



要求されるスピードと俊敏性を提供：
イベント駆動型ビジネスを実現する
イベント・ストリーム・プロセッシング


sonic
SOFTWARE™

要求されるスピードと俊敏性を提供： イベント駆動型ビジネスを実現する イベント・ストリーム・プロセッシング

Stan Davis氏とChris Meyer氏は、著書「Blur」の中で、「ビジネス、そして電子的に接続されている組織のすべての側面はリアルタイムで変化する。そしてすべてのものが、商品、人、組織、国など、他のすべてのものと電子的に接続されつつある。」と述べています。

接続性が変化のスピードを加速し、俊敏性に対する新しい要求を生み出す世界では、この世界にふさわしい新しいタイプの情報アーキテクチャが求められています。そのアーキテクチャは、このような新しい形態の接続性を活用して、この接続性が実現するスピードから組織が真のビジネス・バリューを得られるようにサポートできなければなりません。

これを実現するアーキテクチャがイベント・ストリーム・プロセッシング (ESP) です。

監視、分析、そしてアクション

イベント・ストリーム・プロセッシングは、組織が絶えず変化するデータに迅速に反応できるようにする新しいコンピューティング・パラダイムです。イベント・ストリーム・プロセッシングにより、組織はイベントの(1)監視、(2)分析、(3)イベントに基づいたアクションの実行までをミリ秒単位で実施することが可能になります。この3つの要素、すなわち、イベントの監視、分析、そしてイベントに基づいてアクションを実行できる機能は、多くの組織のビジネスに非常に大きな効果をもたらすことができます。これらの組織は、新しい形態の接続性を今すぐ取り入れ、この接続性がもたらす即応性を活用しなければならぬのです。

- ⇒ 金融サービス業界において、イベント・ストリーム・プロセッシングは、世界で最先端のアルゴリズム・トレーディング、リスク管理、コンプライアンス・システムの基盤として機能できることを実証しています。金融サービス企業は、イベント・ストリーム・プロセッシングによってトレードの好機を捉え、大きな利益を上げています。
- ⇒ 通信業界では、ネットワークを送信されるデータの監視、分析、そしてこのデータに基づいてアクションを実行する機能を提供するインフラストラクチャを活用することにより、サービス・プロバイダは、2.5G、3Gワイヤレス、ブロードバンド、IP ネットワークでQoS (Quality of Service) の要件に対応することが可能になりました。
- ⇒ 公益事業では、電力会社が送電ネットワークからデータを収集して分析できる機能を活用し、現在および今後のエネルギーの需要、および需要を満たすために必要な送電ネットワークのキャパシティに関する有益な情報をリアルタイムで取得しています。
- ⇒ 石油およびガス業界では、パイプライン運用の効率化のために、SCADA (Supervisory Control And Data Acquisition) システムがイベント・ストリーム・プロセッシングのインテリジェンスの活用を開始しています。このインテリジェンスによって、生産/流通施設のリアルタイムの可視性を取り、漏洩やその他の警告を誘発する状況を把握しています。
- ⇒ 軍事偵察活動では、イベント・ストリーム・プロセッシング・ツールが有人偵察機および無人偵察機 (UAV) から衛星データを収集し、離れた地域の最新の状況を地上部隊に通知しています。

どのような状況でも、新しいタイプのデータ、つまりかつてより多様かつ遅延に敏感で詳細なデータを収集、分析し、データに基づいてアクションを実行できる機能があれば、組織は状況に即座に対応することが可能になります。あるケースでは、この機能のおかげで、状況の変化を最初に発見した組織だけが活用できる新しい機会を得ることができます。別のケースでは、脅威または有害な状況を検知してこれらの危険を避けるためのアクションを講じることができます。どのようなケースでも、組織は迅速かつプロアクティブに、そして時には先制的なアクションに実行することが可能になります。相互接続された世界のメリットは、数ミリ秒のスピードで迅速にアクションを講じることができる組織だけが享受できるのです。

新しいアーキテクチャ・モデル

システム同士が新しい接続形態で接続されるようになり、これらのシステムが外界からデータを収集できる範囲が拡張された今、組織は新しい種類の情報を活用できるようになりました。しかし、これらの(次々到達し、即座に変化する膨大で詳細な)情報の属性は、従来のビジネス・アプリケーションのデータの属性とは異なります。従来のアプリケーションは、構造化アプリケーション/モデルを中心に設計されており、アプリケーションが使用するすべてのデータのストレージ、編成、索引作成の管理はデータベースが行います。データベースがすべてのデータの利用を管理するため、データベースに書き込まれ、索引付けが行われるまで、アプリケーションはデータを利用できません。

