

エンタープライズ SBM での 設計変更管理

目次

ページ

概要	1
適応と統合を目的に設計されたSBM	1
SBMプロセスアプリの開発	2
SBMを活用するうえでの責任と人材配置	3
長期にわたり反復する開発作業の調整	6
まとめ	10

SBMは、ビジネスの支援、追跡、モニターを可能にするプロセスベースの作業管理製品です。

概要

この文書では、OpenText™ Solutions Business Manager (SBM)の保守および開発プロセスを体系化して、ビジネスニーズの変化に応じたプロセスアプリの長期的な保守を容易にする方法について説明します。

適応と統合を目的に設計された SBM

SBMは、ビジネスの支援、追跡、モニターを可能にするプロセスベースの作業管理製品です。この製品は部門単位で使用することもできますが、組織全体に適用してさまざまなビジネスプロセスの構造化と可視化を図る、というのが最も効果的な利用方法です。たとえば、IT組織ではヘルプデスクの管理にSBMを利用することがよくあります。社内のユーザーがOpenText™ Request Centerを使用してチケットを送信すると、SBMプロセスによって適切な部門のフルフィルメントプロセスにチケットが転送されます。新しい従業員のオンボーディング、福利厚生、休暇申請は人事部門に送られ、IT部門に電子メールが届き、インフラストラクチャの問題と経費報告書が財務部門に送られます。SBMに組み込まれたレポート機能と監査機能により、プロセス管理者は傾向を大局的に理解できるだけでなく、あらゆるレベルにドリルダウンしてトランザクションやチケットの詳細を把握できます。

変化の大きい組織のニーズを満たすために、SBMは多くの基本的な前提条件に留意して設計されています。

変化は基本

変化はビジネスの基本です。市場の状況、ビジネス目標、ビジネス要件は絶えず変化しており、成功している企業はこれらの変化に常に適応しています。ビジネスプロセスの自動化は、変化を絶えず把握し、最適化の方法を見つけて継続的に改善することができなければなりません。

SBMは、新たに生じるユーザー要件や変化するユーザー要件に迅速に適応できるように設計されています。プロセスの変更は、管理対象のビジネスプロセスに精通している、開発者以外のユーザーでも簡単にアクセスできるように設計されたドラッグアンドドロップインターフェイスを使用して行います。

SBMの目標は、新しい要件が発生した場合や、既存の要件を満たす際の非効率性が見つかった場合に、管理対象の日常的プロセスに携わる人たちがプロセスを調整し適応できるようにすることです。SBMはプロセスの変更を簡素化します。柔軟性に欠けるプロセスにビジネスを適応させるのではなく、ビジネス要件の変更や新たな要件にプロセスが適応できるようにします。

システムもプロセスに参加する

IT環境には、さまざまなツールやシステムが数多く存在します。プロセス自動化ソフトウェアは、現在のシステムと連携するだけでなく、並行して機能する必要があります。人々がプロセスに参加するのと同様、他の記録システムもプロセスに参加しています。プロセス設計者は、システムどうしの接点を特定して最上位のプロセス設計に組み入れます。SBMは、そのような統合ができるよう構築されています。たとえば、SBMのREST Web サービスサポート、SOAP Web サービスサポート、オーケストレーションワークフロー、AppScript、あるいはC++ APIも使用して、SBMと、IT環境を構成する他のシステムとのギャップを解消できます。

SBM プロセスアプリの開発

プロセスアプリの開発は、自動化されるプロセスについて理解しているビジネス所有者の密接な協力のもと、反復して行うのが最適です。ビジネス所有者は、プロセスに参加する人々、それぞれが果たす役割、収集すべきデータ、プロセスを管理する人たちにとって必要なレポートを把握しています。また、ビジネス所有者は作業目標を理解し、そして生産性、収益、品質、リスク軽減、可視化の各目標が確実に達成されるように開発を導きます。

OpenText™ の SBM をベースにしたソリューションを購入する主な目的が、現在のプロセスアプリをビジネス環境特有の状況に適応させることという場合があります。その場合、プロセスの状態やフローの違いを反映するために単にデータモデルを変更したり（プロセス中に収集された情報の変更）、ワークフローマップを変更したりするだけで済むことも少なくありません。

新しいプロセスアプリの開発

要件をあらかじめ詳細に定義しなければならないカスタム実装とは異なり、SBM では参加者の作業を支援するためのシンプルなプロセスから始めて、その後徐々に作業を反復的に改善していくことができます。

