制作

スケーラブルアプリケーション実現に向けた 設計・実装手法の検討

株式会社NTTデータ

川畑 光平

kawabataku@nttdata.co.jp

開発における問題点

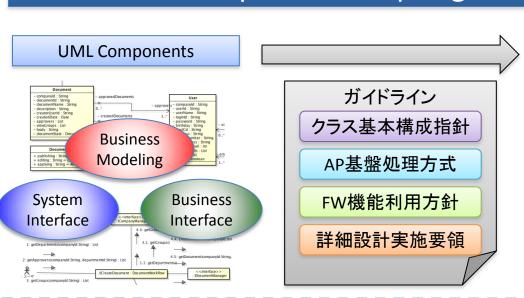
サーバのスケールアウト・縮退を任意に実現可能なシステムアーキテクチャ環境下に適した、スケーラブルアプリケーション設計手法に関する知見が不明確なため、ソフトウェア品質低下や開発コストの増大を招きかねない。



手法・ツールの適用による解決

コンポーネントベース開発手法「UML Components」を用いて設計したアプリケーション設計モデルを、デファクトFWであるSpring Framework及びJPA実装 Hibernateをベースに詳細設計・実装適用するガイドラインを整備し、スケーラブルアプリケーション設計のために必要な拡張を行う。

UML Components – Spring – JPA 適用ガイドライン



- 統一的クラス構成統一的クラス責務
- •再利用性
- •変更容易性

Presentation Layer
Application Layer
Domain Layer
Infrastructure Layer
Layers Architecture



スケーラブル拡張

APサーバに依存しない永続化方式とリクエスト処理の実現 Data Persistence Policy Consistency Level Immediate / RDB Latency / Key Value None / Cache Guide Line C P Categories By Brewer's CAP

評価と今後の課題

[評価]

- ・ガイドライン適用・実装実現により設計の網羅性を担保
- ・ガイドライン適用によりAPアーキテクチャ統一性を担保

[今後の取り組み・課題]

- ・アプリケーション適用後品質のベンチマーク比較
- ・スケーラブルアーキテクチャ上での性能評価
- •Web Service等のスケーラブル拡張問題点の抽出
- 大規模分散処理の適用検証