「(1)データ入力、(2)次に索引作成、(3)次に問い合わせ」というプロセッシング手法は、リアルタイムのイベントベース・システムには適していません(図1参照)。短いバーストの間(通常は1秒のスパン)に数千のデータ・イベントを受信するケースでは、特に受信イベントの問い合わせに使用する新しい

索引を次々に作成しなければならない場合、従来のデータベース駆動型アーキテクチャは新しいデータの流れに追いつけません。また、従来のデータベースがデータにアクセスするには、問い合わせを実行しなければなりません。新たに受信したイベントが検出すべき属性に一致するかどうかを確認する唯一の手段が問い合わせであるため、新しいイベントが連続して流れてくれば、新しい問い合わせが連続して次々に発生します。このアーキテクチャは、元来非効率的で、大きなオーバーヘッドが発生するのです。

この非効率性は、インメモリ・データベースを使用しても解消しません。インメモリ・データベースは、ストレージ・デバイスへのイベント・データの書き込みタスクを排除できますが、多くのイベント・プロセッシング・アプリケーションに要求される応答性は提供できません。完全にメモリ内で実行されるとしても、

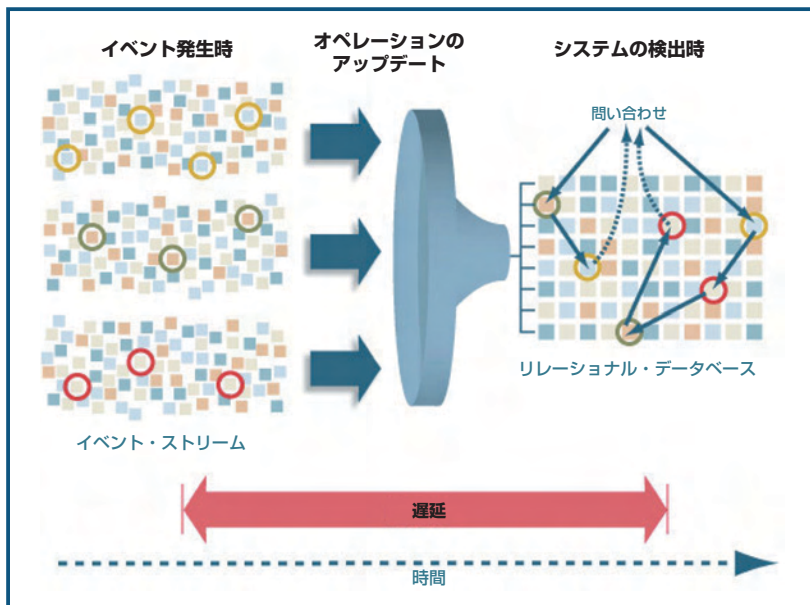


図1：従来のアーキテクチャと固有の遅延

インバウンド・イベントの索引を絶えず作成するタスク、そして同時に実行される問い合わせに時間がかかり過ぎるため、リアルタイムで応答することはできないのです。さらに、インメモリ・データベースかどうかに関わらず、すべてのデータベースは、トランザクションの完全性を保証するために最適化されており、トランザクションの原始性、一貫性、独立性、永続性を保証します。このようなトランザクション保証により、メモリ内で行われるかどうかを問わず、データ処理には固有の遅延が発生します。トランザクションの完了が、新しいイベントへの即座のアクセスよりも必ず優先されるためです。

Progress Apama

Progress® Apama®は、業界をリードするイベント・ストリーム・プロセッシング・プラットフォームです。Apamaプラットフォームは、高速に移動するイベント・データのストリームを監視し、イベント・パターンを検出、分析し、アクションを講じるまでの動作を数ミリ秒で実行しなければならないアプリケーションをサポートするために設計されています。金融サービス業界の最も要求の厳しいイベント駆動型アプリケーションでもその性能を実証したApamaのイベント・プロセッシング・アーキテクチャと、ビジネス・ユーザが利用可能なツール群は、リアルタイムでデータを監視、分析し、このデータに基づいてアクションを実施しなければならないあらゆるビジネスに、豊富なイベント・プロセッシング機能を提供します。

