

# 2004年第2分科会 (DOA+OOP分科会) 活動報告

2004年12月6日



## 本分科会の目的

- DOA+での分析・設計をJava等のオブジェクト指向言語での実装につなげるための具体的な方法を調査する

**DOA+OOP**



# 分科会活動

- 活動期間 2004年2月～10月（毎月1回）
- 分科会登録メンバー130名
- 事例紹介発表＋1時間以上の質疑応答
  - 株式会社キリンビジネスシステム様
  - メディア情報開発株式会社様
  - 東京国際大学教授 佐藤英人様
  - ウルシステムズ株式会社様
  - 株式会社アルゴシステム創研様
  - セントラル・コンピュータ・サービス株式会社様
  - 株式会社富士通システムソリューションズ様
  - 住友電気工業株式会社様
  - 日本電気株式会社様（BIGLOBE構築運営本部）

（表示順は実施順）



# なぜ、DOA+で設計するか

- 確立した設計技法であり、属人性を排除しやすい
- これまでの標準化、人材、ノウハウの活用  
(OOでビジネスシステムを設計できる人材の不足)
- 上流の設計の方法論と実装技術の分離・独立  
(下流がOOに変わっても上流は変えたくない)
- RDBの性能を最大限引き出せる



# DOA+OOPでの問題点

- DOA+での設計書が下流のJavaプログラマーに伝わらない
- 下流の実装方法にばらつきが出る
- DOA+だけではすべてを記述できない
- インピーダンス・ミスマッチ
  - データ設計
  - ドキュメント(コミュニケーション)



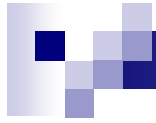
# 解決方法

- DOA+の設計結果をOOPへつなぐ
  - 例) ERD→クラス図、文書によるルール化、ツール利用
  - UML併用によりコミュニケーションを円滑にする
  - DOA+の上流設計を元にしたUMLドキュメントの作成方法のルール化
  - フレームワークやプロトタイプによりドキュメントそのものを削減する



# 解決方法

- 実装場所、実装方法を規定する
  - 文書によるルール化
  - 理論によるルール化
  - R-Oマッピングとそれに基づいた部品
  - ツール(フレームワーク)による部品化によりOO設計を意識させない



# 9つの事例の紹介

## 第1回 DOA+OOP分科会 (2004年2月19日(木)実施)

タイトル	DOA-RAD for Java
講演者	株式会社キリンビジネスシステム様
キーワード	TH法によるDOA-RAD
課題	◆従来のDOA-RADではJavaプログラマーに実装仕様まで規定できない。 しかしながら、上流系はDOA-RADで続けたい
解決策	◆DOA-RAD for Javaを新たに開発。 ◆DOA-RADのアウトプットをUMLにマッピングするルールを作成。 (TH図のクラス図へのマッピング、エラーチェックの実装場所のルール化) ◆JavaコーディングレベルはcFrameworkで標準を作成。大枠を抑えた上で、細かいコーディングはプログラム開発者がコントロール
効果	◆上流工程の設計者はDOA-RADさえ知って入ればよい。 ◆下流工程は外部協力会社に外注できる。

## 第2回 DOA+OOP分科会 (2004年3月22日(月)実施)

タイトル	DOAとOOPの理想の関係 Webtribeアーキテクチャ的視点から
講演者	メディア情報開発株式会社様
キーワード	Webtribe
課題	<ul style="list-style-type: none"><li>◆実装技術に依存しない開発スキル</li><li>◆システム品質と開発生産性の確保</li><li>◆部品の再利用性を追求</li></ul>
解決策	<ul style="list-style-type: none"><li>◆MVC各層個別にフレームワークを作成</li><li>◆粒度の小さいコンポーネントとその実行手順(XML)をリポジトリに登録することで、Javaコーディングレスでのプログラム開発を実現。</li><li>◆DOAのアウトプットに対し、データ項目部品、エンティティ部品、SQL部品、データ制約条件部品を作成。</li></ul>
効果	<ul style="list-style-type: none"><li>◆Javaコーディングを行わないで組み立てられるので生産性、品質を大きく向上できる。</li></ul>

