

# IBMの次世代超高速開発

## 自動化ソリューションを活用したアプリケーション開発の高速化

IBMでは自動化ソリューション、AIやデータ分析技術を活用し、システム開発全体の高度化を推進しています。本稿では、その取り組みの全体像を紹介し、プロジェクト管理・設計・開発・テストの各局面における自動化の取り組みとそれにより得られる効果を解説します。

### ▶▶ 1. 近年のシステム開発の潮流

ビジネス活動の推進にITの活用が不可欠となっている現在、いかに早く新しいビジネス変革をユーザーに届けるかが多くのITシステムやITプロジェクトの課題となっています。これまでの多くのITプロジェクトは、ウォーターフォール開発で進められ、特にその上流工程では多くの時間と工数がOA製品による設計文書の作成に費やされてきました。しかし、このようにして作成されてきた設計文書は、ソースコードなど開発の成果物とは直接ひも付かず、次第にメンテナンスされなくなることが多くあります。このような背景から、設計成果物は最小限にとどめ、動くコードを最重視する高速開発やアジャイル開発が注目を集めています。アジャイル開発の本質は、Minimum Viable Product(以下、MVP)を早くエンドユーザーに届けることで、適切なフィードバックを早く得ることにあると考えています。MVPには、「ユーザーに価値を提供できる最小限度の製品」という意味があり、Frank Robinson氏により提唱されリーン・スタート

アップ活動の中で有名になりました[1]。

このようなアジャイル開発と親和性が高いのが高速開発ツールです。高速開発ツールの特徴は、ツール上で定義した画面やテーブルなどの設計情報を基にソースコードを生成するか、定義内容を解釈するエンジンによって直ちに動作させることができることです。そのため、すぐに実機による動作確認をすることができます。

ただし、実際のシステム開発の現場では、高速開発ツールの適用だけで開発期間の短期化が実現できるわけではありません。どのような企業システムにも競争相手との差別化を図るための独自のビジネス・ルールや業務処理があります。このような競争力の源泉となる各社固有の要件は、高速開発ツールの標準機能で実現することが難しい場合が多く、各ツール特有の深いカスタマイズ・スキルが必要で、そうしたスキルを持つ人がいないというボトルネックが開発期間の長期化やコストの増加につながります。また、独自のカスタマイズやアドオンが増えれば、その品質を担保するために追加のテストを行う必要があり、本番リリース前の統合テストやリグレッ

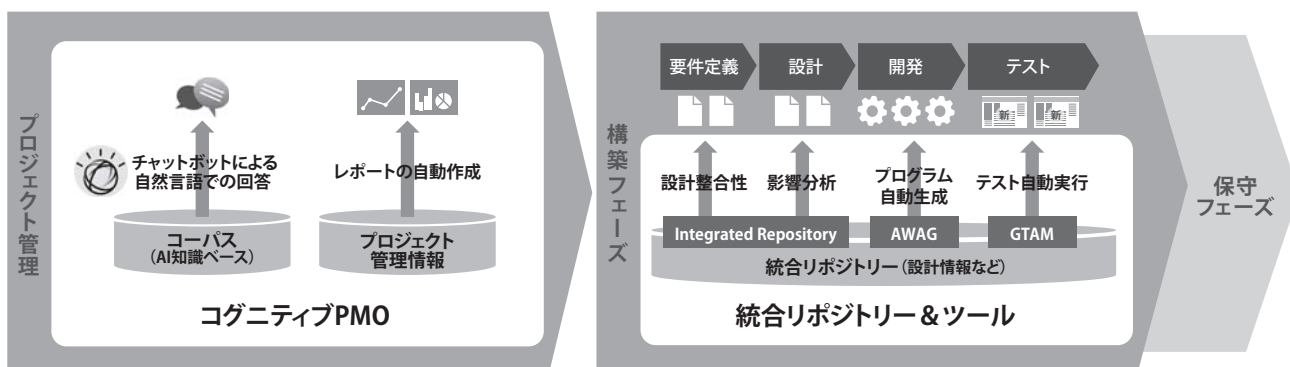


図1. IBMの次世代超高速開発の全体像

ション・テストの工数なども必要となります。さらには、ある程度のプロジェクト規模になるとプロジェクト管理や報告にかかる工数も大きくなり、アプリケーションの設計開発だけを高速化してもプロジェクト全体の工数を少なくすることが難しくなることがあります。