Apamaは、従来のコンピューティング・モデルの転換を通じて、最高のイベント・プロセッシング・パフォーマンスを実現しました。Apamaプラットフォームは、遅延を発生させることなく、インバウンド・イベント・データが到着し次第、データベースに書き込む前にデータ上でリアルタイム・イベント・シナリオを実行します。したがって、問い合わせの実行前にデータベースにイベントのスナップショットを保管し、索引を作成しなければならないシステムで発生する遅延は発生しません。Apamaは、特許取得済みのイベント監視モデルを通じて、ミリ秒単位という比類ない速度で実行できる時間ベースの問い合わせと属性ベース問い合わせをサポートします。

パフォーマンスを重視した設計 – Apama Event Managerのシナリオ

イベント駆動型アーキテクチャに対する要求が、コンピューティング・パラダイムに抜本的な変革をもたらしています。イベント駆動型アーキテクチャは、即座にインテリジェントな決定を下す必要があるため、遅延、同時性、スケーラビリティの面で、通常のアーキテクチャとは異なる要求を満たさなければなりません。イベント駆動型アプリケーションが成熟した今では、遅延の要件に対応しながら、1秒間に数万イベントの処理と、数千の異なるイベント・パターンの監視が同時に行われることも珍しくなくなりました。たとえば、金融市場イベントの発生、イベントの検出、応答（そのイベントに基づいた株式の売買）までの経過時間はわずか数ミリ秒です。それ以上の時間を要した場合は、すべての取引の機会を失うことになってしまいます。

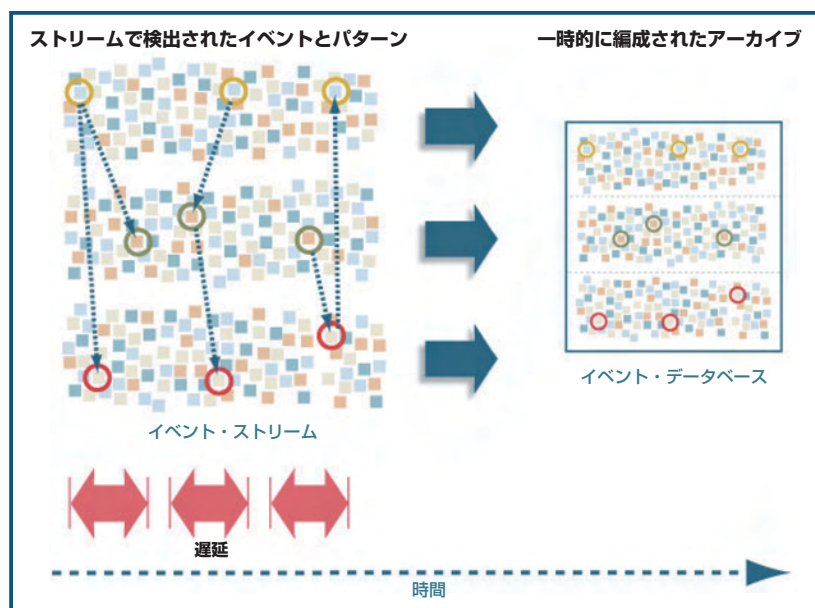


図2：遅延を低減/排除するイベント・プロセッシング・アーキテクチャ

遅延の排除は、イベント駆動アーキテクチャのパフォーマンスにとって重要な要素です。Apamaアーキテクチャは、(1秒間に数万のイベントが到着する)時間に厳格なインバウンド・イベント・データの使用前に、その編成と索引作成を必要とはせず、問い合わせの実行前に問い合わせの索引を作成します。「索引付けされた」問い合わせは、システムが受信するインバウンド・イベント・ストリームを監視します(図2参照)。そして問い合わせ内に事前に定義された分析条件に一致するイベントをストリーミング・データの中から即座に探し出します。Apamaは、このモデルの採用によって、データベースにおけるイベント・ストリーム情報の編成と索引作成のプロセスによって発生する遅延を排除した、高度なイベント・パターン・マッチングの実行を実現しました。