開発中、プロセス設計者は以下を視覚的に定義します。

1. プロセスで収集、表示、操作する必要のあるデータ
2. 組織内の作業の動きを反映するプロセスワークフロー
3. プロセス参加者の役割と責任
4. プロセス参加者に表示されるユーザーインターフェイス
5. プロセスの健全性をモニタリングする担当者に可視化されたデータを提供するレポート

プロセスアプリが十分な成熟度に達して、ビジネス所有者による検証が完了すれば、エンドユーザーにロールアウトできます。

ロールアウト後に行う既存のプロセスアプリの改良

SBM の場合、最初のロールアウトは終了地点ではなく、継続的な改善への出発点です。まずはプロセス所有者を特定することをお勧めします。プロセス所有者はプロセス参加者からフィードバックを収集し、それに対して SBM プロセスを進化させます。

また、プロセス参加者の代表者と定期的に会合するフォーラムを利用することで、フィードバックを確実に収集し、対応策を実行することもできます。このフェーズではフィードバックを使用して次のようなことが可能になります。カスタムフォームの改善、フォームアクションを使用した UI の振る舞いの自動化、REST グリッドを使用した外部データへのアクセス、および関連データにアクセスするための組み込み型レポートの提供です。追加のレポートのニーズが発生した場合は、データ使用者に充実した内容のレポートを提供します。この種の変更は、技術やプログラミングに関して高度なスキルを持たないユーザーでも実行できます。唯一の前提条件は、SBM Composer に精通し、モデル化されているプロセスについての知識があることです。

たとえば、ユーザーからのフィードバックによって、ユーザーが必要なフィールドにアクセスするためには遷移フォームの一番下までスクロールしなければならないことがわかります。SBM を実装することで、プロセス管理の機能要件を達成できるだけでなく、ただちに対応できる、あるいは対応すべき非効率性も特定できます。OpenText™ SBM Composer を使用してフォームの上部にフィールドをドラッグすれば、このような問題は数分で修正できます。

OpenText の SBM をベースにしたソリューションを購入する主な目的が、既存のプロセスアプリをビジネス環境特有の状況に適応させることという場合があります。

まずはプロセス所有者を特定することをお勧めします。プロセス所有者はプロセス参加者からフィードバックを収集し、それに対して SBM プロセスを進化させます。

SBM の価値を最大限引き出すためには、インストールの目的が適切であり、かつそれぞれの目的に合った適切な人材を確保する必要があります。

SBM を活用するうえでの責任と人材配置

SBM の価値を最大限引き出すためには、インストールの目的が適切であり、かつそれぞれの目的に合った適切な人材を確保する必要があります。反復プロセスを迅速に改善するために最も重要な点は、自動化の対象となるビジネスプロセスに精通していることです。変更サイクルが長くなると、ビジネスの変化に合わせてプロセスを変更することが難しくなります。その一方で、SBM システムの統合や作成・保守の技術的側面については、開発を開始する前に慎重に要件を定義する必要があります。それにより、ビジネスと距離を置いて、より長い開発サイクルの中で実装できます。このセクションでは、SBM システムの実装と管理に関わるさまざまな責任について説明します。*

インストールとシステム設定

システム管理者は、通常、SBM に関連するインストール、アップグレード、システム設定のタスクを実行します。このような責任は一般に、社内の物理的なインフラストラクチャやソフトウェアインフラストラクチャの保守を担当する IT スタッフに割り当てられます。

SBM 管理

システム管理者とは対照的に、SBM 管理者は SBM プロセスの日々の運用を管理します。SBM 管理者は、SBM Application Administrator を使用して、SBM ユーザー / グループ、権限、通知、プロジェクト、管理ワークフロー上書きを管理します。ユーザーが必要に応じてレポートを定義することも可能ですが、通常は管理者がプロセス参加者とオブザーバーのニーズを満たすさまざまなレポートを定義します。管理者は、SBM で実装されているビジネスプロセスと SBM ユーザーベースのニーズに精通している必要があります。

SBM プロセス設計の保守

SBM 管理と密接な関連があるのが、SBM Composer を使用してプロセスの設計をインクリメンタルに変更する責任です。SBM の管理者や設計者は、ソリューションフォーラムに参加することで、ユーザーやプロセス所有者と協力して、変化する要件を把握し、プロセス参加者に最大の価値をもたらす変更を迅速に実装できます。プロセスに近い人間が短期間で反復的に設計変更を行うことで、SBM の「変化に対応できる」機能を活用できます。