そこで、IBMでは次世代超高速開発としてアプリケーション開発だけでなくプロジェクト管理の領域も含めたシステム開発のライフサイクル全体を対象に、AIや自動化の仕組みを積極的に取り入れたシステム開発を推進しています。「コグニティブPMO」と「統合リポジトリ&ツール」の2つのソリューションです(図1)。

コグニティブPMOは、プロジェクトの管理やサポートを横断的に行うPMOの作業に「IBM Watson」や分析基盤を適用することでプロジェクト管理作業の自動化を支援し、必要な人的資源を本来実施すべきアクションの立案や設計開発にあてることを可能にします。

一方、統合リポジトリ&ツールは、設計情報をリポジトリ情報として一元管理することで整合性を確保した設計を進めるとともに、自動化テクノロジーによって整合性の取れた設計情報からプログラム・ソースコードの自動生成、さらにはテスト自動化シナリオ作成の支援を行うことで、高速で高品質なアプリケーション開発を推進することを目的としています。

次章以降で、コグニティブPMO、統合リポジトリ、統合リポジトリと連携する自動化ツールについて詳細を

解説します。

## 2. コグニティブPMO

コグニティブPMOは、プロジェクト運営において発生するPMO担当者への多様な問い合わせを肩代わりするチャットボット機能と、プロジェクトの状況を適切に把握するためのレポート自動作成機能を提供しています(図2)。

チャットボット機能は、IBM Watson APIを利用し、多彩な自然言語での問い合わせ文を学習し、質問の意図を理解した上で意図に合った回答を返します。基本となる質疑応答の内容として、プロジェクトの入退館申請、ユーザーID申請などプロジェクト運営で頻繁に発生する質疑応答集から、順守すべき社内プロセスや規定、プロジェクト・マネジメント領域の知識全般に至るまでの質疑応答集を提供しています。チャットボット機能を導入することにより回答作業が軽減し、PMOやリーダーはプロジェクト全体の品質向上などの作業に専念できるようになります。また、課題管理や欠陥管理をはじめとするプロジェクト・マネジメント領域全般の知識体系への回答を提供することで、組織やプロジェクトにおけるプロジェクト管理や品質管理の標準化を推進します。

レポート自動生成機能は、従来PMが人手により実施していたデータの収集・集計とレポートの作成を自動化します。WBS(Work Breakdown Structure)、成果物一覧、欠陥一覧、レビュー指摘一覧などのプロジェク