イベント駆動型システムのパフォーマンスにとってもう1つの重要な要素は、シナリオ分析の並行処理です。イベント・プロセッシングでは、膨大な数のイベントを監視し、条件を検出してアクションをトリガーする必要があります。しかし一般的にイベント駆動型アプリケーションは、単一のモノリシック・アプリケーションではなく、数千とは言わずとも、並行処理しなければならない数百の異なる個別の「シナリオ」で構成されています。

たとえば、ある金融トレーディング・デスクでは、様々な株式、アルゴリズム、期間などの多様な要素を反映した数百または数千にのぼる個別のトレーディング戦略が並行処理されています。また、ある産業プロセス制御アプリケーションは、大規模な工場や流通パイプライン全体に分散されているPLC(プログラマブル・ロジック・コントローラ)やRTU(リモート・ターミナル・ユニット)の数千のセンサーから発信される多様なアラートを反映した、膨大な数のイベント・シナリオを処理しています。

イベント駆動型システムのパフォーマンスに重要な最後の要素は、スケーラビリティです。Apamaは、インバウンド・イベントの高度なパターン・マッチングをサポートする独自の「HyperTree」構造を通じてスケーラビリティの問題を解決しています。HyperTreeは、完全なイベント・シナリオ全体をロードするのではなく、イベント・テンプレート内に属性が宣言的に定義されている多次元イベント・パターンの管理のみを行います。アクションをトリガーする条件に一致するイベントを検出するためには、数千もの(シナリオ自体の複雑さに応じては数百万にのぼる)イベントを監視しなければならないため、特定の1つのイベント条件が一致しても、実行すべきアクションは即座にはロードされません。HyperTreeアーキテクチャは、このタスクの実行のために特別に設計されており、ほとんどまたは全くパフォーマンスに影響を与えずに数万のイベント・テンプレートをサポートすることができます。

時間を捕捉するためのアーキテクチャ – Apama Temporal Sequencing

イベント・プロセッシング・アーキテクチャは、パフォーマンスの要件に対応するだけでなく、イベント・データの基本的な属性として「時間」を捕捉しなければなりません。多くのイベント駆動型アプリケーションでは、「何が」発生したのかと同様に(多くのシナリオでは「何が」発生したかよりも)何が「いつ」発生したかが重要です。したがって、イベント・プロセッシング・アーキテクチャにとっては、各イベントとイベント・パターンの基本的な属性とシーケンス(発生順序)を表す属性両方を完全に捕捉することが非常に重要です。多くのケースでは、これらのイベントのシーケンスはイベント自体と同様に重要な意味を持ちます。

時間的な属性を捕捉しなければならない重要な理由は、イベント・プロセッシング・アプリケーションは、類似するイベント・ストリーム活動がシステムに供給される際は必ず同じ結果となることを保証

するために、厳密に振る舞わなければならないことです。イベント・プロセッシング・アプリケーションが厳密に振る舞うことができなければ、開発者はテスト環境で得られた結果が実世界の配備環境でも再現されるという確信を持つことができなくなり、テストとシミュレーションの結果の重要性が低下してしまいます。要求される厳密な振る舞いを実現するために、Apamaは、時間的な条件を伴うパターンの検出(たとえば、イベントAの後にイベントBが発生し、次にY期間内にイベントC=Xとなるパターンの検出)を実行する「時間的シーケンサ」(図3を参照)を組み込んでいます。このシーケンサにより、時間を中心とする問い合わせを宣言的に指定することが可能になり、高度なイベント・パターンを定義する値ベースの属性(「何が」起こるか)と時間ベースの属性(「いつ」起こるか)両方を検出する能力が飛躍的に向上します。ここでは、時間的シーケンサが時間ベースの値を処理

