1N-3

メタデータ駆動クラウドサービスアーキテクチャの可変性依存関係パターンの提案と評価

黒野 望† 中道 上‡ 青山 幹雄‡

南山大学大学院 数理情報研究科† 南山大学 情報理工学部 ソフトウェア工学科‡

1.研究の背景と課題

SaaS クラウドサービスで，マルチテナントが支援されている．マルチテナントは，リソースを共有しながら，複数のテナントにサービスを提供できる．マルチテナントの実現方法に，メタデータ駆動アーキテクチャが注目されているが，その設計方法は確立されてない．

2.関連研究

2.1 メタデータ駆動アーキテクチャ

メタデータ駆動アーキテクチャは，メタデータによりリソースを制御するアーキテクチャスタイル[5]である．メタデータを基に動的なインスタンス制御を行うことで，テナント毎の多様な要求へ柔軟に対応する．

2.2 OVM(Orthogonal Variability Model)

OVM は，プロダクトラインの可変性を表現するモデル[3]である．OVM では，機能の表現を省略し，可変性のみ表現できる．

2.3 マルチテナントの可変性モデリング

マルチテナントの可変性モデルが提案されている[1]．可変性を OVM を用いてモデル化しているが，表現抽象度が高く，マルチテナントの制御を適切に表現できていない．

3.アプローチ

メタデータはソフトウェアプロダクトラインの可変性を表現できる．そこで，メタデータの可変性がアプリケーションの可変性に連動していることに着目し，可変性依存関係パターンを提案する．

可変性依存関係パターンにより，アプリケーション開発における各プロセスで，メタデータとアプリケーションの可変性を表現するメタデータ駆動アーキテクチャモデルの作成が容易となる．

4.可変性依存関係のパターン化

本稿では，メタデータとアプリケーションの可変性を関心事毎に分類し，依存関係をパターン化する．

4.1 MVC に基づく可変部の分類

メタデータ駆動アーキテクチャでは，要求の可変性がメタデータの可変性に影響し，メタデータの可変性がアプリケーションの可変性に影響する．各レベルの可変性はモデル，ビュー，コントローラに関心事を分割できるため，可変性を MVC に分類する．

4.2 OVM の拡張

メタデータ駆動アーキテクチャの依存関係は，次の 3 つに分類できる．(1)メタデータ間の制約依存関係，(2)メタデータからアプリケーションへの制約依存関係，(3)アプリケーション間の制約依存関係である．しかし，従来の OVM では，可変部と変異体の制約依存関係として，要求と排除しか表現できない．そのため，メタデータ間，メタデータからアプリケーションへの依存関係を正確に表現できない．本稿では，この制約依存関係を表現できるように OVM の制約依存関係の表現を拡張する(表 1)．

Refinement, Restrict, Update, Reference の 4 つの制約依存関係を拡張した．Refinement では，メタデータの可変部により，親子関係が構成されるため，依存関係を拡張した．Restrict, Update, Reference では，メタデータの変更により，アプリケーションのパラメータ値を制限や更新，参照するため，制約関係と依存関係を拡張した．

表 1 OVM の制約依存関係の拡張要素

	従来のOVM	拡張OVM	説明
M-M 依存関係	Requires	Extend	機能を拡張
	なし	Refinement	可変部を詳細化
	Excludes	Excludes	他の可変部や変異体を排除
M-A 依存関係	Requires	Generation	インスタンスを生成
	なし	Restrict	他の要素のパラメータを制限
	なし	Update	他の要素のパラメータを更新
	なし	Reference	他の要素のパラメータを参照
A-A 依存関係	Requires	Use	他の機能を利用
	Excludes	Excludes	他の可変部や変異体を排除

4.3 メタデータ駆動アーキテクチャのパターン化

メタデータとアプリケーションの可変性を，MVC に分類したので，M，V，C で依存関係をパターン化する(表 2)．各可変性依存関係を構造，生成，振舞いの 3 点に分類し，パターン化する．

(1) M-M 依存関係

構造の視点から構成要素と機能拡張として依存関係をパターン化した．構成要素では，各要素における可変性がその要素を構成する要素を決定することを考慮し，パターン化した．機能拡張では，メタデータの可変性により新たな機能が拡張されることを考慮し，パターン化した．

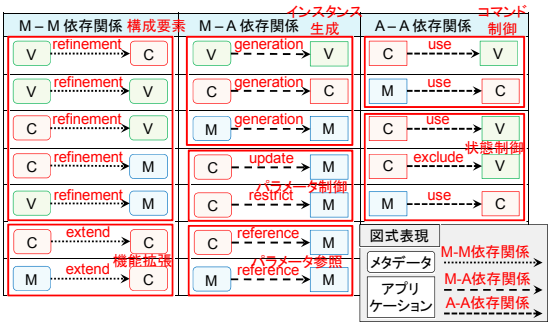
Variability Dependency Patterns for Metadata-Driven Cloud Service Architecture

†Nozomi Kurono, Graduate School of Mathematical Sciences and Information Engineering, Nanzan University.

‡Noboru Nakamichi, Mikio Aoyama, Department of Software Engineering, Nanzan University.

