Paper SAS3201-2019

Creating a Strong Business Case for SAS® Event Stream Processing

Henrique C. Danc, SAS Institute Brazil, Sao Paulo, Brazil

ABSTRACT

SAS® Event Stream Processing has a set of great capabilities that both IT and business areas can easily benefit from. This technology enables seizing opportunities and spotting red flags that usually are hidden in the massive torrent of fast-moving data that flows through our business, without the need for storing any data, by applying simple business rules, pattern detection, geofencing, and real-time training and scoring of analytical models. However, it is not always easy to translate these specific features into direct business benefits for business groups, which usually hold the budget for projects like these. Therefore, it is crucial to demonstrate to business users how they will benefit from this technology and how the features will help them take faster action and achieve better results. This paper guides you through a process to create a strong and persuasive business plan that translates the technology features from SAS Event Stream Processing to business benefits, including a suggested roadmap for adopting SAS Event Stream Processing features.

INTRODUCTION

Traditionally, when we think of analytics, the most common way of creating models is to execute ETL (extract, transform, load) jobs to access stored data from several sources, transform it, write it to a centralized data store, execute further data manipulation to create ABTs (analytical base tables, the model input data), train analytical models, and then deploy the models into production. The deployed models can then be executed on a periodic basis to produce a score based on this historical data, as we can see in Figure 1.

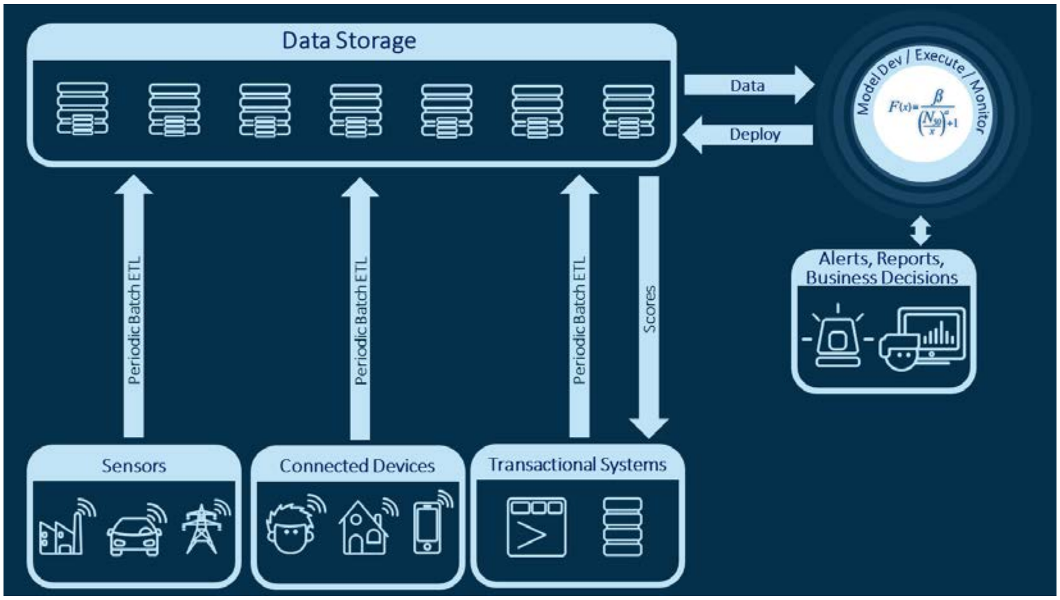

Figure 1. Traditional Analytics Architecture

This is known as data-at-rest model creation and execution. The calculated scores can then be used by a transactional system or application to drive a business process or decision. For example, what is the propensity of a given customer to buy a specific product? What is the probability of a given behavior to be a fraud? among others. We can also monitor the scores utilization, create reports and alerts, and use this information to drive business decisions.

Although this approach is still important, there are situations where businesses can benefit from a more real-time action on the data.

One step toward this direction is to use the data-at-rest trained model and score it in real time, by using a real-time decision engine as shown in Figure 2.