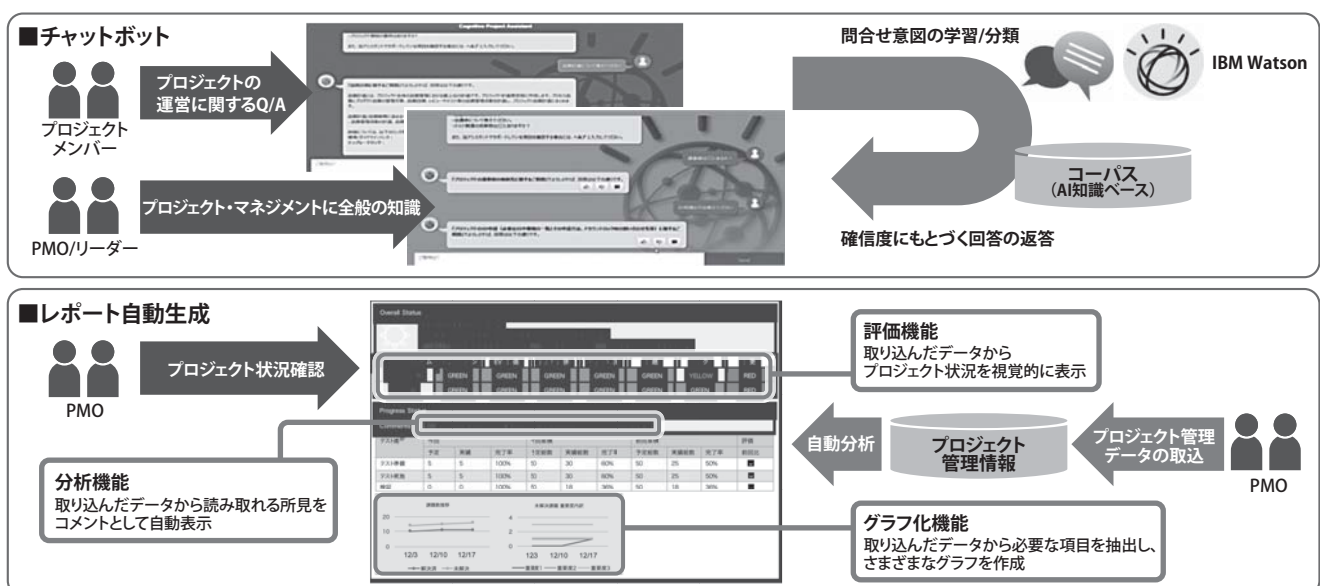


図2. コグニティブPMOの仕組み

ト管理資料を入力して、プロジェクト全体の品質や進捗のレポートをサマリー・レポート、詳細レポートとして可視化します。人手によるプロジェクトの状況報告資料作成は、どうしても人間による恣意的な判断が入ってしまい、適切な報告がタイムリーに行われず課題への対応が遅れてしまうという問題につながることが多くありました。レポート作成自動機能では、適切な閾(しきい)値を定義することで分析機能により機械的に品質や進捗の状況を判定します。このため、公正なプロジェクト状況報告がされることにつながり、適切なタイミングでより具体的なアクションが取れるようになります。

コグニティブPMOの導入により、IBMではPMやリーダーが管理作業ではなく、プロジェクト状況の分析やアクションの立案など本来時間をかけるべき作業にシフトできるようになっています。

### ▶▶ 3. 統合リポジトリ&ツール

統合リポジトリ&ツールは、設計・開発・テストをカバーする単一のツールではなく、独立して利用可能なコンポーネントが設計情報を通して連携する構造を採用しています(図3)。「Integrated Repository」はその中核をなすコンポーネントであり、アプリケーションの設計情報を保持するリポジトリとして求められるさまざまな機能を提供するWebアプリケーションです。特定の

実装技術やフレームワーク・製品を前提としない論理レベルの設計情報を保持する基本機能とともに、その情報をプラグイン機構によって自由に再利用可能とする基盤を提供します。

Integrated Repositoryは、大別して図4の機能を提供します。WebブラウザによるUIに加え、すべての機能はREST APIとして呼び出すことが可能であり、他ツールへの連携はもちろん、継続的インテグレーション(CI)ツールをはじめとした各種自動化ソリューションへの組み込みを容易に実現できます。

各機能の概要は表1に示すとおりです。データモデル管理機能、アプリケーションモデル管理機能で定義されているモデルは、柔軟な属性追加を可能としつつ、定められた構造によって参照関係を認識できるように定義されており、その検証を行うことでモデルつまり設計情報間の基本的な整合性を維持します。加えて、お客様の要件に固有のさまざまなルール(例:命名規則など)は、それらをツールで実現可能なようにプラグイン機構が用意されているので、登録時や定期的なチェックを組み込むことが可能です。

このように、Integrated Repositoryはオンプレミスをはじめクラウドおよびコンテナ環境で動作するWebアプリケーションとして作成された設計支援ツールです。既存のツールやソリューションでは適用が難しい場合でも、