Copyright ©2013 Information Processing Society of Japan.
All Rights Reserved.

表 2 可変性依存関係パターン



(2) M-A 依存関係

生成の視点からインスタンス生成，パラメータ制御，パラメータ参照として依存関係をパターン化した．インスタンス生成では，メタデータに対応するインスタンスが生成されると考え，パターン化した．パラメータ制御では，メタデータの制御による生成インスタンスのパラメータが制限される場合を考慮し，パターン化した．パラメータ参照では，生成インスタンスが他のパラメータを参照する場合を考慮し，パターン化した．

(3) A-A 依存関係

振舞いの視点からコマンド制御と状態制御の依存関係をパターン化した．コマンド制御では，ユーザの入力や制御内容により他の要素の振舞いが変更されることを考慮し，パターン化した．状態制御では，利用するユーザの役職や業務に応じたアクセス制御を考慮し，パターン化した．

また，メタデータ駆動ではメタデータによる動的なインスタンス制御を行っているため，パターン適用として，M-M，M-A，A-A の順に依存関係を決定する．

5. パターンの適用と評価

Salesforce.com 社[4]の提供する Force.com 上でのアプリケーション作成プロセスを例に，提案パターンを用いてメタデータとアプリケーションの可変性依存関係をモデル化し，評価した．

5.1 シナリオへの適用

提案パターンをアンケート管理アプリケーション[2]の作成プロセスに適用した．アンケート管理作成プロセスに沿って可変部を MVC に分類した．可変性依存関係パターンを用いて，アンケート管理作成プロセスを構造，生成，振舞いの順に可変性依存関係をモデル化した(図 1)．例として，CustomObject の可変性依存関係を説明する．まず，構造の視点から可変性依存関係を示す．アンケート管理では，アンケート管理オブジェクトを生成後，オブジェクトに対して入力制限と承認オブジェクトを生成している．入力制御と承認オブジェクトは C の性質であり，機能を拡張していることからパターンより機能拡張の可変性依存関係で示した．同様に，生成，振舞いの順にプロセスから MVC の性質や目的を判断し，可変性依存関係を示した．

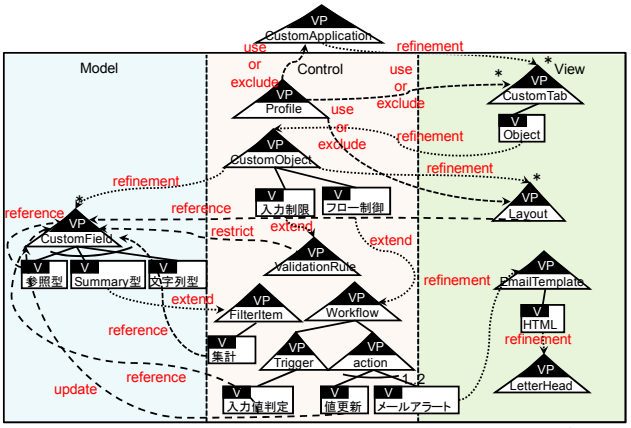


図 1 アンケート管理におけるアーキテクチャモデル

5.2 評価

(1) 拡張した OVM に関する評価

メタデータ駆動アーキテクチャの可変性依存関係を示すために，従来の OVM では表現できない M-A 依存関係の表現方法を拡張した．例題に適用した結果，ValidationRule から CustomField，Trigger から CustomField の依存関係を示せることが確認できた．

(2) 可変性依存関係パターン化に関する評価

アプリケーションで発生する可変性を MVC の 3 つに分類し，パターン化した．テナントが望む変更の関心事が明確になり，その影響範囲が特定できる．また，デザインパターンの目的である構造，生成，振舞いの 3 つに分類し，パターン化を行うことにより，各可変性依存関係の目的が明確化できた．

(3) 開発プロセスへのパターン適用に関する評価

テナントの多様な要求を実現するための全ての可変性の特定とモデル化は困難である．開発プロセスにパターン適用することで，各プロセスから性質と目的でメタデータとアプリケーションの可変性とその依存関係を特定できる．また，特定のプロセスの可変性のみ記述するため，モデルの複雑性が軽減できる．

6. まとめ

メタデータの可変性とアプリケーションの可変性が連動していることに着目し，MVC の粒度でメタデータとアプリケーションの可変性依存関係のパターンを提案した．提案したパターンをアプリケーション開発のプロセスに適用し，モデル化することで評価した．

参考文献

[1] 今岡 純二，Force.com のすべて，日経 BP，2011
[2] R. Mietzner, et al., Variability Modeling to Support Customization and Deployment of Multi-Tenant-Aware Software as a Service Applications, Proc. PESOS ‘09, May 2009, pp. 18-25.
[3] K. Pohl, et al., Software Product Line Engineering: Foundations, Principles and Techniques, Springer, 2005.
[4] Salesforce.com, http://www.salesforce.com/jp
[5] C. D. Weissman, The Design of the Force.com Multitenant Internet Application Development Platform, Proc. ACM SIGMOD ’09, Jun 2009, pp. 889-896

Copyright ©2013 Information Processing Society of Japan.
All Rights Reserved.