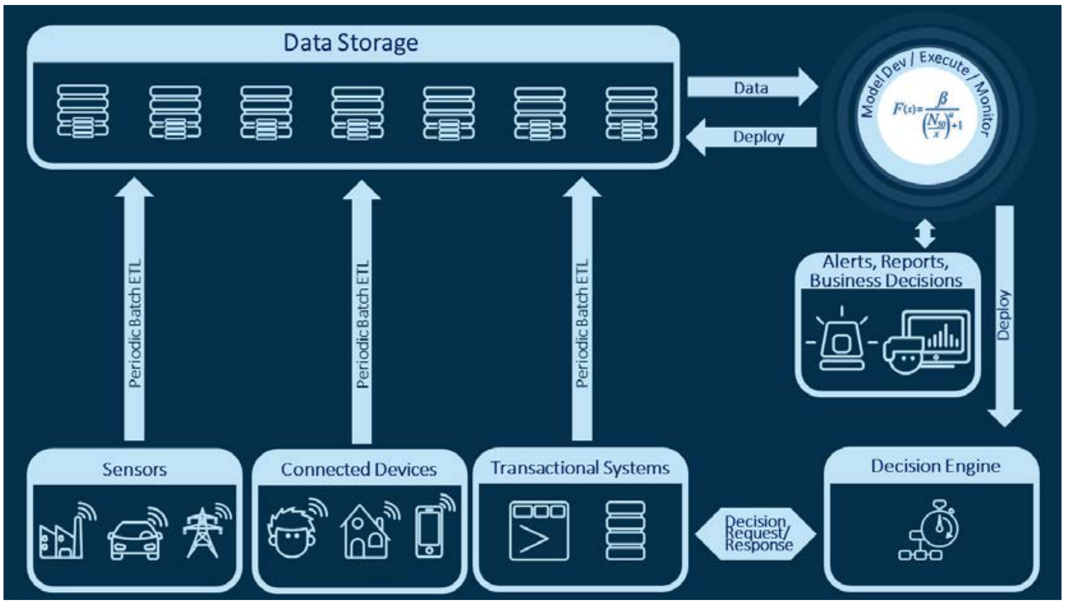

Figure 2. Traditional Analytics plus Real-Time Decisioning Architecture

In this scenario, a system or application (examples are a website, a mobile app, and a call center, among others) directly requests a decision from the decision engine based on some input variables. This makes sense when the model input variables change during a business decision, and this change might influence the score output of the model. Commonly, we use models and business rules inside these decision flows, and we can also use more information if needed, like a web service or database access.

This approach can potentially present gains, because the decision logic is centralized regardless of where it is applied. In addition, changes in the rules or models are easier to make in this type of architecture. But, this approach by itself is also limited, in the sense that it replies only when it is called.

So how can we make it better?

We can analyze everything that happens through our business, aiming to detect patterns of interest, what is relevant and should be acted on now, and what is not relevant and can be safely discarded. We can use several techniques to help in this decision, like scoring analytical models, text analytics, geofencing, pattern detection, among others, as shown in Figure 3.