以下のセクションでは、SBM Composer で行われる設計作業のタイプと、それぞれに必要なスキルレベルについて説明します。

ワークフロー設計

ワークフローは、プロセスに参加している項目が組織内でどのような処理経路をたどるかを定義するものです。ワークフローの最上位レベルは、一連の休止ポイント (状態) と状態間で項目が移動する遷移で構成されます。アクティブな状態には、その項目に対して現時点で責任を持つ所有者がいます。ユーザーは状態間で項目を移行させて責任を別のユーザーに委ねることができます。ビジネスルールを使用して、項目のルーティング方法を収集された情報に基づいて決める決定ノードを定義できます。ワークフローのさまざまなプロパティによって、データの遷移と表示へのアクセスが、項目を表示するユーザーの役割に応じて管理されます。プロセス設計者は、状態や遷移の追加または削除、意思決定の振る舞いを管理するビジネスルールの変更、またはワークフロー、状態、遷移レベルでのデータ表示の変更を行うことで、ワークフローを修正できます。この種の変更はプロセスの進化に対応するために一般的に行われるものであり、ビジネスプロセスや SBM Composer を基本的に理解していれば実行することができます。

*SBM システムでは、その規模や複雑さに応じて、1つの役割に複数の人を割り当てることも、1人の人が複数の役割を果たしたり、SBM の役割と SBM 以外の責任を兼任したりすることもできます。

データ設計

SBM の各アプリケーションには、プロセスのライフサイクル全般にわたって項目内で収集された情報を定義する**プライマリテーブル**があります。新しい要件が発生すると、プロセス設計者は通常、追加の情報を収集するためにプライマリテーブルにフィールドを追加することでデータモデルを変更します。たとえば、データモデルに**【緊急度】**フィールドの追加が必要な場合、プロセス設計者はテーブルパレットの**【選択】**(Selection)フィールドをプライマリテーブルにドラッグし、「緊急度」という名前を付け、フィールドのプロパティに「低」「中」「高」「重大」という選択値を追加します。新しいフィールドは、フィールドの**【フォーム】**(Forms)タブの**【フォームに追加】**(Add to Forms)機能を使用して、関連するフィールドと相対位置を指定することで、カスタムフォームに簡単に追加できます。ワークフローの変更と同様に、データ設計の変更は比較的簡単で、技術やプログラミングに関する特別なスキルがなくても行うことができます。

ビジュアル設計

SBM では、カスタムフォームを定義しない限り、状態と遷移のための「クイックフォーム」が自動的に生成されます。クイックフォームを使用している場合、フィールドレイアウトの制御は**【フォームのすべての行に広げる】**(Span entire row on forms)などのプロパティやフィールドの**【権限】(Privilege)セクション**に限定されます。フィールドの表示順は、状態の遷移プロパティを変更してワークフロー、状態、または遷移プロパティのフィールド権限の順序および遷移ボタンの順序を変更することで制御できます。

ただし、システムのプロトタイピングを行う場合や状態または遷移の使用頻度が低い場合にはクイックフォームが便利ですが、カスタムフォームにはデータ表示をより柔軟に設定できる機能や、フォーム上でアクティブな振る舞いを定義できる機能があり、また組み込み型レポート、HTML フレーム、および REST Web サービスを使用したシンプルな統合が可能になります。カスタムフォームを使用すると、ビューとデータ入力フォームの両方を最適化して、エンドユーザーの利便性を向上できます。

カスタムフォームの設計は簡単にできます。OpenText™ SBM Composer のビジュアルフォームエディターでは、使い慣れたドラッグアンドドロップ操作でフィールドの追加や移動ができます。クイックフォームをベースにしてカスタムフォームを作成することも、フォームをまったく新しく設計して、フィールド、コンテナ、アクションボタン、画像、リンク、レポート、REST グリッドなどを追加することもできます。カスタムフォームでは、フォーム上の遷移ボタンの配置や、ツールバーとヘッダーのアクションおよび表示方法をすべて詳細に制御できます。

ビジュアル設計の次の段階では、フォームに振る舞いを追加します。たとえば、重大度に応じて詳細タブを自動的に選択するような場合は、次のようなフォームアクションを追加します。

