Event Processing with State Machines
Table of Contents

3 | Introduction
3 | Real-Time Operational Visibility—A Solution to Important and Complex Problems
4 | Operational Visibility—Using Event- and Rule-Driven State Machines
5 | Entity Life Cycling with TIBCO State Machines
6 | Event- and Rule-Driven State Transitions
7 | SLA Management—The Power of the Missing Event
7 | Tracing with Entry Action and Exit Action
8 | Summary
Introduction

In today’s fiercely competitive world, providing superior, on-time customer service is key to success. To meet or beat service level agreements (SLAs), organizations need complete, real-time visibility into their operations.

Most business operations consist of a number of automated and manual business processes that span multiple applications running on different types of systems. Couple this complexity with a high volume of operations, and process management becomes a difficult problem.

This paper discusses the TIBCO event-driven approach to real-time operational visibility using the state machines functionality in the TIBCO event processing platform. State machines are also used for managing telecommunications networks and modelling payment transactions.

Real-Time Operational Visibility—A Solution to Important and Complex Problems

Much of the complexity surrounding business processes has to do with the history of the company’s IT operations. As processes expand over time and new applications are added, business processes spread across a number of different applications that are each good at solving a part of the overall problem. This fragmentation makes it difficult to keep track of processes end-to-end and to diagnose problems. Opportunities are lost, and efficiency suffers.

For example, before a quote could be generated at an insurance company, policy applications went through eight processes, each running on different software applications on different machines. The company was finding it extremely difficult to track the applications as they moved from one software application to another. Policy applications that were stuck were often discovered only when the customer called in to inquire about status.

At a large telecommunications company, hundreds of customer orders were getting suspended every day, and no one could figure out where or why. Salespeople were getting frustrated, as were customers.

In both cases, real-time visibility into operations was the change needed. Not only is it important for a company's internal operations, but if done well, real-time visibility also provides a serious competitive advantage. For example, as jobs flow through the process at an electronics repair company, customers get real-time status updates, leading to significant improvement in customer satisfaction.

With the ability to track packages, customers get complete visibility into their expected delivery. We take it for granted now, but some years ago this was a dream. It all started because private logistics companies needed a competitive advantage to differentiate from government-owned postal services.
Operational Visibility—Using Event- and Rule-Driven State Machines

To meet service level agreements in a world where customers get frustrated easily and have plenty of other options, start with complete end-to-end operational visibility. It includes the ability to track-and-trace entities, such as insurance applications, repair jobs, and packages. A good track-and-trace application would not only track the flow of the entity along its lifecycle, but also define SLAs for each stage in the lifecycle, monitor those SLAs, and raise alerts immediately if an SLA was violated.

State machines are an excellent paradigm for modeling entity lifecycles. The progress of an entity across its lifecycle stages can be represented with states in the state machine. The state machine can move from one state to another when new events come in, when there is a change in a property of the entity, or when a certain amount of time passes.

As an entity moves through states in the state machine, its progress is logged for traceability. The state machine can also generate events that can be put on an enterprise service bus or saved in a data store for display in a tracking system such as a real-time dashboard. State machines are also excellent tools for monitoring SLAs because they can send an alert if an entity stays in a state for too long.

To use state machines for real-time operational visibility, applications used by the entity need to be event-enabled. For example, at an insurance company, the eight software applications used for processing policy applications need a mechanism to notify the state machine. Whether this happens at the beginning or end of the processing, or both, the state machine needs to track the progress of the insurance application. In fact, the state machine can be used to move data from one computer application to the next. Even applications that use databases to maintain state or that run on legacy systems such as mainframes can potentially use event-enabling technologies to publish state changes.

If a state machine can perform multiple transactions, rules can be used to determine the correct transition. For example, life insurance policy applications greater than $1 million dollars could be sent through special expedited processing. Retail orders could be routed differently using rules that are defined based on customer category, location, or items in the order. A state machine that monitors and manages a complex lifecycle needs a sophisticated rules engine that can execute rule- and event-driven state transitions.