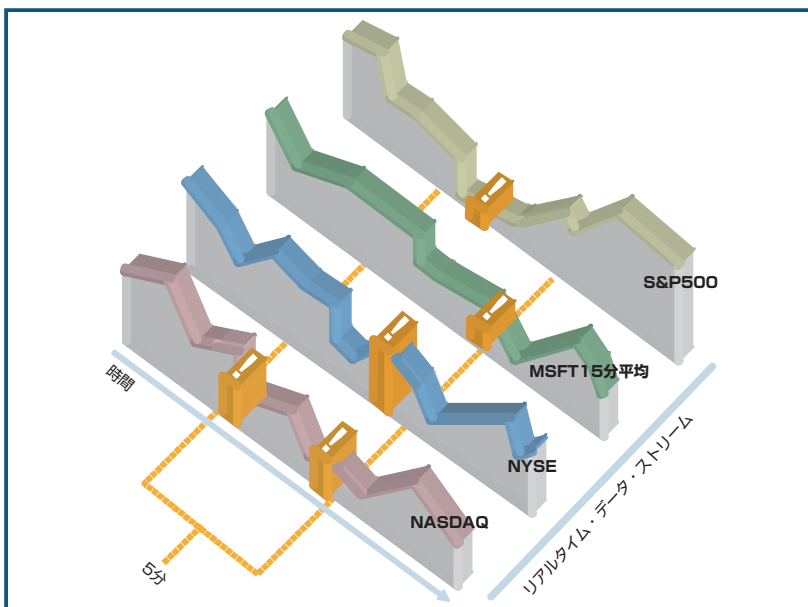


図3：属性と時間の関係の捕捉

し、すでに説明したHyperTreeがイベント・テンプレート内に指定された属性を監視します。

時間的シーケンシングは、従来のデータ・アーキテクチャには絶対に実現できないパフォーマンスをApamaアーキテクチャが提供する領域でもあります。従来のデータベースは、各イベントの時間的属性を取得し、次に問い合わせを使って目的のイベントからの経過時間を計算することでしか、時間的條件を捕捉できません。このようなモデルは、重要でない目的のためにシステムに膨大な要求を課してしまいます。対照的にApamaアーキテクチャは、データベースへの問い合わせを必要とせずに、発生する時間的條件をリアルタイムで検出することができます。

予測した結果を得るためのアーキテクチャ – Apama EventStore

Apamaアーキテクチャの基盤は、データベースの仲介や時間のかかる問い合わせに依存せずに、高速に到着するイベントを分析できる機能です。しかし、イベント駆動型アーキテクチャでも、イベント履歴のリポジトリとしてイベント・データベースが重要な役割を果たします。Apamaは、総合プラットフォームの一部として、システムが受信したロー・イベントと、Apama独自の分析処理で生成されるすべてのイベントを取得するリアルタイム・イベント・データベース、EventStoreを提供します。

イベントの属性と属性間の時間的な関係を捕捉し、時系列でイベントを記録するEventStoreは、任意の長さのイベント・シーケンスを再生できるTiVoのような機能を提供します。この再生機能を補完するのが「早送り」操作モードです。早送り操作モードでは、時間のかかるイベント・シーケンスの実行を早送りできるので、すべてのイベントの時間的シーケンスを維持しながら、テストをより迅速に実行することができます。コアのApamaイベント・プロセッシング・アーキテクチャと同様に、EventStoreは、共通のリポジトリ内で様々なタイプのイベントを管理できるため、実世界でのApamaの実行結果がそのままイベント履歴に補完されます。

EventStoreにより、組織は以下のことが可能になります。

- ⇒ イベント・シーケンスをApamaトランザクションの監査証跡として取得し、因果関係を割り出したり、イベントの発生順序を確認したりできます。エンドユーザや分析アプリケーションは、必要な時間パラメータを設定することで、イベント・ストリーム・プロセッシング履歴を確認できます。
- ⇒ 本番環境への配備の前に、イベント履歴を利用して、新しいイベント・プロセッシング・シナリオ（または既存のシナリオのバリエーション）を徹底的にテストできます。配備予定のシナリオをテストするシミュレーション環境にイベント・シーケンスを供給し、実世界のデータを活用してこのシナリオの高度なwhat-if分析を行うことができます。

イベントの多様性をサポートするアーキテクチャ: Apamaアダプタとセマンテック・マッピング

イベント・プロセッシング・プラットフォームがサポートしなければならないイベントの種類は、このプラットフォームを配備する組織のビジネスと同様に多岐にわたります。狭く定義されたイベント（例：ティック・データのみ、RFIDデータのみ）のために最適化されたスケーラブルなアーキテクチャを提供することも可能ですが、イベント・プラットフォームの真価は、単一のアプリケーション内で多種多様なイベントとフォーマットをサポートできる能力にあります。たとえば、プラント制御アプリケーションの場合には、温度、湿度、気圧を捕捉するセンサー・データが含まれ、非常に多くの属性が存在します。金融業界では、クロス資産トレーディングのために、イベント・プラットフォームは、株式、債券、デリバティブをはじめ多数のデータ・タイプをサポートしなければなりません。イベント・プロセッシング・プラットフォームは、いずれの場合も、1つの共通のプラットフォーム内で多様なフォーマットをサポートする必要があります。

