

Webシステム開発基盤 「AZAREA-Gene」 ～自動生成を効果的に活用した 知識集約型開発基盤について～

ICT営業本部
サービスプロデューサー

井場 辰彦



1. はじめに

システム開発プロジェクトは年々、短納期化・低コスト化が進んでおり、もはや属人的な能力やプロセス的な工夫「のみ」では限界に近付いているといっても過言ではない。また、従来の役務提供型の受託開発は熾烈な競争にさらされており、ゼロスタートからのフルスクラッチのみでは競争力を維持することも難しく、武器となる道具を持った、新しい形でのシステム開発が模索されている。こういった状況から、開発・テストなどの自動化を積極的に活用した、他社との差別化・競争力の源泉となり得る、当社の知識・ノウハウが集約された開発基盤が必要とされている。

本稿では、上記背景をもとに製造された当社独自の開発基盤「AZAREA」シリーズの中から、Webシステムの開発基盤である「AZAREA-Gene」を紹介する。AZAREA-Geneがどのような背景・コンセプトのもとに開発され、どのような機能を持っているのか、今まではイメージが掴めていなかった方にも理解していただければ幸いです。

2. 背景

当社では以前から、当社の主力業務である業務システムを構築するための「典型的なアーキテクチャ」(以降、典型的アーキテクチャ)の整備に注力してきた。業務システムに必要なとされる機能は、その業務に固有の機能のみではなく、ログイン・ログアウトや定型的な単項目チェック機能(必須チェック、桁数チェック、etc.)など、どのような業務システムでも必要とされる共通的な機能も多い。そういった共通的な機能を、案件ごとに一から構築することは非常に非効率であり、かつ品質のばらつきも多い。それらの機能や使い方、典型的なモジュール構造(狭義な意味でのアーキテクチャ)を事前に整備し、開発基盤という形でまとめたものが、典型的アーキテクチャである。こういった典型的アーキテクチャは、実際にいくつ

かの案件に導入され、生産性や品質の向上に一定の寄与を果たしてきた。そういった背景の中で、さらなる機能性・生産性の向上を目指し、経営戦略「知識集約型企業への変貌」に基づいて、知識集約の基盤として典型的アーキテクチャを進化させたものが、AZAREAシリーズおよびAZAREA-Geneである。

AZAREA-Geneは、二大プラットフォームであるJava/.NETの両方に対応し、自動生成を活用した生産性の向上と、差別化につながる多くの組み込み機能を持つ。3章では、AZAREA-Geneの説明に先立ち、これまでの自動生成における課題を述べる。4章では、それらの課題を踏まえた、AZAREA-Geneの自動生成に対する基本的な仕組み・コンセプトについて説明し、5章にてAZAREA-Geneの機能を紹介する。最後に、6章にて、これまでのAZAREA-Geneの導入事例を紹介する。

3. これまでの自動生成の課題

これまでもいくつかの自動生成ソリューションが世の中に登場してきたが、業務システム、特に日本の企業においてはあまり活用が進んでいないのが実情である。低コスト、短納期化の流れは確実にあるものの、それでもシステム当たりの平均開発期間は海外の企業に比べて圧倒的に長く、さほど自動生成が寄与していない傾向が見受けられる。

そこで、これらの実情も踏まえ、当社ではまずこれらの自動生成、中でも「100%自動生成」が何故、日本の企業において活用が進まないのか、検討を行った。それらは、100%自動生成を実現するために、そのプラットフォームの通常の開発言語(Java、.NETなど)ではなく、独自の簡易言語を採用する傾向にあるのだが、確かに一定の範囲内において100%自動生成が実現できる反面、主に以下のような課題が存在することが挙げられた。

●デバッグが困難

開発時には独自言語を使用するが、実際に動作する際には、通常の開発言語に変換されるケースが多い。そのため、独自言語上のどの部分が動作上問題になっているのか、突き止めることが困難な傾向にある。

●初期設定量が多い

実際に画面が動き出すようになるまでの初期設定量が多く、わずかの設定ミスで、動作しないこともある。「デバッグが困難」の問題と併せ、実際に動くようになるまでのハードルが、意外に高い。