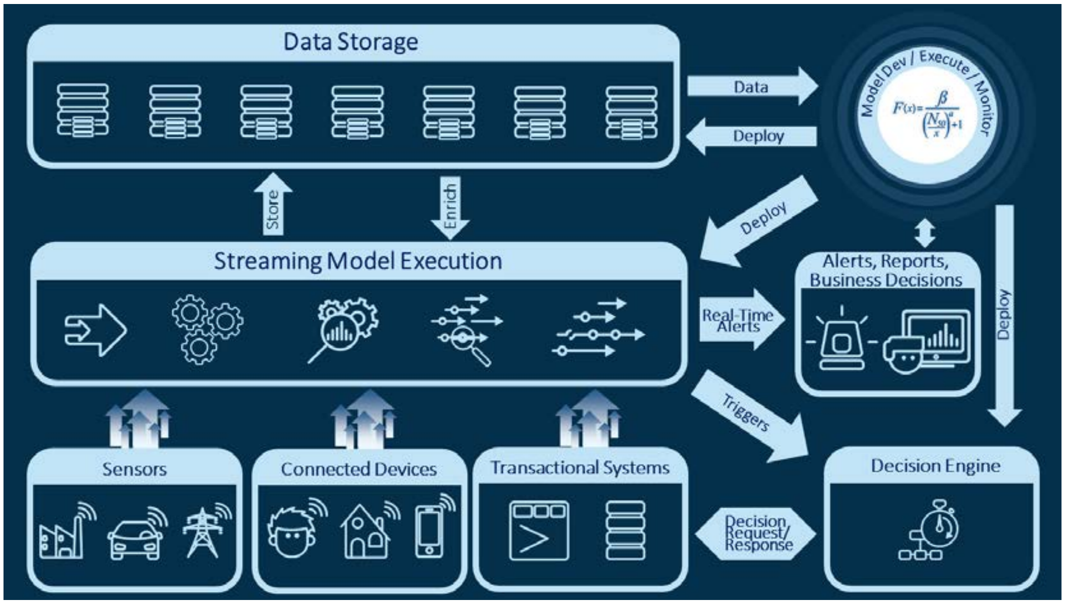

Figure 3. Traditional Analytics, Real-Time Decisioning, and Stream Processing Architecture

The difference from the previous approach is that in this one we analyze what happens in a passive way, that is, independent from a request of any kind and from any system or application. When we detect what is relevant based on the rules, models, and other techniques, we can act upon it. This is what we call stream processing.

In this approach we can also replace the periodic batch extraction part of the ETL processes with our streaming engine as we can see in Figure 3. As the engine receives events in real time, it can forward these events for storage, eliminating the need for another process.

The approaches seen above in all three figures can and usually are used in conjunction, so that they can complement each other. For example, we can use the streaming engine to detect what is relevant and then request a decision from the decision engine to select what to do in this specific scenario.

This paper will explore the most important SAS Event Stream Processing features and then translate these features into business benefits. We will also provide a suggested implementation roadmap for Event Stream Processing.

SAS EVENT STREAM PROCESSING FEATURES

SAS Event Stream Processing is a streaming engine that was engineered from day one to be fast – capable of processing millions of events per second with low latency – so that we can apply calculations, perform manipulations, and apply analytics on streams of events.

Streams of events or simply streams are records of what happens in our business. It can be a website or mobile app login or navigation, a purchase, a sensor reading, an opened email, a call center contact, and a location change, among others. It is basically a record of data. But, it is called stream because unlike traditional ETL processes – where usually a job runs periodically to extract, transform, and load data – we capture it as soon as it happens. There is no periodic job. Instead, we store the query and it is always on for processing these events.

We can start with doing simple event processing on streaming data – like calculations, filters, and aggregations, that sometimes can bring value for our business – and then start increasing our flow complexity by adding data quality manipulations, time-pattern detection, text analytics, streaming analytics and machine learning, and geofencing.

We can do all this using a graphical flow designer or using a provided Python API.

The following section will introduce these features by explaining them in a simple way. It is intended to provide basic understanding of the existing Event Stream Processing technical features that will then be mapped into business benefits in a later section – so do not worry too much about the specifics of the technical details – just focus on the overall functionality.

EVENT PROCESSING

These are the most basic manipulations that can be made with streaming events. It includes computations, filters, and aggregations.

Computations enable a one-to-one transformation of input events to output events. Computations can be used to standardize several stream layouts into one target layout and to calculate columns for augmenting input events.

Filters let only events that satisfy a given condition to continue to pass along to the flow.

Aggregations allow us to create aggregate views or metrics of our input streams. But, different from aggregations done in batch, in the streaming world we do real time or streaming aggregation. This means that the aggregated values are recalculated every time a new event comes in. This can be overall or for a specify subject – like a customer. We can specify the different aggregation periods for the same input data – for example, data consumption in the last hour and the last three hours, and so on.