Apamaプラットフォームは、イベント・フォーマットから接続を分離する独自のIntegration Adapter Framework (IAF)を提供します。IAF Adapter は、2種類の統合サービスを提供します。

- ⇒ トランスポート統合サービス – APIを通じ、特定のタイプのネットワークやデータ・サービスへの双方向の接続性を提供します。たとえば、Sonic ESB、SonicMQ®、TIBCO Rendezvousなどのトランスポート、ODBCをサポートするデータベース、業界固有の様々なイベント・ストリーム（例：金融業界のReuters、GL Trade、FX、FIXなど）用の即座に利用可能なアダプタを提供します。
- ⇒ セマンテック・マッピング・サービス – IAFは、XMLマッピング・ルールを利用して、受信するトランスポート・メッセージのフィールドをユーザが設定したイベントに動的にマッピングします。

したがって、1つのタイプのイベント処理に特化した複数のアプリケーションを使用する必要はありません。Apamaプラットフォームは、単一のイベント駆動型アプリケーション内で多種多様なイベントの高度な分析をサポートします。

俊敏性を提供するアーキテクチャ – Apama Scenario Modeler

どのようなテクノロジー・プラットフォームも、アプリケーションの開発が非常に困難で多大な時間を要する場合、その価値は大幅に低下してまいります。開発段階のアプリケーションは、ビジネス・オペレーションに一切競合優位性を提供しません。リアルタイム・プラットフォームの配備を検討している組織は、候補となっている各プラットフォームで、イベント・プロセッシング・アプリケーションをいかに

迅速に配備できるかも考慮しなければなりません。また、イベント・プロセッシング・アプリケーションは本質的に反復的に動作しますが、監視が必要なイベントや、イベントに応じて実行すべきアクションが絶えず変化するため、その変化に合わせてアプリケーションを進化させる必要があることも認識する必要があります。複雑なコード主導型の開発プロセスでは、組織が変化するビジネス環境に対応する新しいアプリケーション・バージョンを実装する能力が低下してまいります。

Apama イベント・プロセッシング・プラットフォームは、統合開発環境である Apama Event Modeler (図4を参照) を提供し、高度なイベント・プロセッシング・アプリケーションの迅速な設計、配備、進化をサポートします。Apama Event Modeler を使用すれば、ビジネス・ユーザでもイベント駆動型アプリケーションの作成と修正を行うことができます。した

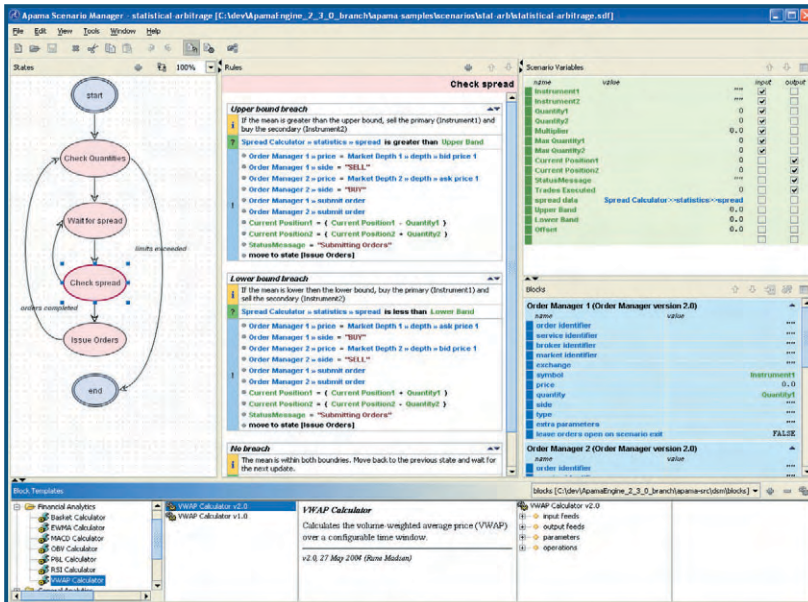


図4：Apama Scenario Modeler

たがって、低水準のコーディングのみを使用して開発に数週間から数ヶ月かかるアプリケーションと同等の機能を持つイベント・ストリーム・アプリケーションを、数時間から数日で開発することができます。