### 第3回 DOA+OOP分科会 (2004年4月22日(木)実施)

タイトル	DOAをベースとする オブジェクト設計の考え方 — データと参照可能性を用いた責務の配置 —
講演者	東京国際大学教授 佐藤英人様
キーワード	RADON法 Responsibility Allocation based-on Data Object and Navigability
課題	◆データに関してはDOAがあるが、プログラムに関してはそのような方法論がなく、処理の実装場所にバラツキが生じ、プログラムの重複が避けられない。
解決策	◆正規化されたデータオブジェクト間の参照可能性に基づいて、オブジェクト間に責務を機械的に配分する方法。関連の特性に着目して参照方向を機械的に決定する。対象範囲はMVCモデルのMの部分。 ◆結果として導き出された実装場所は通常のOOよりもDOAの考え方に近い。データオブジェクトと参照可能性による責務の配置。
効果	◆設計者による実装仕様のバラツキをなくす。

## 第4回 DOA+OOP分科会 (2004年5月28日(金)実施)

タイトル	フレームワークによる エンタープライズアプリケーション構築アプローチ
講演者	ウルシステムズ株式会社様
キーワード	UMLaut ソフトウェアフレームワーク
課題	<ul style="list-style-type: none"><li>◆RDBとオブジェクト指向の両方の技術を適材適所での活用すること。</li><li>◆O-Rマッピング時のインピーダンスミスマッチ解消。</li></ul>
解決策	<ul style="list-style-type: none"><li>◆UMLautソフトウェアフレームワークではRDBの強みを生かす設計に基づいたDAOを提供。</li><li>◆O-RマッピングならぬR-Oマッピング。RDBモデルを基準にオブジェクト実装する。SQLの機能をサポート</li><li>◆開発プロセスはデータモデリングを重視。</li></ul>
効果	<ul style="list-style-type: none"><li>◆RDBMSとオブジェクト指向とのインピーダンスミスマッチをなくし、RDBMSの強みを引き出すことができる。</li><li>◆開発プロセスの安定性確保。</li></ul>

## 第5回 DOA+OOP分科会 (2004年6月22日(火)実施)

タイトル	Hybrid型開発技法 B-DM - オブジェクト指向実装と親和性が高いDOAによる論理構造を導くために
講演者	株式会社アルゴシステム創研様
キーワード	Hybrid型開発技法 B-DM
課題	<ul style="list-style-type: none"><li>◆ビジネスシステム開発で、OOのアプローチによる設計は非常に難しい</li><li>◆バッチプログラム開発の技法がない</li></ul>
解決策	<ul style="list-style-type: none"><li>◆要件定義から論理設計までを、上流はDOA、下流にオブジェクト指向を適用した開発工程を定義。設計情報をメタモデル化し、手順、成果物を規定。要件定義から論理設計までのプロセスと成果物(22種類)を規定</li><li>◆ バッチ系のシステムも重視</li><li>◆ 論理設計で実装場所を規定 (ビジネスシステムテンプレート)</li></ul>
効果	<ul style="list-style-type: none"><li>◆要件定義から論理設計までのプロセス、成果物を細かく定義することで上流での品質を確保。</li></ul>

## 第6回 DOA+OOP分科会 (2004年7月21日(水)実施)

タイトル	システム開発標準におけるDOAとオブジェクト指向技術の活用
講演者	セントラル・コンピュータ・サービス株式会社様
キーワード	開発標準Coup[ku:]
課題	<ul style="list-style-type: none"><li>◆インピーダンス・ミスマッチ:オブジェクト指向技術を前提とする開発では、構造化技法で整理した分析結果を最適な設計結果へマッピングできない</li><li>◆コミュニケーション:開発作業の分業化が増進し、これまで以上にコミュニケーションを重視する開発プロセスの導入が必要になってきた。</li></ul>
解決策	<ul style="list-style-type: none"><li>◆UPとICONIXを参考にした開発プロセス。DOA+OOP。</li><li>◆上流設計(ビジネスモデリング)にXupperを利用。gereXUMLでERDをUMLに自動変換。変換後はKonesaを利用</li><li>◆実装設計のためにロバストネス図、シーケンス図を作成</li></ul>
効果	<ul style="list-style-type: none"><li>◆DOAでの設計結果をオブジェクト設計へ最適なマッピング</li><li>◆表記法の統一によるコミュニケーション向上</li></ul>

