# ワークフロー型 Web アプリケーションのための アクセス制御に関する研究
## ～　アスペクト指向を用いたプロダクトライン化の試み　～

M2004MM022 加藤 隆広
指導教員　野呂昌満

## 1　はじめに

近年，インターネット環境の整備が進むにつれ，ワークフロー型 Web アプリケーションの開発が頻繁に行われるようになっている．ワークフローとは，電子的な方法でビジネスプロセスを定義し，手続きの処理手順を規定することである．関係者の間を情報や業務が円滑に流れるようにするためのシステムがワークフロー型のアプリケーションである．

ワークフロー型 Web アプリケーションは、不特定多数のユーザによる利用が想定される．したがって、セキュリティに対する要求は高い．Web アプリケーションにおけるアクセス制御のためのセキュリティ対策としては，公開鍵暗号基盤 (PKI)[8] が重要な機構である．しかしながら，PKI の機能ではユーザ認証が可能であるが，エンドシステムのアクセス制御は行えない．

一方，エンドシステムのアクセス制御の枠組みを実現する方法としては，さまざまなモデルが研究され，提案されている．それらは任意アクセス制御 (DAC) モデルと強制アクセス制御 (MAC) モデルに大きく分類される [2]．また，ポリシー中立的な，ロールに基づくアクセス制御モデル (Role Based Access Control)[7] が注目されてきている．しかし，ワークフロー型 Web アプリケーションにおけるアクセス制御モデルの実現に関する研究はいまだ初期段階である [2]．アプリケーションロジック内に散在してしまうアクセス制御ロジックによって，システムの保守性，再利用性を妨げる要因のひとつとなっている．

本研究の目的は，ワークフロー型 Web アプリケーションにおけるアクセス制御ソフトウエアのプロダクトライン化を行うことによって，アクセス制御ソフトウエアの再利用性を向上させることである．アクセス制御ソフトウエアのプロダクトライン化を実現するにあたっては，アクセス制御ロジックを部品化することが必要である．しかし，前述したようにアクセス制御のロジックは，アプリケーションロジック内に散在しているので，部品はアプリケーションロジックのコンテクストに依存してしまう．部品をアプリケーションロジックのコンテクストから切り離す手段としてアスペクト指向の考え方を用いる．アクセス制御ロジックを横断的関心事 (crosscutting concern) として抽出することにより，アプリケーションロジックのコンテクストからの分離が可能となる．ワークフロー型 Web アプリケーションのモデルとして，企業内勤怠管理システムを想定し，実現を試みる．

アクセス制御ソフトウエアのプロダクトライン化を実現することによって，異種システムや分散システム間での大規模な相互接続，あるいはシステムの統合といった要件に対応しうるシステムの保守性，再利用性の向上につながると考える．

## 2　ワークフロー型 Web アプリケーションのアクセス制御の実現

ワークフロー型 Web アプリケーションのモデルとして，企業 A の単純な勤怠管理システムを仮定する．企業 A の組織は，複数の部門の下にそれぞれ複数の project があり，project ごとに従業員 (employee) が所属する．employee の登録した勤怠情報は project の責任者である project leader が審査し，審査済の勤怠情報を部門の責任者である project manager が承認することによって登録が終了する．

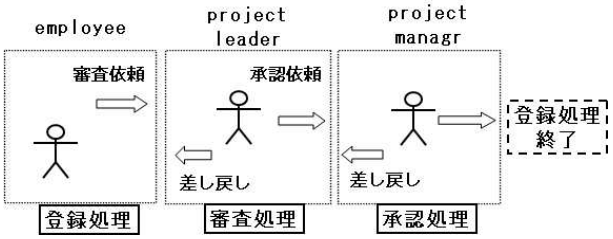


図 1　勤怠管理システムのワークフローモデル

### 2.1　勤怠登録システムのアクセス制御要件

勤怠管理システムにおけるアクセス制御要件としては，画面遷移に関する制約と勤怠情報に関するデータベースに対する制約に大別することができる．このうち画面遷移に関する制約を図 **2** に示す．