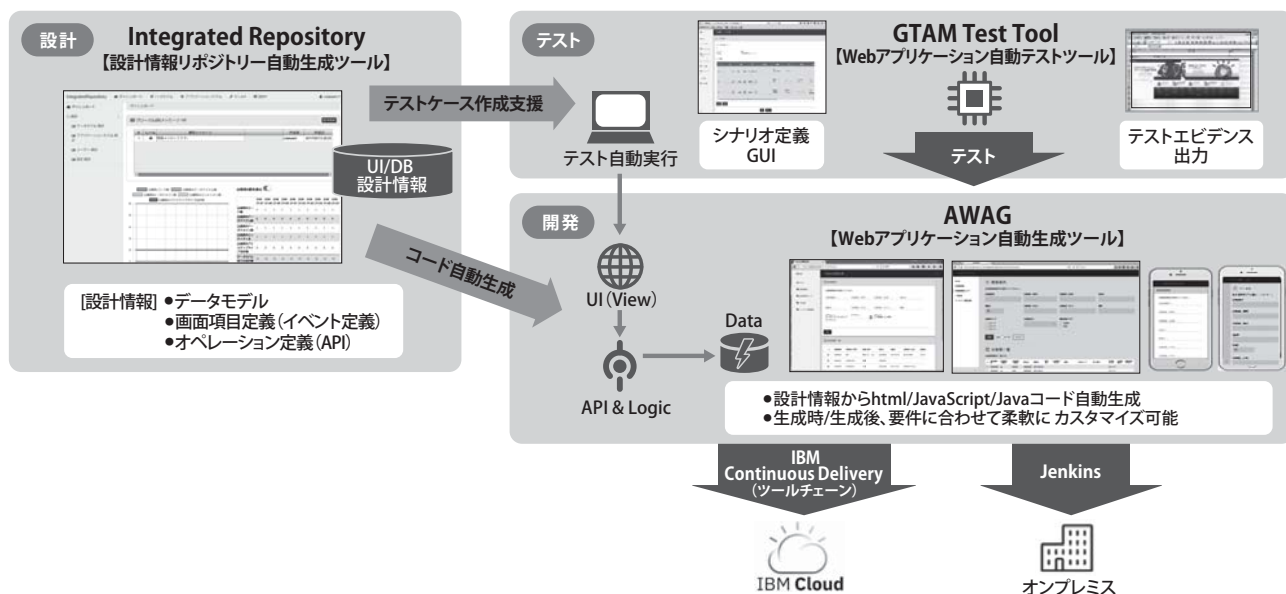


図3. IBMの次世代超高速開発における統合リポジトリ&ツール



ツール基盤機能の上でプロジェクト独自の機能を追加・拡張できることで、さまざまなプロジェクトでその適用方法が検討可能であり、各種自動化ツールとの連携を容易に実現します。

## ▶▶ 4. Webアプリケーション自動生成

「Automated Web Application Generator」(以下、AWAG)は、Integrated Repositoryで定義された設計情報からWebアプリケーションのソースコードを生成する機能を提供しています。AWAGにより生成されるアプリケーションの特徴として、次の2点があります。1点目は、人手によるカスタマイズと保守を考慮に入れた保守性の高いソースコードを生成することです。2点目は、モバイルやクラウドとの親和性の高いアプリケーション・アーキテクチャーを採用していることです(図5)。

本稿の冒頭でも述べたとおり、設計情報から直ちにアプリケーションを実行可能な状態として提供することは開発の高速化に欠かせません。この実行可能なアプリケーションの形態として、設計情報を解釈して製品固有のランタイム上で動作するランタイム型のツールと、設計情報からソースコードを生成するソースコード生成型のツールが存在します。ランタイム型のツールは、ビルドやデプロイなど実行環境で稼働させるまでの間に余計な作業がなく最も高速に稼働確認ができますが、複雑な要件には高難

易度のカスタマイズが必要となるなどの課題もあります。このため、AWAGではソースコード生成型のアプローチを採用し、プロジェクトごとの複雑な要件には一般的なWebアプリケーションの開発知識で柔軟に対応できることを目指しています。IBMのサービス部門でカスタム開発を行う場合、アーキテクトが代表的なアプリケーションのパターンを分析し、カスタマイズ性も考慮したソースコードのアプリケーション構造を決定し、プロジェクト標準となるアーキテクチャー・テンプレートとして開発ガイドやサンプル・アプリケーションを作成することが一般的です。AWAGでは、実際のプロジェクトで作成された実績のあるアーキテクチャー・テンプレートを採用し、この標準を参考にコード生成を行うことで、ソースコード・レベルでの可読性とカスタマイズ性を確保しています。

AWAGで生成されるWebアプリケーションは、クライアントサイドにはシングルページ・アプリケーション(Single Page Application、以下SPA)、サーバーサイドにはREST APIとして実装されるアプリケーション・アーキテクチャーを採用しています。SPAは、画面表示に必要なソースコード(HTML、JavaScript、CSS)を単一のページの資源として最初に読み込むことで、操作性や応答性などユーザー体験の改善を狙ったアーキテクチャーです。このため、サーバーサイドはHTMLなどレイアウトに関する情報の生成は不要で、必要な業務データのみを処理