Apamaのユーザは、グラフィカルな状態図と表示されるロジック・フローを活用して、非常に高度なイベント・プロセッシング・シナリオを作成できます。監視が必要な時間とデータのフロー・パターン、制約、イベント値は、ポイントアンドクリック方式の組み立てプロセスを通じて表現できます。Scenario Modelerは、アクションをトリガーするイベントと、講じるアクションを定義するイベント間の高度なイベント相関関係を捕捉します。また、Event Modelerでは、ビジネス・ユーザが利用できる直感的なツール内で「監視、分析、アクション」までのオペレーション・シーケンスを作成することができます。低水準のコーディングに依存せずに、完全なイベント・プロセッシング・シナリオを作成し、テストを実施し、配備することが可能になるのです。

再利用のためのアーキテクチャ – Apama SmartBlocks

グラフィカル環境の有用性を飛躍的に向上させるのが、Apama Event ModelerのSmartBlocksです。SmartBlocksは、ポイントアンドクリック式のインタフェースを通じてイベント・プロセッシング・アプリケーション内に簡単に組み込むことができる、再利用可能なイベント・プロセッシング・ロジックです。SmartBlocksを利用すれば、低水準のプログラミングを一切必要とせずに、新しいイベント・シナリオに事前定義されたロジックを挿入することができます。Apama Event Modelerには、SmartBlocksとして組み込まれたパッケージ済みのイベント・オペレーションが含まれています。ユーザは、シナリオ開発のプロセスで、これらのSmartBlocksの出力パラメータを定義し、イベント・プロセッシング・シナリオに組み込むことができます。

パッケージ済みのSmartBlocksは、再利用可能なイベント・ロジックの唯一の基盤です。イベント・プロセッシング・アプリケーションの実装は、本来進化していくプロセスであり、お客様がカスタム開発したSmartBlocksを実際のイベント・プロセッシング・シナリオに配備できるようにするためには、独自のSmartBlocks作成ツールをお客様に提供しなければなりません。Apamaアーキテクチャは、このことを前提に構築されています。

したがってApamaには、新しいSmartBlocksの作成と設定をグラフィカル環境で実施し、イベント入力属性、実行する機能、出力属性を指定できるBlock Builderツールが含まれています。このツールにより、お客様は導入後即座に使用可能な標準の機能パレットを、市場/お客様が作成した機能で拡張することができます。作成する機能は、必要なランタイム実行速度を実現するイベント・プロセッシング・シンタックスで表現し、視覚的にアクセスできるようにパッケージ化できます。したがってビジネス・ユーザは、シナリオのモデリングおよび実装プロセスでこれらの機能にアクセスできます。

柔軟な実行のためのアーキテクチャ – Apama Dashboard

Apamaは、パフォーマンスと開発の面で優位性を提供することに加え、3つのタイプの使用モデルに対応する実行環境の柔軟な選択肢を提供します。

- ⇒ Apama Dashboard – 通常、イベント・プロセッシング・アプリケーションには、ダッシュボードを介してアクセスします。ダッシュボードは、ユーザがイベント・プロセッシング・シナリオの開始/停止、シナリオ実行を管理するパラメータの入力/変更、実行中のオペレーションの監視、実行中のアプリケーション内のトリガーに基づいた警告の受信に使用するインターフェースです。Event Modelerツール内からの配備を容易にするために、ダッシュボード・ウィザードがブラウザベースのダッシュボードのルックアンドフィールを「ワンクリック」のステップバイステップ方式で定義できるインターフェースを提供します。ユーザは、このツールを使用して入力/出力および様々なレイアウト・オプションを定義できることに加え、サード・パーティ製のWebカスタマイゼーション・ツールを使用してユーザ・インターフェースのルックアンドフィールをさらにカスタマイズすることもできます。
- ⇒ カスタマイズされたインターフェース/サード・パーティ製インターフェース – 多くの組織がイベント・プロセッシングとともに他のアプリケーションの配備を必要とすることを想定し、Apamaにはダッシュボード機能へのアクセスを提供する完全なダッシュボードAPIセットも組み込まれています。したがって組織では、導入時の設定のままWebダッシュボードを使用するのではなく、高度にカスタマイズされたインターフェースを実装することができます。または、サード・パーティ製インターフェースにApamaを統合し、サード・パーティ製アプリケーション内からイベント・プロセッシング・サービスを呼び出すことも可能です。
- ⇒ インターフェースなし – イベント・プロセッシング・アプリケーションにはユーザ・インターフェースを必要としないものも存在します。このようなアプリケーションの配備のために、Apamaは、他のアプリケーション内から直接呼び出してイベント・プロセッシング・サービスを開始/停止するためのAPIを提供します。C++およびJava®で利用可能なこの機能により、開発者はCorrelator内でのイベント・プロセッシングの実行を完全に制御することが可能になります。外部アプリケーションは、MonitorScriptおよびイベントを使用してイベント・アプリケーションを作成でき、イベント・サービスで発生するイベントに反応できます。