企業内の勤怠管理システムにおける画面遷移に関するアクセス制約は，職位ごとに設定することができる．これにより，画面遷移に関するアクセス制御の実現には RBAC モデルを使用することができると考える．

| アクセス主体 (subject) | アクセス対象(object) | | | | | |
|---|---|---|---|---|---|---|
| 職位(role) | 画面 | | | | | |
| | 勤怠入力 | 審査依頼 | 承認依頼 | 審査 | 承認 | 差し戻し |
| employee | ○ | ○ | − | − | − | − |
| projectleader | ○ | ○ | ○ | ○ | − | ○ |
| project manager | ○ | ○ | ○ | ○ | ○ | ○ |

○ : 表示権限あり
− : 表示権限なし

図 2 勤怠登録システムの画面アクセス制約

### 2.2 画面遷移に関するアクセス制御の実現

アクセス制御技術では，情報資源の不正利用を防止する目的で，決められた規則に従って資源へのアクセスを制限する [9]． アクセス要求が発生するたびに，アクセス主体 (subject) とアクセス対象 (object) の間でアクセス権限を判定する機能を果たすのがアクセス制御ソフトウエアであり，アクセス制御ソフトウエアはアクセス制御ロジックによって実現される．画面遷移時におけるアクセス制御ロジックの起動については，**図 3** に示す状態遷移機械で表すことができる．状態遷移時に Action を含

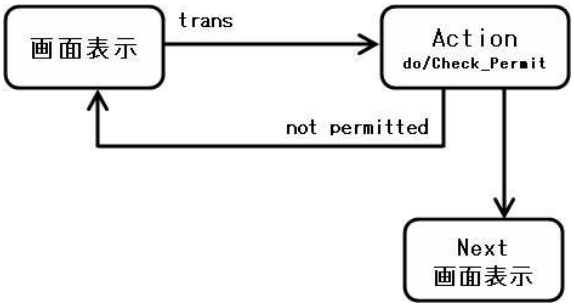


図 3 画面遷移時の状態遷移機械

むモデルは，**図 4** のように state パターンと command パターンによって実現することが可能である．アクセス制御ロジックは，command パターンの中で実現される．しかし，state パターンと command パターンによって実現した場合，**図 5** に示すように Action の呼び出しが state パターン内に横断的関心事として散在し，さらに command パターン内にアクセス権限の判定処理の呼び出しが crosscut する形となる．

## 3 アクセス制御ロジックの部品化

アクセス制御ソフトウエアをプロダクトライン化する前提として，アクセス制御ソフトウエアを構成するロジックを部品化することが必須である．アクセス制御ロジックを部品化するにあたっては，crosscut しているアクセス制御ロジックをアプリケーションロジックから分離する必要がある．アクセス制御ロジックをアプリケーションロジックから分離する方法として，アスペクト指向によるオペレーションフックを用いた実現方法を以下に示す．

