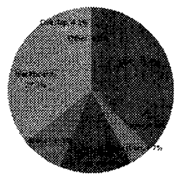

図3. データの集計図

4．2 考察

分析した結果により，次のような効果が明らかになった．

（1）「共通フレーム 2007」がソフトアーキテクチャパターンに適用

「共通フレーム 2007」をソフトウェアアーキテクチャパターンと類比させて整理・体系化した．さらに，任意の作業を洗い出したとき，その作業の属するパターンに従って，次々と関連する作業を洗い出すことができる．

（2）作業の洗い出しの漏れの防止

図 3 の結果により，「共通フレーム 2007」の 91.9%の作業がソフトウェアアーキテクチャパターンでパターン化したため，それを利用すると，作業の漏れを 91.9%防ぐことができる．

（3）作業の関連構造に占める割合の上昇

新たなパターンを加えて，さらに作業関連を洗い出すことができたことより， パターン化した作業が 96.0%まで高めることができた．

5．おわりに

本稿では作業の漏れの問題を解決する手法の提案と期待効果を述べた．今後，本手法のツール化が必要となる．

参考文献

[1]大木幹雄，根本安曇，増渕和也，兪 丹萍：“WBS の誘導抽出支援ツール「WBSEditor」の試作と検証” 第 70 回情報処理学会 5Q-3(2008).