## 第7回 DOA+OOP分科会 (2004年8月26日(木)実施)

タイトル	DOA+OOP開発を強力に支援する QuiQpro-Javaご紹介と適用事例紹介
講演者	株式会社富士通システムソリューションズ様
キーワード	QuiQpro開発支援ツール
課題	◆お客様の基幹業務システム開発をWeb化するには大きなリスクが伴なうと共に、開発チームにとって大きな負担がかかる
解決策	◆DOAベースの設計情報(項目設計、エンティティ設計、コード/コンディション設計、入出力情報設計、メッセージ・アクション設計)から、J2EE3層のソースコード自動生成。上流ツールXupper II/ERwin/Data-INTIMATEからQuiQproに設計情報連携が可能。 ◆各種UI部品(カーソル制御、チェック、フォーマットなど)を標準提供。 ◆要件定義のためのプロトタイプ作成機能あり
効果	◆CASEツール、モックアップ機能、プログラム自動生成機能による生産性向上と、設計品質とプログラム品質向上 ◆開発支援ツールによる生産性向上と標準化

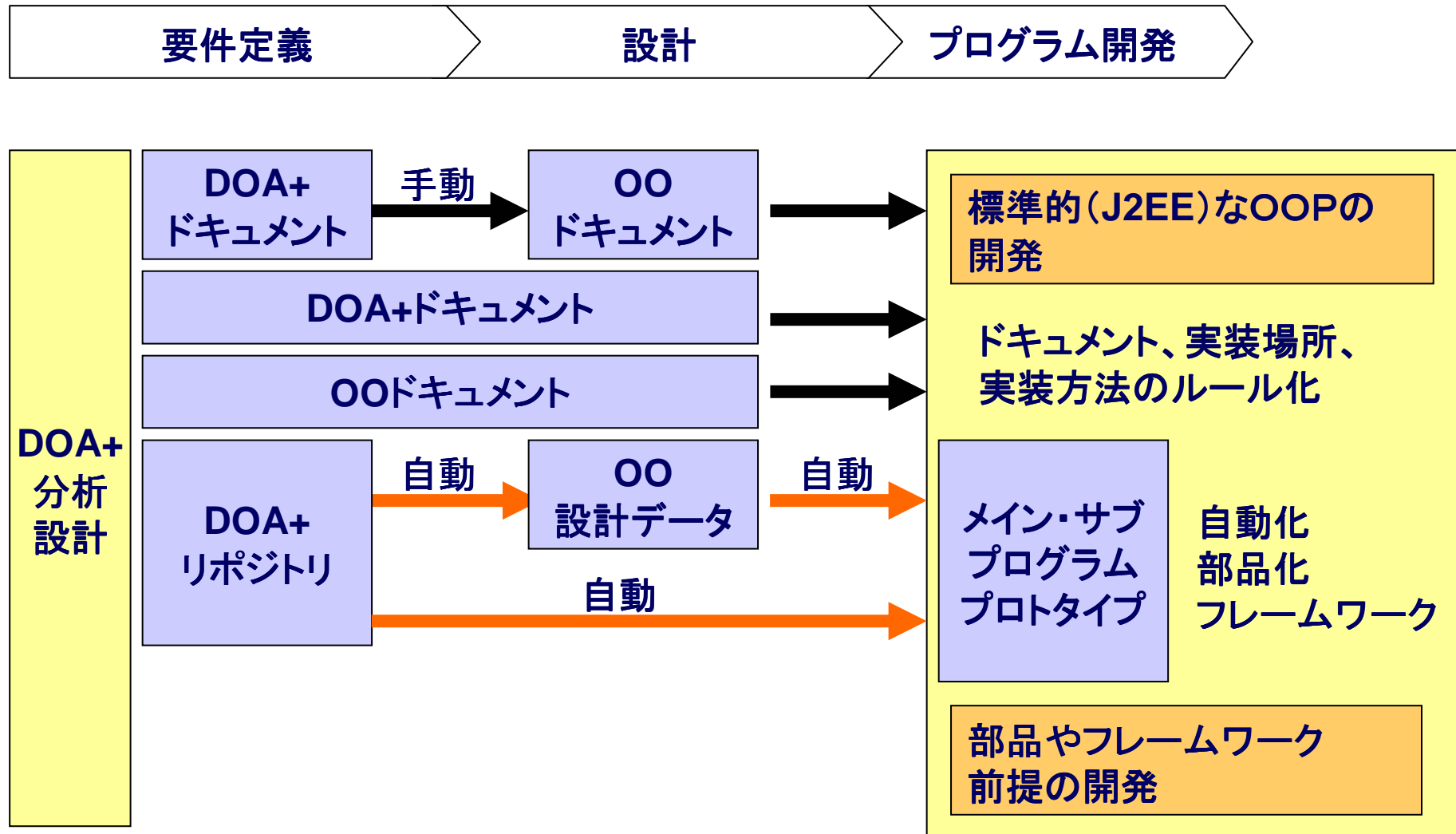
## 第8回 DOA+OOP分科会 (2004年9月28日(火)実施)