To handle a high volume of transactions, state machines also need enterprise-grade capabilities such as fault-tolerance and load balancing.
Entity Life Cycling with TIBCO State Machines

In the TIBCO event processing platform, a concept is a data-structure that can be used to represent an entity whose state needs to be maintained over time for event correlation. A customer order or an application for insurance is an example of a concept.

TIBCO provides an in-memory data grid (IMDG) that is built into its event processing platform and can be used to store a large number of concepts spread across multiple machines. These concepts can be stored in memory and optionally persisted on local disks or a database. The IMDG also provides enterprise grade capabilities for the state machine, including fault-tolerance, load balancing, and persistence.

A large percentage of TIBCO event processing customers use state machines for obtaining real-time visibility into their operations because they are fully event-driven and rules-based. One or more state machines can be associated with a particular concept. One of the state machines can be declared the main machine for that concept, and if a concept has a main state machine, the machine starts as soon as the concept’s instance is created.

As soon as a new order is created in the system, the state machine tracking the order lifecycle can be started automatically. If a main state machine is not assigned to a particular concept, any state machine associated with the concept can be started programmatically, for example, in the action block of an inference rule using the catalog function included with the product.

Every state machine has a start state and one or more end states. There can be multiple states in between the start state and the end state. State machines can consist of simple states that define the flow of the concept from beginning to end. Simple states can be grouped into a composite state, execute in parallel as a group of concurrent states, or call on other state machines. The TIBCO event processing platform provides an Eclipse-based graphical modelling environment that can be used to design state machines.

---

State Machines in the TIBCO Event Processing Platform
Event- and Rule-Driven State Transitions

In the TIBCO event processing platform, state machines are fully rules-based and event-driven. As the state transitions, you can use inference rules to relate the concept to an incoming event.

As example, if a state machine is tracking an insurance application that is waiting on a check of the customer’s driving history, it can move to the next state when it receives an event from the application performing the check. The content of the event can be used to determine the next state. For instance, if the customer’s driving history is excellent, the insurance application may be sent to another department for further processing that will look at the customer’s health records and determine the risk. If the customer’s driving history is not great, but not too bad either, the application may be sent for further investigation. If the customer’s driving history is very bad, the application may be rejected right away, moving the state machine to its end state.

At every transition, the state machine can wait for one or more inbound events and can evaluate conditions inside rules when an event arrives, determining the next state to transition to. Since the state machine functionality is embedded and closely integrated within the event processing platform, it is possible to receive events that drive the state machines from a variety of different transports, including the enterprise service bus. State machine transitions also have access to the powerful inference rules and decision capabilities of the TIBCO event processing platform.
SLA Management—The Power of the Missing Event

One of the primary uses of state machines is to monitor SLAs while tracking an entity’s lifecycle. If an insurance application is moving through eight processes that run on eight software applications, the company would be able to enforce its SLAs for responding to the insurance applicant, but only if they are aware of a potential violation of the SLA, for example, the application is stuck.

Within TIBCO state machines, every state can be configured to have a timeout. If the expected event or state change doesn’t happen within the timeout period, the state machine can be configured to timeout at that particular state. When a time-out situation occurs, the state machine can transition to a special state designed to handle exceptions, such as delays in processing. The timeout action can also raise the necessary alerts in the form of outbound events that can then be received by other applications or processes that will handle the exception. Timeouts can be configured for individual states, and even for the state machine as a whole if the sum of all the steps is expected to take less time than the total time allocated to all individual states. State machine timeout can be treated the same way as an individual state timeout.

SLA Management—Capturing Missing Events with State Timeouts

Tracing with Entry Action and Exit Action

Every state inside a state machine in the TIBCO event processing platform has an entry action and an exit action. As the names suggest, the entry action executes as soon as the state machine enters that particular state, and the exit action executes when the state machine leaves that particular state. For the purposes of auditability, these actions can be used to log information about the state machine. Information such as the time the state machine entered the state and the time it left the state can be useful.
Summary
State machines are a powerful event processing tool for tracking and tracing the life cycles of entities. They can monitor missing events and raise alerts in real time while maintaining complete auditability. With a fully event- and rules-driven model, state machines provide complete, real-time operational visibility.