When

このフォームがロードされたとき

Or **【重大度】**(Severity)フィールドの値が変わったとき

If

【重大度】(Severity)フィールドの値が**【緊急】**の場合

Then

【詳細】(Details)タブを有効にする

Else

【標準】(Standard)タブを有効にする

動的フォームアクションを作成するためのフォームアクションは多数あり、フォーム上で実行したいと考えるイベント、条件、アクションのほとんどに対応できます。該当する条件がない場合は、ビジネスルールを参照します。またフォールバックとして、JavaScript スニペットを記述してより複雑な条件を実装することもできます。同様に、該当するアクションがない場合は、イベントの発生時に条件を満たすとアクションを実行するような小規模な JavaScript スニペットを記述します。

SBM の各アプリケーションには、プロセスのライフサイクル全般にわたって項目内で収集された情報を定義する**プライマリテーブル**があります。新しい要件が発生すると、プロセス設計者は通常、追加の情報を収集するためにプライマリテーブルにフィールドを追加することでデータモデルを変更します。

カスタムフォームの設計は簡単にできます。SBM Composer のビジュアルフォームエディターでは、使い慣れたドラッグアンドドロップ操作でフィールドの追加や移動ができます。

ある組織では、1つの部門システムでSBMを使用した後、組織内の広範なプロセス向けに積極的に新しいプロセスアプリを作成し、大きな成功を収めました。

カスタムフォーム作業のほとんどは、特別なスキルやプログラミング知識のないプロセス設計者でも行うことができます。ユーザーの視覚体験を向上させるためにビジュアル設計者を採用することもできますが、その必要が生じることはほとんどありません。

SBM プロセス設計の作成

プロセスアプリの作成は、前述の保守作業と似ています。異なる点は、既存のプロセスのインクリメンタルな変更ではなく、プロセス設計者がプロセス所有者と協力してプロセスアプリ全体の構想や設計を進めるという点です。ある組織では、1つの部門システムでSBMを使用した後、組織内の広範なプロセス向けに積極的に新しいプロセスアプリを作成し、大きな成功を収めました。まず、自動化、制御、監査、レポート機能のメリットを得られる手動プロセスを特定します。そして、プロセス所有者と協力して、プロセスのステップと遷移、収集するデータ、参加者の役割を定義します。次に、SBMを使用して、ビジネス上の問題を解決するプロセスアプリを迅速に作成します。ほとんどの作業は、よく知られたメタファーや使いやすいインターフェイスを使用してプロセスをレイアウトするSBM Composerのアプリケーションエディターで行います。プロセス設計者は、テーブルエディターでデータモデルを設計し、ワークフローエディターでワークフローをレイアウトし、役割を割り当ててプロセスを自動化し、そして開発環境でプロセスを短期間で稼働させることができます。その後、エンドユーザーへのロールアウトに適した最低限実用的な製品になるまで、繰り返し設計を改善します。

プロセス設計の作成と保守を担当する人がビジネスアナリストの肩書きを持っている人の場合もありますが、SBM Composerに精通している必要があります。プロセス設計者には、特別なソフトウェアエンジニアリングや開発者向けのトレーニングは必要ありませんが、Microsoft ExcelやPowerPointなどのビジネス生産性向上ツールを使用するためのスキルを備えている必要があります。

統合の設計と実装

プロセスの作成や改良とは対照的に、外部システムとの統合は、詳細な計画、複雑な実装、徹底したテストを必要とする時間のかかる作業になることがよくあります。統合の複雑さは、統合のタイプや関連するサードパーティ製品によって異なります。最も単純なケースは、カスタムフォームの比較的単純なREST Gridを使用した統合です。必要になるのは、サードパーティのREST Web サービスインターフェイスに関する知識があることと、SBM Composerでカスタムフォームを作成する方法を理解していることだけです。外部システムがビジネスロジックの粒度でSOAP Web サービスをサポートしている場合、統合を実装するには、SBM オーケストレーションを開発するより複雑な作業が必要になる場合があります。最も複雑な状況では、外部システムからデータにアクセスするためのプログラミングが必要になることがあります。

外部システムとの統合は多くのSBMユーザーにとって最も複雑な作業となることが多く、OpenTextではできるだけシンプルなものにしようと努めています。ビジネスレベルのSOAPおよびREST Web サービスインターフェイスを豊富に備えたシステムを統合する場合は、フォーム上でSBM オーケストレーションとREST統合を使用すれば、プログラミングしなくても統合できます。ただそれでも、作業の複雑さはビジネスアナリストが楽に行えるレベルより高くなる場合があります。統合開発者に必要なスキルセットは、実行する作業によって異なりますが、ソフトウェアエンジニアのスキルセットに近い傾向があります。