結論

情報システムがかつてないほど多くの要素が相互接続される環境をサポートできれば、イベント・ストリーム・プロセッシングの価値が向上します。このような新しい相互接続の副産物として産まれるのが、膨大なイベント・ストリーム・データ、つまりこれらのシステムの接続点で発生する膨大なイベントです。これらのイベントを監視、分析し、イベントに基づいてアクションを講じる機能は、差別化要素や競合優位性が一時的にしか得られない世界でビジネスを展開する企業にとって、大きな鍵となるでしょう。同様に、脅威の検知と撃退のスピードがものを言う世界でも、イベント・プロセッシングは迅速な対応を実現するソリューションの重要なコンポーネントとなるはずで

Apamaの独自のイベント・プロセッシング・プラットフォームは、イベント・プロセッシング・アプリケーションが要求するパフォーマンスと俊敏性を提供します。したがって組織では、迅速に機会（または危険な脅威）に対応することが可能になります。イベントは、高速に到着し、変化率も高いことから、「捕捉することが難しい」と思われがちですが、適切なツールがあれば、組織はこの環境を活用して競合優位性を得ることができます。捕捉が困難とされるイベント内にもパターンが存在しています。しかし、そのパターンを発見するには、適切なツールが必要です。

この適切なツールがApamaなのです。Apamaは、以下のような特徴を持つリアルタイム・イベント・プロセッシング・プラットフォームです。

- ⇒ イベントベース・コンピューティングのために特別に設計されている
- ⇒ 数千のイベントと、並行処理されるイベント・シナリオのすべての実行をミリ単位で実現する比類のないパフォーマンスを提供する
- ⇒ イベント・ストリーム・プロセッシングに不可欠な時間的な属性を取得する
- ⇒ プロセス内でのビジネス・ユーザによる実装を高速化するグラフィカルな開発/配備モデルを提供する
- ⇒ イベント駆動型ソリューション内で対応しなければならない多数のイベントをサポートする
- ⇒ 履歴データの監査と新しいイベントベース・アプリケーションのテストに使用するイベント・リポジトリを提供する

Apamaは、発生して長時間経過している「何か」の確認だけでは不十分と考えている組織に、イベント・プロセッシング・パワーを提供します。つまりApamaは、現在発生しているイベントに反応するパワーを提供するのです。



ソニック ソフトウェアについて

ソニック ソフトウェア株式会社は、米国プログレス ソフトウェア・コーポレーションの日本法人です。エンタープライズ規模のメッセージングシステム「SonicMQ®」と、SOA構築のために業界で初めてエンタープライズ・サービス・バス (ESB) を実装した注目の「Sonic ESB®」をベースとする企業システム統合ミドルウェア製品、そしてC++/Java開発環境を支える業界トップのオブジェクト・データベース管理システム「ObjectStore®」ファミリなど、ユニークで競合力の高い製品を日本市場に展開しています。

ソニック ソフトウェア株式会社

〒102-0082 東京都千代田区一番町18番地 川喜多メモリアルビル
Tel : 03-3556-7610 Fax : 03-3556-7642
<http://www.sonicsoftware.co.jp> email: info@sonicsoftware.co.jp

ObjectStore、Apama、SonicMQ、Sonic ESBは、プログレスソフトウェア・コーポレーションの米国およびその他の国における商標または登録商標です。この文書に含まれるその他の登録商標やサービスマークは、それぞれ各社に帰属します。