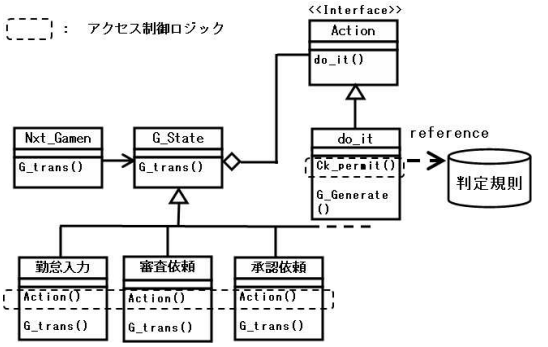


図 4 アクセス制御を含む画面遷移の実現

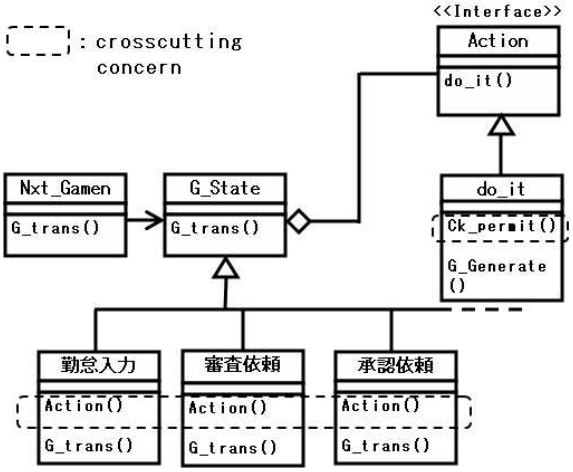


図 5 画面遷移時の横断的関心事

### 3.1 アスペクト指向による横断的関心事の分離

横断的関心事を分離するために，アスペクト指向設計のオペレーションフックの考え方を用いる．判定処理を呼び出す join point をアスペクト間記述に記述することにより，concern を分離したものが**図 6** である．**図 5** で横断的に存在していた concern のうち，まず command パターンである Action に関する concern を Aspect として抽出する．さらに，command パターン内に存在するアクセス制御に関する concern を Aspect として抽出する．これによって，アプリケーションロジックからアクセス制御ロジックを分離することができる．アクセス制御ロジックを分離することによって抽出した部品を**図 7** に示す

## 4 プロダクトライン化の試み

2 節で勤怠登録システムの画面制御に関するアクセス制御ロジックを RBAC モデルに基づいて実現した．3 節では，アクセス制御ロジックをアスペクト指向設計によってアプリケーションロジックから切り離すことを行った．しかしながら，ワークフロー型 Web アプリケーションにおけるアクセス制御ロジックは，画面遷移時以

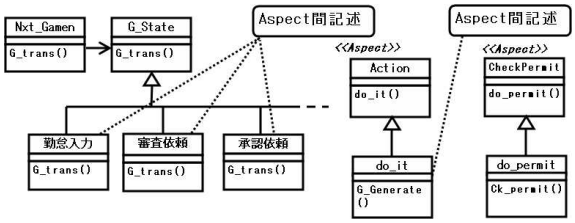


図 6　オペレーションフックを用いた concern の分離

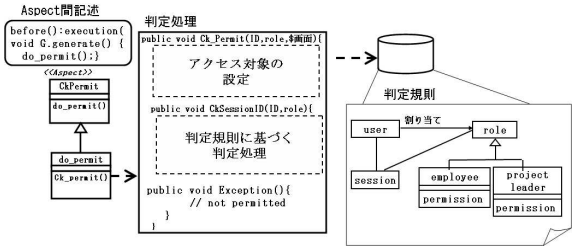


図 7　アクセス制御ロジックの部品

外にもデータベースアクセス時等で必要であり，RBAC モデルのみで表現できるとは限らない．

さまざまなアクセス制御の要件に柔軟に対応し，アクセス制御ソフトウエアの保守性・再利用性の向上を図るため，RBAC モデルだけでなく，DAC モデル，MAC モデルに基づいたアクセス制御を想定し，アクセス制御のプロダクトライン [4] 化を試みる．

ソフトウエアプロダクトライン技術を用いることによって，ソフトウエアの生産性や一貫性を向上させるだけでなく，市場投入にかかる時間，費用，エラーの割合を減少させることが可能である [4]．