統合は長期的で、より詳細な計画を立案するなどの大きな手間がかかるため、上記のインクリメンタルなプロセス変更とは異なり、その開発は継続的なビジネスと密接に結びついている必要はありません。アプリケーションに依存しないオーケストレーションなどのSBM機能を活用して、手間のかかる開発作業を隔離し、完了した時点でプロセス開発者が使用可能な成果物として提供できます。

長期にわたり反復する開発作業の調整

前述のように、SBM での開発には極めてタイプの異なる 2 つの作業が必要になります。作業のほとんどは管理とビジュアル設計に関係するもので、技術やプログラミングに関する深い知識がなくても遂行できます。必要なのは、ビジネスプロセスに近くかつ理解していて、そして SBM の Composer と Application Administrator について熟知していることがすべてになります。一方、オーケストレーションや統合では通常、異なるレベルの技術スキルを適用する必要があります。そうした複雑な作業は、反復的な保守作業より長期にわたり、かつ慎重に進めなければならない傾向があります。それでは、この動きが早く、反復的で、適応的な変化を、長期的な作業を伴う SBM プロセスに合わせて調整するにはどうすればよいでしょうか。このセクションでは、この問題を解決するために設計された SBM のさまざまなアプローチと機能について説明します。

反復的な変更

一般に、反復的な変更は、その作業が明確に定義され、以下のように適切にカプセル化されている限り、長期的な作業との詳細な調整なしに継続することができます。たとえば、開発中のオーケストレーションが現在のアプリケーションの要素(特定のフィールド、状態、遷移など)に依存している場合、反復的な変更が行われてもそれらは変更しません。長期的な作業で使用するインターフェイスの安定性を維持することは、作業が完了したときに作業の統合が簡素化される場合があります。一方で SBM プロセスの流動性を維持して、ユーザーやビジネスのニーズに合わせて動的に進化させることもきわめて重要です。長期的な作業が適切にカプセル化されていれば、迅速かつ容易にプロセスに組み入れることができます。

反復的な変更の典型的なプロセスについては、政策方針書『SBM Path to Production for Enterprises(企業の本番環境への SBM のパス)』で詳しく説明しています。

長期的な開発作業

SBM で長期的な開発作業を行う場合に中心となる目標は、作業を構造化することで、シンプルかつ明確に定義されたインターフェイスを使用して関連するプロセスアプリに結びつけることです。重要なのは、複雑さを露呈させることなく、カプセル化することです。開発作業は SBM の内外で発生する可能性があり、通常は一般的なプロセス設計者のスキルや楽に作業ができるレベル以上が求められます。このような作業に関わる開発者には、反復的な開発、ステージング、本番環境に使用されるプライマリ SBM システムとは独立した SBM システムの開発インスタンスが必要となります。

SBM 外での開発

このタイプの開発の例としては、サードパーティの基幹業務ソフトウェアのデータを SBM カスタムフォームの REST Grid に表示する REST サービスの作成があげられます。場合によっては、これに外部システム上の API とのインターフェイスとなるコードモジュールの作成や、サービスのホスティングが伴うこともあります。同様に、SBM オーケストレーションでの使用に適したビジネス単位で、または移行時の Web サービスアクションとして、外部システムの機能を公開する SOAP Web サービスの作成とホスティングを伴う場合があります。開発中の SBM システムのインターフェイスが SBM 要件と互換性があることも保証されなければなりません。上の例で言えば、SBM を使用して、開発中の REST および SOAP Web サービスをテストすることになります。その性質上、このタイプの開発作業は、明確に定義された高レベルの SBM インターフェイスを提供するため、カプセル化されています。

SBM 内での開発

このカテゴリの開発では、SBM が提供する、より技術的なさまざまな統合テクノロジーを使用します。例としては、適切な障害処理と補償を伴うオーケストレーションの開発があげられます。SBM C API を呼び出すための複雑なスクリプトやソフトウェアの開発もこのカテゴリに該当します。このような作業は、SBM 外で行われる開発と連動していることがよくあります。たとえば、外部システムでは低レベルの API しか公開されていない場合があります。そのような場合、API をビジネスレベルの Web サービスに組み込むことで、SBM オーケストレーションで使用可能になります。

必要なのは、ビジネスプロセスに近くかつ理解していて、そして SBM の Composer と Application Administrator について熟知していることがすべてになります。

