3B-7

# モジュール化された製品における選択リリース方式の提案

芳賀　悠一[†]　上野　浩一郎[†]　秋間　孝道[††]
三菱電機株式会社　情報技術総合研究所[†]
三菱電機株式会社　インフォメーションシステム事業推進本部[††]

## 1.　はじめに

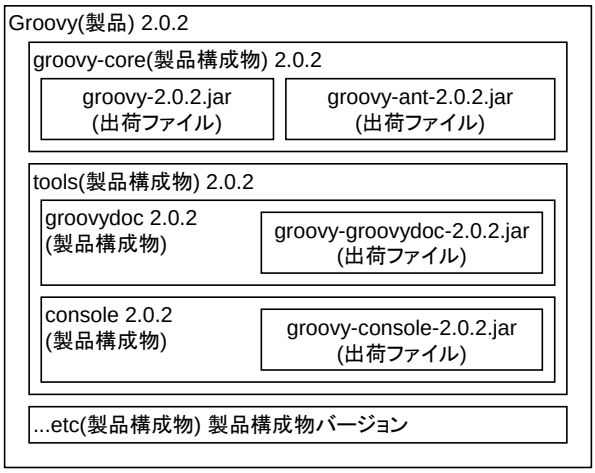
従来，ソフトウェア製品開発におけるソースファイル等のバージョン管理は数多くの開発支援ツールによって実現されてきた．一方，製品のバージョンやリリースの管理については，開発プロジェクト毎に独自管理されている現状がある．また，その管理が属人的になっている場合も多く，管理の複雑化や管理情報の不足，管理作業の引継ぎが困難になる状況が発生する問題がある．この問題を解決するため，筆者等は，開発プロジェクト間で統一化したリリース管理方式の定義に取組んでいる．

本稿では，モジュール化された製品を対象とした選択リリースの管理方式について述べる．本方式で管理データと管理フローを定義することで，管理の複雑化や属人的管理を防止し，リリース後のリリース情報のトレースを確実にする．

## 2.　選択リリースとは

近年，ソフトウェア開発において場当り的に繰返される改変による製品の肥大化を抑止するため，製品のフィーチャ（特徴や機能など）をモジュール化して他の製品に流用することで，製作コストの削減を図る動きがある．モジュール化により製品が製品構成物(製品を構成する部品)に分割されることで，流用性の向上が図られ，また，選択リリースが可能となる．

本方式で取り扱う選択リリースは，リリース先毎に取捨選択された製品構成物をリリースすることを指す．製品が適切にモジュール化された場合，製品構成物と出荷ファイル(製品構成物を構成する出荷対象のファイル)との関係は 1 対多となる．




※例として，OSSの動的プログラム言語Groovyを挙げる．

**図 1　製品構成例(Groovy[1]より抜粋)**

### 2.1.　リリース管理対象

本方式における管理対象は，製品，製品構成物，出荷ファイルである(図 1)．製品構成物に含まれる出荷ファイルは，開発チームで管理するものとし，リリース管理チームでは製品構成物との対応関係のみ管理対象とする．

### 2.2.　課題

選択リリースでは，リリース先毎に選択する製品構成物および製品構成物バージョンが異なり，リリース管理が複雑である．そのため，あるリリース先で発生した障害に対して，他のリリース先を追跡して，障害を含む出荷ファイルを含むリリース先を特定することが困難という課題がある．

## 3.　選択リリース方式の提案

本稿では，リリース後の障害発生時の障害影響範囲を特定（トレース）するため，本章で述べるリリース管理データおよびそれを用いたリリース管理フローのモデルを定義し，この 2 つを用いた選択リリース方式を提案する．

### 3.1.　リリース管理データ

リリース情報を管理するためのリリース管理データのモデル（クラス図）を図 2のように定

“Proposal for selective release process for modular products”
Yuichi HAGA[†],Koichiro UENO[†],Takamichi AKIMA[††]
[†]Information Technology R&D Center, Mitsubishi Electric Corporation.
[††]Information System ＆ Network Service Group, Mitsubishi Electric Corporation.