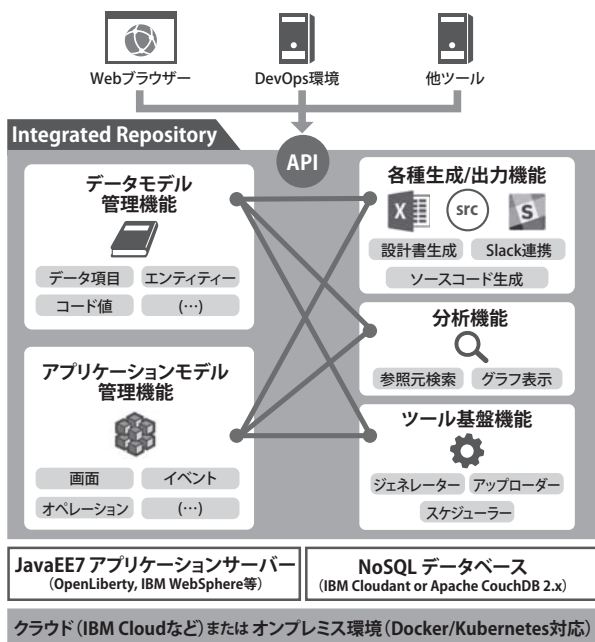


図4. Integrated Repositoryの提供機能と動作構成

表1. Integrated Repositoryの機能

	機能名	説明
1	データモデル管理機能	データ項目を表現する「データアイテム」や、それらをまとめた「エンティティ」(DBのテーブルに相当)などをモデルとして登録します。これらを総称してデータモデルと呼称しアプリケーションモデルから参照利用します。
2	アプリケーションモデル管理機能	画面やコンポーネント、オペレーションといったアプリケーションの構成要素をモデルとして登録します。
3	ツール基盤機能	登録したデータを利用して何らかのデータやファイルを生成する「ジェネレーター」や、ファイルを受信して処理する「アップローダー」、定期的な処理を実行する「スケジューラー」が動作する基盤(プラグイン機構)を提供します。
4	各種生成/出力機能	ツール基盤機能の上で実装された部品群で、設計書やソースコードなど、登録データの二次利用に関する機能を提供します。
5	分析機能	登録されたデータの関係を元に、その依存関係を表や関係図として可視化する機能を提供します。

するシンプルなREST APIとして実装することが可能です。また、サーバーサイドの処理をクライアントのUIに依存しないREST APIとして提供することにより、APIをWebアプリケーションだけでなく、モバイル端末やタブレット端末などのさまざまな機器からも利用できます。

このようなアプリケーションの構造は、近年注目されているマイクロサービス・アーキテクチャーとも合致します。マイクロサービス・アーキテクチャーとは、Martin Fowler氏とJames Luis氏によって提唱された考え方[2]で、アプリケーションを独立したプロセスで動作する小さなサービスの組み合わせとして開発するという考え方です。小さなサービスがMVPに相当し、独立して開発とリリースを行うことで、全体を一枚岩のモノリシックなアプリケーションとして開発する場合と比較して、開発のスピードアップを期待できます。また、この小さなサービスが独立して稼働するという考え方がコンテナ型の仮想化技術とマッチしているため、システム開発のクラウド化、オンプレミスとクラウドとの相互可搬性を高めるアプローチとしても注目されています。このような背景から、AWAGはアプリケーションのマイクロサービス化、クラウド・ネイティブ化を実現するための手段としても利用されています。

## 5. Webアプリケーションテスト自動化

「Guide and Toolkit for Application Modeling」(以下、GTAM)は、多くのテスト自動化事例から得られた

知見を基に開発されたテスト・フレームワークおよびツール群で、Webアプリケーションの画面テストを自動化し、効率よくテストを実施するための機能を提供しています。

通常、画面を自動で打鍵させるには画面の操作手順を定義したテスト・スクリプトを作成する必要がありますが、ほとんどの場合Javaなどのプログラム言語で書くことになり、テスト・スクリプトの作成コストや要求スキルが高くなります。また、テストでは同じ画面に対して何パターンもの入力値をチェックすることが一般的ですが、その入力値をテスト・スクリプトの中で記述しようとすると、同じような処理をパターンの数だけ用意する必要があり、保守コストも増大します。