長期的な作業で使用するインターフェイスの安定性を維持することは、作業が完了したときに作業の統合が簡素化される場合があります。その一方で、SBM プロセスの流動性を維持し、ユーザーやビジネスのニーズに合わせて動的に進化させることも極めて重要です。

SBM のオーケストレーションの 4 つのタイプ:

- 同期オーケストレーションワークフロー
- アプリケーションに依存する非同期オーケストレーションワークフロー
- アプリケーションに依存しない非同期オーケストレーションワークフロー
- 外部から開始される非同期オーケストレーションワークフロー

アプリケーションに依存する非同期オーケストレーションワークフローは、SBM アプリケーションから開始される最も一般的なオーケストレーションです。

SBM 開発のカプセル化の手法

アプリケーションワークフローでの遷移に関するアクションは、カプセル化された開発作業とのやり取りのためのインターフェイスポイントのほとんどとして機能します。たとえば、同期ワークフロー、非同期ワークフロー、AppScript、Web サービスなどはすべて、遷移に関するアクションから開始されます。もう 1 つの主要な統合ポイントは、エンドユーザーに表示されるフォーム上で行われる「単一画面上」でのジャストインタイム統合です。以下のセクションでは、さまざまなタイプの統合と、適切な分離とカプセル化によってその後のプライマリ SBM システムとの統合を容易にする開発手法を説明します。

一般的な考慮事項 - 内部名: 以下に説明する統合手法の多くは、SBM Web サービス、AppScript、C API などを使用して SBM アプリケーションのワークフローとの相互作用を図るものです。フィールド名は変更される場合もあるため、フィールドと相互作用するすべての Web サービスでは、フィールドデータベース名を使用する必要があります。フィールドデータベース名は初期導入後に変更することはできません。ワークフロー、状態、遷移への参照は、完全修飾内部名を使用する必要があります。これらの項目の名前プロパティはあいまいであったり、ニーズに応じてアプリケーションを修正すると名前が変更されたりするためです。

オーケストレーション: このセクションでは、オーケストレーションの 4 つのタイプについて説明します。長期的な開発作業でオーケストレーションを開発している場合は、可能であれば、後の統合を容易にするために別のプロセスアプリでオーケストレーションを開発します。各タイプのオーケストレーションは以下のとおりです。

- **同期オーケストレーションワークフロー:** このオーケストレーションワークフローは、遷移アクションによって SBM アプリケーションから呼び出されます。項目データは、オーケストレーションワークフロー入力にマッピングされ、オーケストレーションワークフロー出力によって更新されます。同期オーケストレーションは、それを呼び出すアプリケーションと同じプロセスアプリに存在している必要があります。そのため、オーケストレーションの開発後に、SBM Composer の比較/マージ機能を使用して、プライマリ SBM システムのプロセスアプリにマージする必要があります。同期オーケストレーションワークフローの仕様では、同期オーケストレーションの入力と出力を定義する必要があり、これらのフィールドが開発作業中に変更されることはありません。
- **アプリケーションに依存する非同期オーケストレーションワークフロー:** SBM アプリケーションから開始される最も一般的なオーケストレーションです。このオーケストレーションは、遷移アクションを使用してイベントを発生させることで実行されます。イベントデータは、そのアプリケーションのプライマリテーブルに存在するデータに対応しています。このようなオーケストレーションは、外部システムと SBM アプリケーションの両方で Web サービスを呼び出すことがあります。このオーケストレーションの場合も、開発中はアプリケーションとオーケストレーションとのインターフェイスを変更せずに保持する必要があります。プライマリテーブルのフィールド名を変更するときに、オーケストレーションとのインターフェイスが変更されないようにするには、アプリケーションの **[応答なしのイベント]**(Event without Reply) オーケストレーションリンクの **[定義をロックする]**(Lock definition) チェックボックスがオンになっていることを確認する必要があります。この方法は、SBM 2009 R4.02 以降で利用できます。

このようなオーケストレーションは、呼び出し元のアプリケーションと同じプロセスアプリの一部である必要はありません。別のプロセスアプリで互換性のあるオーケストレーションを作成するには、開始アプリケーションから「応答なしのイベント」(Event without Reply) というオーケストレーションリンクをエクスポートし、別の空のプロセスアプリでオーケストレーションを追加します。Application Explorer の **[アプリケーションリンク]** (Application Links) 項目を右クリックして **[新しいイベント定義の追加 ...]**(Add New Event Definition) メニュー項目を選択します。ダイアログボックスが表示されたら、**[イベント定義ファイルから作成]** (Create from event definition) オプションを選択し、エクスポートしたイベント定義に移動して、開発作業を進めます。また、テスト対象のイベントを発生させる SBM システムが必要になります。