●プラットフォームの機能に利用制限がある

通常の開発言語を利用していないため、当然、独自言語がカバーしていない範囲の機能は、利用できない。業務システムは、特殊なプロトコルや他システム連携を利用しているケースも多く、自動生成の対応範囲から「外れる」レベルのカスタマイズに対して、どれだけ柔軟に対応できるかが問題となる。「多少生産性は落ちるが、利用自体はできる」レベルと、「そもそも利用そのものができない」レベルでは、大きな差が生じる。

上記のような課題を踏まえ、次章以降にて、AZAREA-Geneの基本的な仕組みおよび各種機能について述べる。

4. AZAREA-Geneの概要

AZAREA-Geneは、前章で述べた一般的な自動生成の課題を踏まえ、既存の自動生成ではなく、独自のコンセプトを持った、独自の自動生成をベースに構築されている。本章で

は、その概要として、自動生成の基本的な仕組み・コンセプトについて述べる。

4.1 初期設定

AZAREA-Geneにおいては、実際に動作する画面を作成するまでの初期設定量を可能な限り減少させるため、モデル駆動型アーキテクチャ(MDA: Model-Driven Architecture)的な要素を取り入れている。すなわち、ある機能を作成する際に、まずはその機能で使用するデータ構造(テーブル、ビュー、クエリ、ワークテーブル、etc.)を定義し、定義したデータ構造に基づき、一通り動作する標準の初期設定画面を生成する。データ構造の定義は、AZAREA-Geneに付属のツールを用いることにより、.NET/Javaによらず、同様に定義することが可能である。

4.2 設定変更その1・UI設定

初期設定からの設定変更のうち、画面に関する設定の変更は、「UI設定」を用いて行う。不要な初期画面の削除や必要なアドオン画面の追加、画面レイアウトの変更や各種入力コントロール(テキストボックス、ドロップダウンリスト、etc.)の種類変更・設定値変更など、画面に対する一通りの設定変更は、UI設定を用いて行うことができる。これにより、「初期設定」にて定義したデータ構造に対して、非常に自由度の高い画面生成を行うことが可能となる(図1)。

4.3 設定変更その2・更新設定

初期設定の時点においては、データベースの物理的なテーブル構成と、画面で扱う論理的なデータモデルの構成は結びつきが強く、あるテーブルに対して、一つの編集画面が設定されている状態となっている。しかし、実際の業務システ

図1 UI設定



ムにおいては、物理的なテーブル構成と論理的な画面データモデルとの間にずれがあるケースも多く、画面の見た目上1テーブルを扱っているように見えるが、実際のデータベース上では、複数のテーブルに分割されているようなケースも多い。

このような場合、AZAREA-Genieにおいては、「更新設定」を用いることにより対応が可能となる。「初期設定」にて、Joinなどのクエリーを使用して複数テーブルをもとにデータ構造を定義し、更新設定を用いて、データ構造中の各項目を、物理的なテーブルのどの項目に反映するかを設定することにより、物理的なテーブル構成によらない、柔軟な画面構成を取ることが可能となる(図2、図3)。

4.4 設定変更その3・エンティティ設定

個別の画面ごとの設定ではなく、データ構造そのものに対する設定変更は、エンティティ設定を用いて行うことが可能である。例えば、ある項目に対する単項目チェック(桁数チェ

ック、範囲チェック、etc.)や、キー項目のオートインクリメント設定などが可能である(図4)。

4.5 まとめ

上記のように、AZAREA-Genieにおいては、MDAの要素を取り入れることにより、必要な初期設定の量を可能な限り減少させることに成功している。また、UI設定・更新設定・エンティティ設定の3種類の設定を用いることにより、初期設定からの柔軟な設定変更が可能である。加えて、それらの設定内容から、(独自行語ではなく)Java/.NETなどの通常のプログラミング言語のソースコードを、可読性を考慮した、カスタマイズ可能な形で生成することにより、デバッグ容易性・プラットフォームの機能の利用容易性を確保している。

このように、AZAREA-Genieでは、無理に100%自動生成を目指すのではなく、業務システムにおいては必ずカスタマイズが必要であることを前提に、生産性とカスタマイズ性のバラ