そこでGTAMでは、テスト・スクリプトをプログラム言語で書くのではなく、ExcelまたはWebアプリケーション上でテスト・シナリオを定義できるようにすることで、テスト自動化のスキル要件を下げ、保守性を高めています。また、テスト・スクリプトと入力データの定義を分離することで、複数のデータ・パターンを定義したデータ・ファイルを作成するだけで何パターンものテストが実行できる仕組みを採用し、テスト・スクリプトの作成・保守コストを低減しています(図6)。

GTAMはIntegrated Repositoryと連携してその設計情報を参照して選択形式で画面の操作内容を定義したり、画面項目情報を基に入力データのひな型を自動生成したりする機能を提供しています。このことで、設計情報と整合性のあるテスト・スクリプトを効率よく作成す

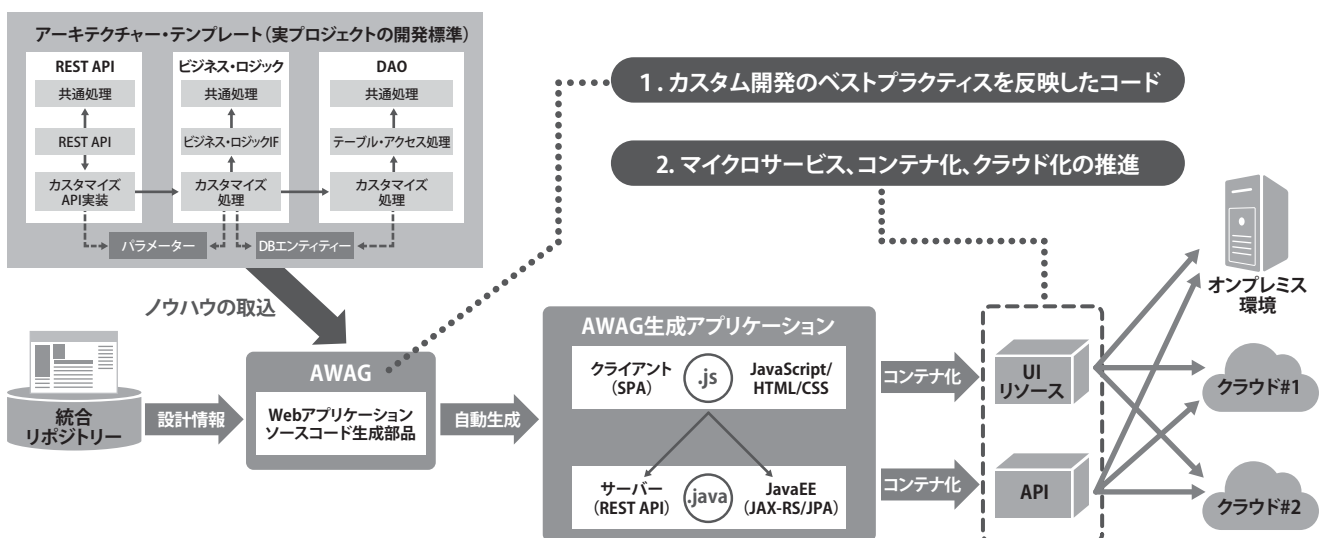


図5. AWAGを活用したコード生成の狙い

る仕組みを提供し、テスト・スクリプトの作成負荷をさらに軽減できるようになります(図7)。

また、実際のプロジェクトでは第三者によるテスト結果の検証のためテスト実施時のエビデンスの提供を求められることが多く、テスト・エビデンスを手動で作成しているとテスト自動化の効果が薄くなってしまいます。GTAMでは、自動鍵とともに画面の操作エビデンスを自動生成する機能を提供し、エビデンスの手動作成作業の排除、エビデンスの統一性によるレビュー担当者の負荷軽減につなげています。

## 6. おわりに

本稿では、IBMでのシステム開発における自動化の取り組みとして、IBMの次世代超高速開発、コグニティブPMOおよび統合リポジトリ&ツールで提供される機能について紹介しました。今後もIBMでは、自動化や分析機能の拡充によるシステム開発のさらなる品質強化と自動化を検討しています。Integrated Repositoryの最新版では分析機能が強化され、設計情報間の依存関係の可視化機能が搭載されています。これにより、画面遷移図の自動作成や特定の設計要素の変更に対する影響範囲の特定が容易となります。また、UML図の描画機能を提供することで仕様の動的な振る舞いも表現可能となります。GTAMでは、ブラウザを起動せずにテストを自動実行するヘッドレスモードを提供し、さらなるテストの自動化を推進していきます。