DATA QUALITY

Data quality is an important aspect of data governance, which can benefit from the analytical life cycle, as well. It allows us to monitor, identify, and standardize the data that is in different formats between our transactional systems – that otherwise would be unrelated data. It also allows us to join data from different sources even if they don't have a common key between them, based on a specified degree of sensitivity.

We can apply data quality techniques in streaming data, and this can be a part of a data governance policy. It can also be the starting point of the collection of data for creating more accurate analytical models.

In-stream data quality uses the same operators and components of SAS® Data Quality, including SAS® Quality Knowledge Base for Product Data and SAS® Quality Knowledge Base for Contact Information. The quality knowledge base (QKB) is a collection of files that store data and logic that define data management operations. SAS products like SAS Event Stream Processing reference the QKB when performing data management operations on your data.

Each QKB supports data management operations for a specific business area. SAS® Quality Knowledge Base for Contact Information supports the management of commonly used contact information for individuals and organizations like names, addresses, company names, phone numbers, and more.

Data and logic in the QKB are organized into a set of definitions. Each definition includes a single context-sensitive data management operation. For example, a definition in the QKB might contain data and logic used to parse phone numbers. Another definition might contain data and logic used to determine the gender of an individual's name.

You can use definitions in the QKB as delivered by SAS, or you can customize the QKB by modifying definitions or creating new definitions for use with your own business data. Since the QKB is used by all SAS products, changes made to the QKB are automatically available to your entire enterprise.

In Table 1 we summarize the commonly used data-quality techniques that are available in SAS Event Stream Processing.

Technique	Description
Casing	Applies casing rules (upper, lower, or proper) to a string. The function also applies context-specific casing logic using a case definition in SAS Quality Knowledge Base.
Gender Analysis	Determines the gender of an individual's name using a gender analysis definition in the QKB.
Identification Analysis	Analyzes identification using context - for example, determine whether a string is a name, an address, a phone number, or other available types.
Match Code Generation	Generates a match code for a string using a match definition in the QKB.
Pattern Analysis	Generates a pattern for a string using a pattern analysis definition in the QKB.
Standardization	Standardize data using a standardization definition in the QKB.

Table 1. Commonly Used Data Quality Techniques available is SAS Event Stream Processing

IN-STREAM PATTERN DETECTION

We can detect patterns of interests in real time using SAS Event Stream Processing.

To do this, we define events of interest – that is, what must happen or must not happen for a pattern to be considered satisfied – and then we connect these defined events of interest into an expression that uses special logical operators and optional temporal conditions.

In Table 2 we summarize the supported operators for pattern detection.

Operator	Description
and	All its operands are true.
or	Any of its operands is true.
fby {optional temporal condition}	Each operand is followed by the one after it. If a temporal condition is specified, it is also taken into consideration.
not	The operand is not true.

notoccur	The operand never occurs.
is	The operand ensures that the following event is as specified.

Table 2. Supported Operators for Pattern Detection

These operators allow us to monitor patterns such as the following:

- The customer contacts the call center more than once in 30 minutes.
- The customer logs in, puts some products in the shopping cart, but does not complete a purchase in that day.
- The customer still contacts our company about the same issue as last month.

TEXT ANALYTICS

We can perform text analytics on stream data. It is a way of extracting value from unstructured texts that flow throughout our business.

The techniques that are available are listed in Table 3.

Technique	Description
Text Category	Categorizes a text field in incoming events. The text field can generate zero or more categories, with scores.
Text Context	Abstracts classified terms from an unstructured string field. Use the Text Context window to analyze a string field from an event's input to find classified terms.
Text Sentiment	Determines the sentiment of text in the specified incoming text field and the probability of its occurrence. The sentiment value is "positive," "neutral," or "negative." The probability is a value between 0 and 1.
Text Topic	Runs Text Topic models, generated by SAS® Visual Text Analytics or SAS® Visual Data Mining and Machine Learning, on events to score and identify themes in streaming text.