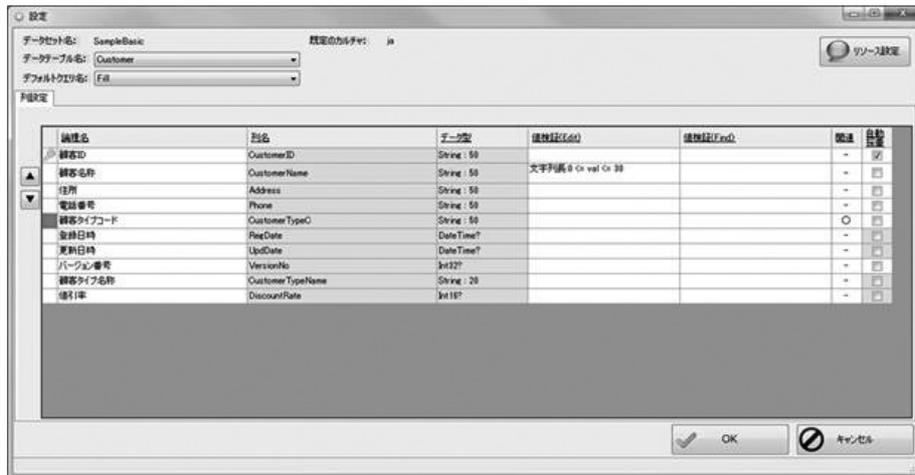
図2 論理データモデルと物理テーブルのマッピング設定(テーブル単位)



図3 論理データモデルと物理テーブルのマッピング設定(カラム単位)



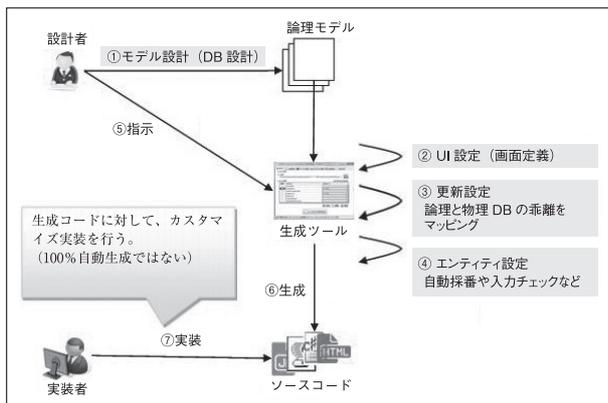
図4 エンティティ設定画面



ンスが取れた、実用的な自動生成を実現することを、その基本コンセプトとしている(繰り返すが、無理に100%自動生成を実現することは、AZAREA-Geneのコンセプトではない)。