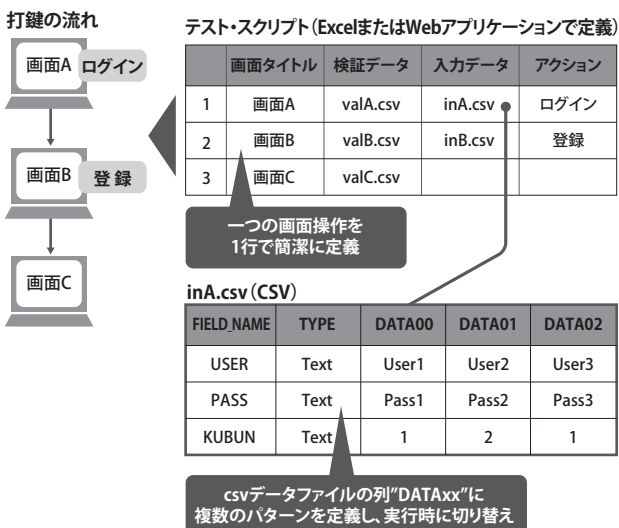


図6. GTAMでのテスト・スクリプトとデータ定義

### [参考文献]

- [1] SyncDev: MVP (MINIMUM VIABLE PRODUCT) ,  
<http://www.syncdev.com/minimum-viable-product/>
- [2] MARTINFOWLER.COM: Microservices - a definition of this new architectural term,  
<https://martinfowler.com/articles/microservices.html>



日本アイ・ピー・エム株式会社  
グローバル・ビジネス・サービス事業  
アドバンスド・ソフトウェアアセット  
Web技術 担当  
アドバイザー・アーキテクト

**北田 純弥**  
Junya Kitada

2000年日本IBM入社。金融や流通のお客様を中心にWebアプリケーション開発案件のシステム構築に従事。社内外で開発プロセスの展開などの業務を経て、現在は次世代のWeb技術の推進と展開活動を担当。



日本アイ・ピー・エム株式会社  
グローバル・ビジネス・サービス事業  
アドバンスド・ソフトウェアアセット  
クラウド・アプリケーション技術 担当  
シニア・アーキテクト

**松原 武司**  
Takeshi Matsubara

1999年日本IBM入社。複数の業種・業界のお客様に対するアプリケーション開発場面における各種標準化の策定や技術支援に従事。以降、アプリケーション・フレームワークやツールの設計と開発、アセット・ビジネスの展開と推進を実施。



日本アイ・ピー・エム株式会社  
グローバル・ビジネス・サービス事業  
アドバンスド・ソフトウェアアセット  
アドバイザー・ITスペシャリスト

**小山 悟**  
Satoru Koyama

1999年日本IBM入社。ワークフロー製品の開発と複数の業種・業界のお客様へのワークフロー・アプリケーションの展開や技術支援に従事。現在は画面テストの自動化ツールの設計、開発、展開活動を担当。

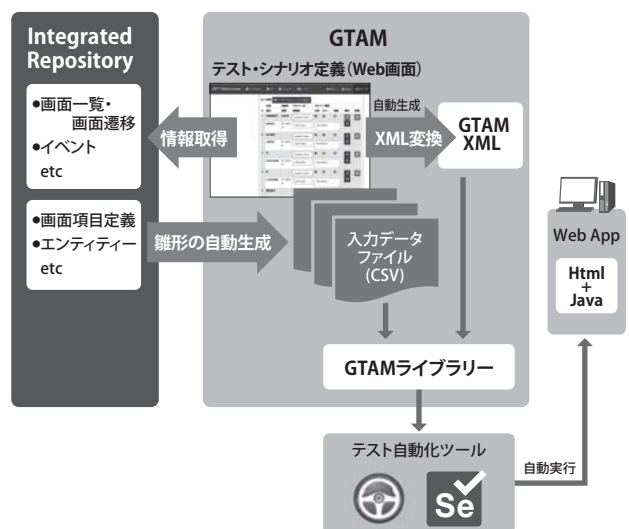


図7. GTAMとIntegrated Repositoryとの連携