アクセス制御ソフトウエアを FODA[3] モデルのフィーチャーダイアグラムを用いて要求分析を行ったものが，**図 8** である．

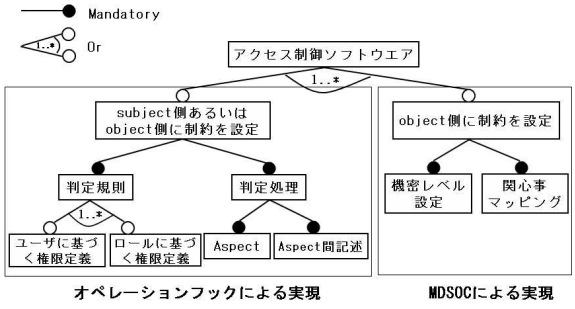


図 8　アクセス制御ソフトウエアのフィーチャーダイアグラム

実現にあたっては，DAC モデルについては，RBAC モデルと同様，アクセス主体に基づいて権限が設定されるため，オペレーションフックを用いた実現を行う．一方，MAC モデルについては，アクセス対象自体に権限を保持するのが特徴であるため，むしろ MDSOC[5] の考え方であるハイパースライスを用いた実現を行う．
アクセス制御モデルが決定されると，それぞれのフィーチャーをたどって最下位のフィーチャーを抽出する．最下位のフィーチャーに部品が対応し，アクセス制御ソフトウエアを構成する．このうち，判定処理部品に関しては，アクセス制約が設定される subject によって可変的なふるまいを行う必要がある． このため，**図 9** に示す

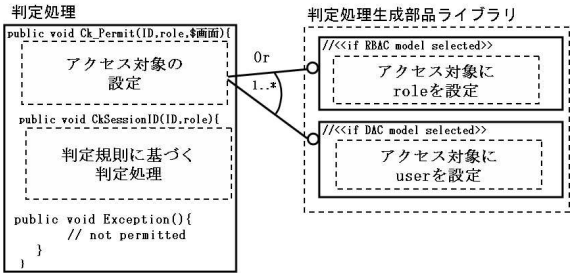


図 9　判定処理部品の生成

ように subject に関する情報を設定し，判定処理部品を生成するための判定処理部品生成部品ライブラリを用意する．あらかじめアクセス制御モデルに対応するフィーチャーごとの判定処理部品を生成するための部品ライブラリを用意しておくことによって，可変的に判定処理部品を生成する．

MDSOC を用いて勤怠登録システムの画面制御に関するアクセス制御ロジックを実現した場合の concern マッピングを，**図 10** に示す．

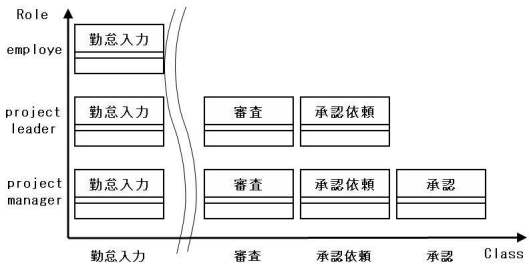


図 10　MDSOC を用いた場合の concern マッピング

## 5　考察

本研究では，ワークフロー型 Web アプリケーションのモデルである企業 A の単純な勤怠管理システムの画面制御を前提に，アクセス制御ソフトウエアのプロダクトライン化を試みた．プロダクトライン化を実現するにあたって，画面遷移時のアクセス制御に関する concern を Aspect 指向の手法を用いて分離し，アクセス制御ロジックの部品化を行った．さらにフィーチャーダイアグラム

を用いて Aspect の使い方，部品の実現方法の違いを抽出することとした．本研究で実現したプロダクトライン化によって，アクセス制御ソフトウエアの再利用性・保守性の向上について一定の成果を得ることができたと考える．

### 5.1 プロダクトラインとしてのバリエーションの拡張について

一般的なワークフロー型 Web アプリケーションを対象としてアクセス制御ソフトウエアのプロダクトラインを適用する場合を想定した拡張が必要である．アクセス主体 (subject) が何であるか，アクセス対象 (object) が何であるかによって，アクセス制御ソフトウエアのプロダクトラインは適切な部品を提供しなければならない．

本研究で提案した方法では，アクセス主体に対応する部品は，オペレーションフックを用いる場合には判定規則のバリエーションとして，MDSOC を用いる場合には機密レベルの設定条件として要素を持たせている．アクセス主体が，user または role に限らず，task あるいは process であっても同様に要素のバリエーションを追加することで実現が可能である．