タイトル	DOA+ と OOP によるモデル駆動型アプリケーション開発
講演者	住友電気工業株式会社様
キーワード	楽々Framework 開発支援ツール
課題	<ul style="list-style-type: none"><li>◆開発のQCDの向上</li><li>◆エンジニアリング性の高い開発の実現</li></ul>
解決策	<ul style="list-style-type: none"><li>◆DOAによるモデル駆動型開発。DB設計情報から直接プロトタイプを生成</li><li>◆項目オブジェクトでデータ項目の表示属性や入力チェックなどを管理</li><li>◆Javaソースを生成するのではなく、300種類の業務コンポーネント(パターン)を組み合わせて画面入出力処理を自動組立。</li><li>◆T字形ER手法の思想がベースとなっているが、他の手法でも利用可能。</li></ul>
効果	<ul style="list-style-type: none"><li>◆生産性と品質の向上</li><li>◆高度な標準化によりJava初心者でも短期間に使いこなすことができる。オブジェクト指向の専門知識はほとんど不要。</li></ul>

## 第9回 DOA+OOP分科会（2004年10月28日実施）

タイトル	ビジネスルール中心／アーキテクチャ中心による 業務システムのノンプログラミング開発
講演者	日本電気株式会社様(BIGLOBE構築運営本部)
キーワード	BizMAPフレームワーク
課題	◆低コスト、短納期、バグゼロの品質 ◆ビジネスルール記述とソフトウェア実現のギャップ解消
解決策	◆アーキテクチャ中心、ビジネスルール中心の開発手法 ◆ビジネスルールを画面とトランザクションに2領域に分け、各々を、L0:データ項目、L1:データ加工、L2:データ組立、L3:ワークフローの4種類の組み合わせのみで処理を記述。 ◆アプリケーションアーキテクチャが確定しており、6種類のコンポーネントの組み立てで、業務システムを実現する。 ◆開発作業は、このビジネスルールの作成と、コンポーネントへのマッピング作業に置き換えられており、いわゆるプログラミング作業は発生しない。
効果	◆高い生産性、短納期開発とバグのほとんどない品質。

# DOA+OOPのさまざまなつながり方





## まとめ

- DOA+とOOの作業を分離。下流の開発は標準のJ2EEで
- DOA+からOOへ自動変換。下流の開発は標準のJ2EEで
- DOA+からJ2EEを意識せずに下流の開発へ

# 結論

- ・ビジネスシステムが台帳システムである限り
  - ・ビジネスシステムがRDBを利用する限り
- 上流設計工程にはDOA+か、  
少なくともDOA的なアプローチが必須となる。

DOAとOOPをうまく結びつける工夫は必要だが、  
解はいくつもある。