- **アプリケーションに依存しない非同期オーケストレーションワークフロー**：外部システムを含む、複数の SBM アプリケーション間で再利用できる非同期オーケストレーションワークフローです。このようなワークフローのイベントインターフェイスは、アプリケーションに依存せず、外部システムに対して十分な指示を提供するためにオーケストレーション開発者が定義します。つまり、これらのワークフローは、外部システムに対する再利用可能な共通のインターフェイスを定義するためのコマンドであると考えられます。ワークフローは、遷移に関するアクションを使用して SBM アプリケーションから呼び出されます。アプリケーションに依存する非同期オーケストレーションワークフローと異なる点は、アプリケーションの項目データをイベントインターフェイスにマッピングできることです。アプリケーションに依存するイベントインターフェイスは、アプリケーションの項目データによって自動的に定義されます。

アプリケーションに依存しない非同期オーケストレーションワークフローを作成するには、新しいプロセスアプリを作成し、Application Explorer でプロセスアプリを右クリックし、[新規作成] (Add new)、[オーケストレーション] (Orchestration) の順に選択してオーケストレーションを追加します。Application Explorer の [アプリケーションリンク] (Application Links) 項目を右クリックして [新しいイベント定義の追加 ...] (Add New Event Definition) メニュー項目を選択します。ダイアログボックスが表示されたら、[新しいカスタムイベント定義の作成] (Create new custom event definition) オプションを選択し、[オブジェクトタイプ] (Object types) と [イベントタイプ] (Event types) を定義します。これらの値は、イベントマネージャーがイベントの発生時に、実行するオーケストレーションを決定するのに使用されます。次に、イベント内で引き渡されるカスタムデータを定義します。これらの値は、イベントを使用して実行されるオーケストレーションワークフローへの入力値になります。

- このアプリケーションリンクで定義されたイベントを発生させるには、最初に **[イベント定義のエクスポート ...]** (Export event definition) ボタンをクリックしてエクスポートし、それを .mtd ファイルに保存します。次に、開始アプリケーションで、Application Explorer の **[オーケストレーションリンク]** (Orchestration Links) の **[外部イベント]** (External Events) 項目を右クリックして、**[新しい外部イベントの追加 ...]** (Add New External Event) メニュー項目を選択し、イベントファイルに移動してアプリケーションにインポートします。最後に、このイベントを使用するには、遷移のアクションを作成し、オーケストレーションワークフローを選択して最初のウィザードページのステップ 2 で、**[ローカルイベント]** (the local event) をクリックして **[外部イベント]** (an external event) に変更します。ウィザードを終了して、インポートした外部イベントを選択し、データマッピングをクリックして、アプリケーション内の項目データをイベントの入力にマッピングします。この方法は、SBM 10.1 以降で利用できます。

- **外部から開始される非同期オーケストレーションワークフロー**：外部システムから発生したイベントによって開始される非同期オーケストレーションワークフローです。このようなオーケストレーションワークフローのイベントインターフェイスは、外部システムの要件に従って定義されます。通常、特定のアプリケーションの項目を操作するための SBM Web サービスが呼び出されます。

外部システムから開始できるオーケストレーションを作成するには、上記と同じ手順を実行して、オーケストレーションを作成し、新しいカスタムイベント定義を定義します。そして、イベント定義をエクスポートする代わりに、**[外部イベント WSDL のエクスポート]** (Export external event WSDL) ボタンをクリックして、イベントの発生に適した Web サービスを指定する WSDL ファイルを作成します。詳細はドキュメントでご確認ください。

その他の高度な開発手法：その他のタイプの高度な開発には、AppScript 開発や C API 開発などがあります。多くの SBM ではこのようなインターフェイスを活用していますが、フォームアクション、Web サービス、オーケストレーションを使用して必要な作業を完了できれば、より保守がしやすいシステムになります。

長期的なプロジェクトの開発が完了したら、その作業をプライマリ SBM システムに戻す必要があります。

この種の開発をプライマリ SBM システムとは別の開発作業として行う場合に重要なのは、インターフェイスポイントがよく理解され、安定していることです。SBM 10.1 で導入された内部名機能により、以前よりも開発作業が大幅に簡単になりました。以前は、ワークフロー、状態、遷移などの SBM ワークフローモデルへの参照はすべて名前で行われていました。フォーム上には状態名と遷移名が表示されるため、ワークフローの通常の保守や進化によって名前が変更され、統合が中断される可能性があります。現在は、AppScript、Web サービス API、または CAPI を使用したこれらの項目への参照には内部名を使用します。内部名はプロセスアプリの最初の発行後も変更されることはありません。