Table 3. Supported Text Analytics Techniques

IN-STREAM ANALYTICS AND MACHINE LEARNING WITH MODEL SUPERVISION

SAS Event Stream Processing allows in-stream scoring of models trained on historical data-at-rest. It also allows in-stream training and scoring of analytical models and techniques. For both scenarios, several techniques are available, and we can control which model to execute on-the-fly, without any downtime, manually or automatically based on business rules or triggers – this is called model supervision.

Online Models

Online models use algorithms that are packaged with SAS Event Stream Processing Analytics and that are trained in SAS Event Stream Processing projects.

There are two types of online model algorithms:

- Train and Score: training and scoring are performed at different points with streaming data.
- Calculate: the algorithm or technique are applied directly to streaming data.

Below we list the algorithms and techniques that are available.

Online Train and Score Algorithms

- Streaming K-Means Clustering
- Streaming DBSCAN Clustering
- Streaming Linear Regression
- Streaming Logistic Regression
- Streaming Support Vector Machines

Online Calculate Algorithms

- Streaming Summary (Univariate Statistics)
- Streaming Pearson's Correlation
- Segmented Correlation
- Streaming Distribution Fitting
- Short-Time Fourier Transform
- Compute Fit Statistics
- Compute ROC Information
- Calculate a Streaming Histogram
- Streaming Text Tokenization
- Streaming Text Vectorization
- Image Processing Algorithm
- Moving Relative Range
- Term Frequency — Inverse Document Frequency
- Lag Monitoring
- Cepstrum Transform
- Video Encoding
- Subspace Tracking

Offline Models

Models are referred to as offline when they are specified, developed, trained, and stored outside of SAS Event Stream Processing. The following algorithms are available as offline models.

- Bayesian Network
- Butterworth Filter
- Factorization Machine
- Gradient Boosting Tree
- Random Forest
- Robust Principal Components Analysis
- Stability Monitoring Scoring
- Support Vector Data Description
- Support Vector Machine
- Dirichlet Gaussian Mixture Model
- Generalized Linear Multi-Task Learning
- General Linear Model
- Generalized Linear Regression Model
- Generalized Additive Models
- Deep Neural Networks (DNNs)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Recommender system

IN-STREAM GEOFENCING

SAS Event Stream Processing supports location analytics. This is done through geofencing.

A geofence is a virtual perimeter for a real-world geographic area. You can dynamically generate a geofence as a radius around a specific location or create one as a set of specific boundaries. Areas and locations of interest are defined as geometries. The Geofence window – the object that performs these types of analyses inside the project – supports the following geometries: polygons, polylines, and circles. Geometries are published as events, one event per geometry.

Polygon

A polygon is a plane shape representing an area of interest. The Geofence window supports polygons, multi-polygons, and polygons with holes or multiple rings.

When working with polygons, the Geofence window analyzes each event position coming from the streaming window and returns the polygon in which this position is located. If there are multiple matching geometries (in cases of overlapping polygons) then multiple events (one per geometry) can be produced.

Polyline

A polyline is a shape representing a border or a tripwire that is defined as a list of position coordinates that represent the polyline's segment or segments. The sequence of the points in the segment definition defines the polyline's vector direction and its left and right side.

When working with polylines, the Geofence window analyzes each event position that comes from the streaming window. It returns the ID of the polyline that is within the distance defined by the radius property value. Distance is measured from the position to the closest segment. When a polyline is detected, the Geofence window returns the exact distance from the position to the polyline. This value is negative when the position is on the left side of the polyline. It is positive when it is on the right side. A tripwire crossing can then easily be detected by a sign change using a Pattern window.

Circle

A circle encompasses the position of a location of interest and is defined with three values: two coordinates, X and Y (longitude and latitude), that represent the center of the circle and a radius distance around the center.