アクセス対象に対応するには，アクセス対象ごとにアスペクトを生成し，アスペクト間記述に対象へのアクセスの point cut を記述する，あるいは対象ごとにハイパースライスを生成することで実現が可能である．

### 5.2 アクセス制御ソフトウエアの一般化について

本研究では，企業 A の単純な勤怠管理システムというワークフロー型 Web アプリケーションにおいてアクセス制御ソフトウエアのプロダクトライン化を実現するにあたり，状態遷移機械における遷移時に Action からアクセス制御ロジックを分離する方法を示した．

遷移時に Action を伴う状態遷移機械は，ワークフロー型 Web アプリケーションに限らず，一般的なアプリケーションでも用いられる．したがって，本研究で示したアクセス制御ソフトウエアのプロダクトラインは，一般的なアプリケーションにも適用が可能であると考える．

また，状態遷移機械における遷移時に Action からアクセス制御ロジックを分離する方法は，アクセス制御ロジック以外にも適用することが可能である．このことは，一般な状態遷移機械の Action をプロダクトライン化する可能性を示唆するものであると考える．

## 6 おわりに

本研究では，ワークフロー型 Web アプリケーションにおけるアクセス制御ソフトウエアを横断的関心事として抽出し，アプリケーションロジックからのアクセス制御部品の切り離しを可能とした．また，アクセス制御ソフトウエアのプロダクトライン化によって，アクセス制御モデルの変更，あるいはアクセス制御を行うタイミングの変更にも対応することが可能となった．これにより，アクセス制御ソフトウエアの保守性・再利用性を高めることができたと考える．

しかしながら，アクセス制御の要件が状態遷移によって動的に変化する場合，あるいは権限の階層化が発生する場合などを考慮するには至っていない．これら複雑なアクセス要件への対応，さらに品質確保の手段に関しては今後の課題としたい．

**謝辞**

本研究を進めるにあたり多大な助言を頂き，また，熱心にご指導下さいました野呂昌満教授に深く感謝いたします．また，親身になってご指導および的確なアドバイスを頂いた張漢明助教授，同じ研究室の仲間としてさまざまなアドバイスを頂いた石見知也君，小久保佳将君，八木晴信君，石川智子君，坂野将秀君，久松康倫君，本多克典君，水野耕太君に深く感謝いたします．

## 参考文献

[1] Eclipse Project, http://eclipse.org/, Feb 2006.

[2] J. B. D.Joshi, W. G. Alef, A. Ghafoor, and E. H. Spafford, “Security Modules for WEB-Based Applications,” *Communications of the ACM*, vol. 44, no. 2, pp. 38-44, Feb 2001.

[3] K. Kang, S. Cohen, J. Hess, W. Novak, and A. Peterson, “Feature Oriented Domain Analysis Feasibility Study,” *CMU/ SEI-90-TR-21*, Nov 1990.

[4] N. Loughran, A. Sampaio, and A. Rashid, “From Requirements Documents to Feature Models for Aspect Oriented Product Line Implementation, ” in *Proc. Workshop on MDD in Product Lines*, Oct. 2005.

[5] H. Ossher and P. Tarr, “Using Multidimensional Separation of Concerns to (Re)Shape Evolving Software,” *Communications of the ACM*, vol. 44, no 10, pp. 43-50, Oct 2001.

[6] I. Ray, N. Li, R. France, and D. K. Kim, “Using UML to visualize Role-Based Access Control constraints,” in *Proc. SACMAT' 04*, Jun. 2004.

[7] R. S. Sandu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-Based Access Control Models,” *IEEE Computer*, vol. 29, no. 2, pp. 38-47, Feb 1996.

[8] P. Wing, and B. O’Higgings, “Using public-key infrastructure for security and risk management,” *IEEE Communications Magazine*, pp. 71-73, Sep 1999.

[9] “情報セキュリティハンドブック, ”電子情報通信学会, オーム社,ISBN:4274079805, 2004.