開発、ステージング、本番システムへの長期的な変更の統合

長期的なプロジェクトの開発が完了したら、その作業をプライマリ SBM システムに戻す必要があります。

オーケストレーション

別のプロセスアプリで開発されたオーケストレーションの場合、このプロセスは極めてシンプルです。外部システムによって開始されるオーケストレーションは、オーケストレーションを含むプロセスアプリを導入すれば、外部システムでイベントが起動すると実行されます。その他の非同期オーケストレーションタイプでは、呼び出し元のアプリケーションを変更して、オーケストレーションを呼び出すイベントを発生させるだけです。オーケストレーションが外部イベントを使用して定義されている場合は、呼び出し元アプリケーションのデータをオーケストレーションの入力にマッピングする必要があります。アプリケーションデータモデル(アプリケーションに依存するオーケストレーション)に基づいて定義されている場合は、遷移のアクションでイベントを発生させるだけです。

同期オーケストレーションまたは関連アプリケーションと同じプロセスアプリに保持されているオーケストレーションの場合、Composer の比較およびマージツールを使用して、オーケストレーションとそのオーケストレーションリンクを開発プロセスアプリの詳細計画からプライマリ SBM システムのプロセスアプリにコピーし、作業を行う場所の遷移からオーケストレーションを呼び出す必要があります。

APPSCRIPT のマージ

AppScript は、プロセスアプリ間でファイルとしてコピーできます。アプリアプリスクリプトおよびアプリアスクリプトと相互作用しているアプリケーションとの間のインターフェイスが安定している限り(たとえば、内部名を使用しているなど)、プライマリシステムにコピーされた後も引き続き機能します。

その他の変更のマージ

最後に、別のシステムで行った開発は、Composer の比較およびマージ機能を使用して、プライマリ SBM システムのプロセスアプリにマージすることができます。この機能を使用すると、プロセスアプリの別バージョンで行った変更を、プライマリシステムのプロセスアプリにコピーできます。この機能は、オーケストレーション、フォーム、ワークフロー、AppScript、ルール、役割、フィールド、補助テーブルなど、ほぼすべてのタイプの設計要素に使用できます。設計要素内に変更がある場合、それを比較することで、差異の概要を示すレポートが表示され、使いやすい SBM Composer 環境でその違いが視覚的に強調表示されます。

この手法により、個別に行った開発作業をプライマリシステムに組み込むことができます。ただし、長期的なプロジェクトの場合は個別の設計要素にカプセル化する必要があります。これにより、既存オブジェクトの別バージョンのプロパティをソートするという大変な作業をすることもなく、プロジェクトを 1 つのユニットとしてプライマリシステムに移動できます。

別のシステムで行った開発は、Composer の比較およびマージ機能を使用して、プライマリ SBM システムのプロセスアプリにマージすることができます。

まとめ

柔軟性に欠けるプロセスにビジネスを適応させるのではなく、変化するビジネスにプロセスを適応させることを可能にすることで、SBM の価値を引き出すことができます。

時間の経過に伴うプロセスの適応と進化に必要なほとんどの変更は、SBM を使用することで簡単かつ直感的に行うことができます。SBM は、ビジネスに精通した非技術者が反復的なプロセス変更を行えるように設計されています。そのような人々とプロセスとの距離を近く保つことが大切です。

プロセス所有者は、プロセス参加者（申請者、実行担当者、プロセスオブザーバー、プロセス所有者）の代表者からなるフォーラムを定期的で開催する責任を負います。このフォーラムでは、ソリューションの使用方法について話し合い、改善できる点を学ぶことができます。チームが提供するフィードバックに基づいて、プロセスを繰り返し改善します。

統合などの一部の作業は、長期的なプロジェクトとして開発者が行うのが最適です。独立した開発を可能にし、プライマリ SBM システムへ簡単に組み込めるような方法で開発作業をカプセル化します。

すぐに利益をもたらす小規模なプロジェクトと大規模な開発作業を 1 つにまとめないでください。そうすると、「変化に対応できる」という SBM の価値提案を損なうことになります。

詳細はこちら：

www.opentext.com

お問い合わせ