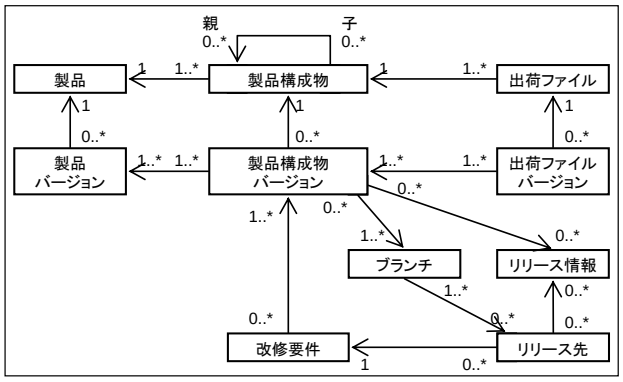



図 2　リリース管理データのモデル概要

義した．このモデルでは，製品，製品構成物，出荷ファイルの各バージョン間での連携を定義することで，バージョンレベルでのトレースを可能にした．

3.2.　リリース管理フロー

3.1節のリリース管理データを適切に管理するため，リリース管理フローのモデルを定義した．リリース管理フローでは大別して，表 1のように 4 作業項目，11 作業に整理した．例えば，作業『[3-1]リリース計画の立案』ではリリースする製品構成物の選択を行い，リリースユニット(リリースする製品構成物の一式)の構成が計画される．これらの作業は既存のプロセス[2][3]にも対応する作業は存在するが，リリース管理という 観点で一貫したプロセスとなっていない。そのため本フローでは，製品の定義からリリース後のインシデント管理まで一貫した作業とそのフローを定義した．

3.3.　効果の例

3.1節の図 2のデータ管理，3.2節の管理作業により，次のトレースが可能となる．

図 3に，障害発生時のインシデント管理におけるリリース管理データの参照例を示す．インシデント管理は，リリース先から障害連絡を受け，障害対応チームに解析を依頼する．解析の結果，障害混入した『出荷ファイルバージョン』を特定し，その『出荷ファイルバージョン』以降の出荷ファイルが含まれる『製品構成物バージョン』を特定する．これらの『製品構成物バージョン』には障害が混入している可能性が高く，これらのリリース先を『リリース情報』から特定する．

これにより，障害対応後の製品を障害の発生したリリース先以外にも展開し，障害の発生を未然に防ぐことが可能である．また開発においては障害の対応漏れを防ぐことが可能となる．

表 1　作業項目

| No. | 作業項目 | 作業 | 参考プロセス |
|---|---|---|---|
| 1-1 | 製品定義 | 製品階層の定義 | SLCP：2.2.2.1 構成識別体系の確立 |
| 1-2 | | バージョン構成の計画 | |
| 2-1 | 製品構築 | 製品構成物間依存関係の確認 | - |
| 2-2 | | 製品の構築 | SLCP：2.2.3.1 構成の変更管理 |
| 3-1 | 製品リリース | リリース計画の立案 | ITIL：リリース計画立案 |
| 3-2 | | リリースユニット構築 | ITIL：リリースの設計，構築，開発 |
| 3-3 | | リリースユニット試験 | SLCP：2.2.5.1 構成品目の完全性の保証<br>ITIL：リリースの受入 |
| 3-4 | | リリース審査 | SLCP：2.2.6.1 リリース及び出荷の制御<br>ITIL：配布 |
| 3-5 | | リリース実施 | |
| 4-1 | インシデント管理 | リリース先コミュニケーション管理 | SLCP：2.2.4.1 管理者記録と状況報告の準備 |
| 4-2 | | プロダクト変更管理 | |

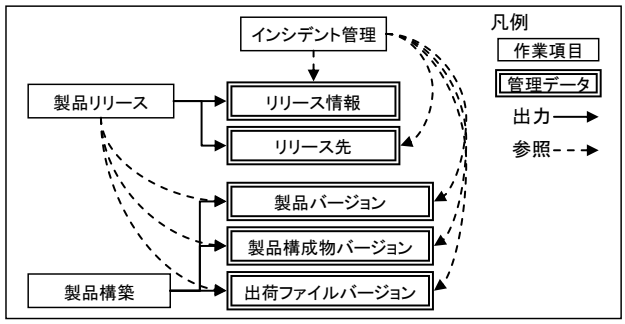



図 3　トレースにおける管理データ参照例

4.　おわりに

本方式では選択リリース管理のためのリリース管理データとリリース管理フローについて定義した．本方式にてリリース管理を行うことにより，リリース情報のトレースが確実になる．

今後は，本方式を開発プロジェクトに適用し，検証および改善を行う．

**参考文献**

[1]Groovy, http://groovy.codehaus.org/
[2]独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター：共通フレーム 2007 第 2 版，pp.160-163
[3]ITIL, http://www.itil.co.uk/index.htm