以下に、基本的な自動生成のイメージを示す。

図5 自動生成のイメージ



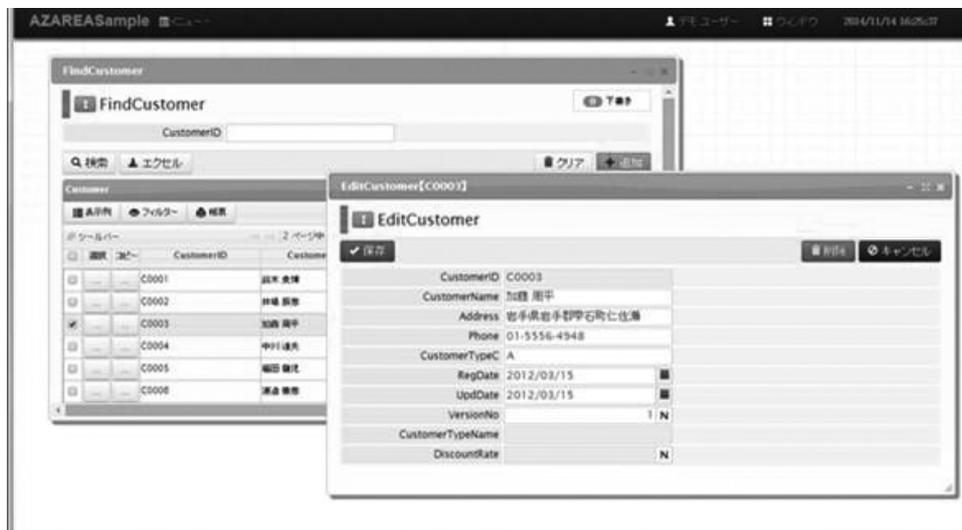
5. AZAREA-Geneの機能説明

前章までのAZAREA-Geneの概要をもとに、本章では、AZAREA-Geneの持つ代表的な機能について、述べる。

5.1 画面の基本機能

AZAREA-Geneにて生成される画面は、HTML5の機能も活用しており、単純なWebシステムとは一線を画する、高機能な作りとなっている。代表的なものが「マルチウィンドウ」機能であり、1ページの中に、複数のウィンドウを同時に表示することが可能となっている(図6)。例えば、複数の検索結果を同時に参照しつつ、複雑な伝票の登録を行うといったようなことも可能である。また、スマートデバイスにも標準で対応しており、PC/タブレット両用の画面も、効率的に作成することができる。その他、設定・カスタマイズにより、様々な機能が利用可能であるが、一部は次節「初期設定状態における標準機能」にて、残りは「その他の主要機能」にて、順次述べる。

図6 AZAREA-Gene標準画面例



5.2 初期設定状態における標準機能

初期設定状態においては、データ構造ごとに、検索画面と編集画面が1対ずつ生成される。「受注情報-受注明細」のように、明細情報を持つデータ構造に対しては、各明細ごとの編集画面も、併せて生成される。このように、あるデータ構造に対して1対で生成される検索画面・編集画面・明細編集画面を通して、検索～一覧表示～データ選択～登録・更新・保存・削除の一通りの操作を行うことが、初期設定状態で、可能である。

それに加え、検索画面においては、検索結果のページング・ソートや表示列のカスタマイズ、検索結果のExcel形式でのダウンロードや簡易帳票の出力、検索条件(いわゆる、SQLのWHERE句)の動的生成などが利用可能である。

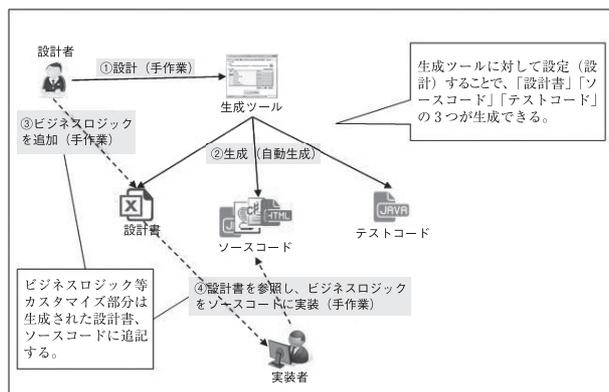
また、編集画面においては、親子データの一括更新や既存データのコピーによる新規登録、未登録のデータの下書き保存・破棄確認、楽観的排他制御などの機能が、標準状態で利用可能である。

このように、AZAREA-Geneにおいては、初期状態でも比較的高機能な画面が自動生成され、これに対して金額計算、整合性チェックなどのビジネスロジックをアドオンしていくことが、標準的な開発スタイルとなる。

5.3 設計書作成・単体テストサポート機能

AZAREA-Geneにおいては、画面を自動生成するための設定情報をもとに、画面のみではなく、生成された画面を自動打鍵するためのテストサポート機能や、設定情報をExcel形式に変換した、画面設計書の一部として使用可能なドキュメントの生成を行うことも可能である。現状は、あくまでも「サポート機能」であるため、自動生成のみで完全な設計書が生成されたり、テストが完全に自動化される訳ではないが、それでもすべてを手作業で行っていたこれまでと比べると、大幅な効率化が可能である。このように、プログラミングのみではなく、設計書作成やテストといった他の工程においても、自動生成と手作業を効果的に組み合わせ、柔軟性とのバランスを取りつつ作業を効率化できるところが、AZAREA-Geneの魅力の

図7 設計書、プログラム、テストの関係



一つである。

5.4 その他の主要機能

AZAREA-Geneには、上記以外にも様々な組み込み機能が存在する。本節では、これまでに紹介した機能も含め、主要な機能を一覧にて紹介する。

表1 AZAREA-Geneその他機能

機能名	説明
CSVサポート	検索画面・編集画面生成時に、その更新処理に対する、CSVによる一括アップロード・ダウンロード機能も同時に生成される。
カメラ連携	タブレット(iPadなど)のカメラ機能と連携し、撮影した写真をその場で補正・登録することが可能。
Google Map連携	タブレットの位置情報機能と連携し、Google Mapに位置情報を表示・取得することが可能。
添付ファイル	任意のテーブルデータに対し、任意のファイル(担当者間の引継ぎ情報や契約情報(PDF)など)を添付することが可能。
ギャラリー	添付ファイルを画像データ向けに拡張した機能。商品画像等を、サムネイルなどを用いてギャラリー表示することが可能。
下書き保存	登録途中のデータの一時保存・呼び出しが可能。
コピー新規	類似したデータをコピーし、新規にデータを登録することが可能。
自動採番	主キー情報の自動採番機能を提供。
変更検知	未保存のデータが存在する画面をCloseしようとした場合に、その旨をアラート表示することが可能。
ログ出力・ログ管理	オペレーションログの取得機能および取得したログのビューアーを提供。

6. 導入事例

AZAREA-Geneは、既にCAC社内外のソフトウェア開発プロジェクトへの導入が進んでおり、その効果を発揮しつつある。そういった事例の中から、本章では、CACグループ企業A社のパッケージ開発について、紹介させていただきたい。

CACグループ企業A社は、業界で多数の導入実績を持つC/S型パッケージソフトの開発・販売を、事業の柱の一つとしている。現行のパッケージはリリースから長い期間を経ており、今後アーキテクチャが陳腐化することが懸念され、機能追加やメンテナンス性にも問題が生じつつあった。

今後のビジネス展開を考慮すると、C/S型のままではタブレットを始めとする多様なデバイスへの対応が難しく、Webシステムとして刷新するとの方針が決定されたため、従来型のフルスクラッチ開発と、AZAREA-Geneを導入し、自動生成を活用した新たな開発手法を比較検討することとなった。

長年にわたって機能追加が行われてきた既存のソフトウェア資産は、従来型のフルスクラッチ開発では数億円規模の開発費用が必要になるとの見積りが出た一方、AZAREA-Geneを用いた開発では、主要な機能のプロトタイプを作成して開発生産性を評価した結果、大幅に開発費用の低減が実現可能であるとの結論に達し、A社ではAZAREA-Geneを採用することとなった。単純な開発生産性の他にも、AZAREA-Geneは、自動生成を活用しつつもカスタマイズ性とのバランスもとれており、パッケージソフトに求められる要件を実現できる見込みが立てられた点も、大きな評価ポイントとなった。

そのパッケージソフトは現在開発中であるが、まもなくシステムテストに入る段階にあり、開発は順調に進んでいる。これは、大規模なソフトウェア開発プロジェクトにもAZAREA-Geneが適用可能であることの一例であり、こうした事例により、AZAREA-Geneは今後、CAC単体のみではなく、CACグループ全体での効果的な活用が望める段階となった。

7. おわりに

自動生成は、その特性上、どうしても「向き」「不向き」という

ものが存在する。残念ながら、現状はAZAREA-Geneにおいてもそれは例外ではなく、特に、初期設定状態である程度動作する画面を生成することが可能であるという性質上、初期設定に近い動作の画面を作成する場合と、初期設定と全く異なる動作に大幅に設定変更・カスタマイズする場合で、実際以上に「落差」を感じてしまうケースもあるかもしれない。AZAREA-Geneは、カスタマイズ性の確保を視野に入れた開発基盤ではあるものの、やはり、すべてをカスタマイズしてしまえば、自動生成の恩恵が薄れてしまうことも事実である。よって、現状では、可能な限り多くの画面を初期設定の動作にあわせつつ、本当に重要な少数の機能に絞ってカスタマイズを行うといった利用方法が、望ましい。これまでのシステム開発では、重要な機能もそうでない機能もおしなべて注力しなければならなかったが、自動生成を活用したシステム開発ではこのように、非重要機能は徹底的に自動生成を活用して省力化し、その分を重要機能に注力するといった、メリハリのある工数配分ができることも、魅力の一つである。そういったメリハリを利用し、限りあるIT予算の最適化とITに対する投資効果の最大化を実現することの一助に、AZAREA-Geneが貢献できれば幸いである。