When working with circles, the Geofence window analyzes each event position coming from the streaming window and returns the circle ID that matches the following criteria:

- If the position lookup distance is set to 0, then the position behaves like a simple point. It is either in or out of the circle. If it is in the circle, there is a match.
- Similarly, for circle geometries, if the circle radius is set to 0, then the circle behaves like a bare point. This point must be in the position lookup distance area to have a match.
- For any other values of the position lookup distance and the circle radius, the position and the circle's center must be in each other's distance. Otherwise, there is no match. The position is within the circle and the distance between the circle's center and the position is lower than the lookup distance.
- If both the position lookup distance and the circle radius are equal to 0, then they must be the exact same point to have a match.

OPEN-SOURCE INTEGRATION

SAS Event Stream Processing supports open source and other languages integration for three distinct objectives:

- Real-time in-stream scoring of Python and C models.
- Programmatic creation of SAS Event Stream Processing projects in Python.
- Custom integration using Python, Java and C.

EASY-TO-USE GRAPHICAL DEVELOPMENT AND DEPLOYMENT INTERFACE

SAS Event Stream Processing comes with a powerful modeling interface, SAS® Event Stream Processing Studio, which leverages drag-and-drop components for creating projects in a graphical way as we can see in Figure 4. This way, you don't need to code. You can code, but only if want to.

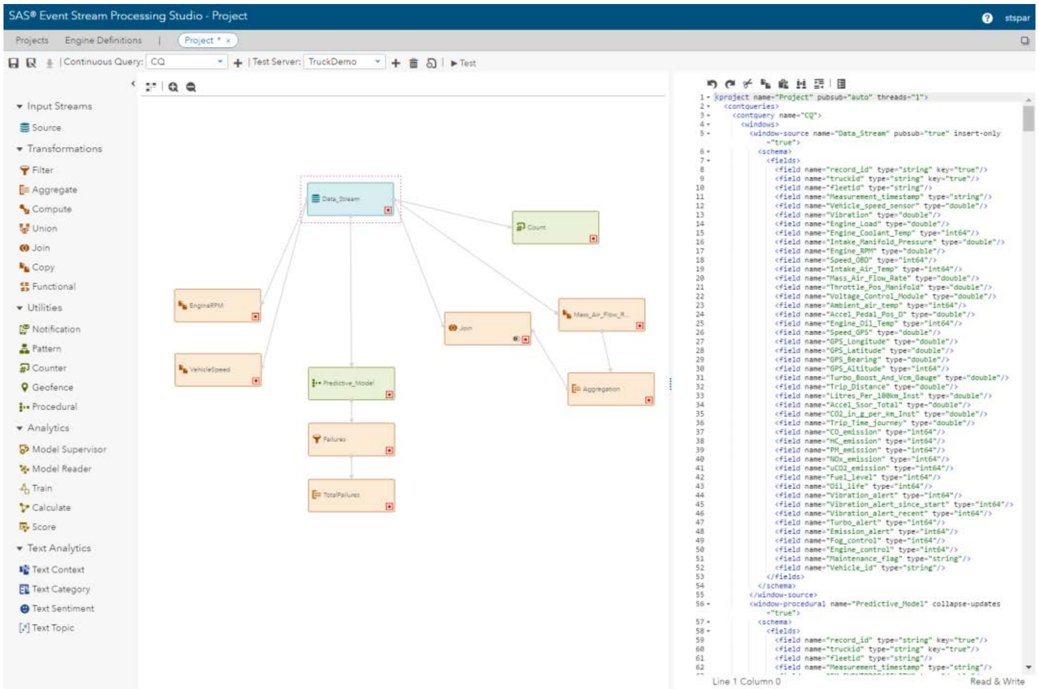

Figure 4. SAS® Event Stream Processing Studio

It also features an add-on, SAS® Event Stream Manager, that can be used for project deployment and monitoring as shown in Figure 5 and Figure 6. It uses a graphical interface to keep all deployment and monitoring operations centralized in one single place.

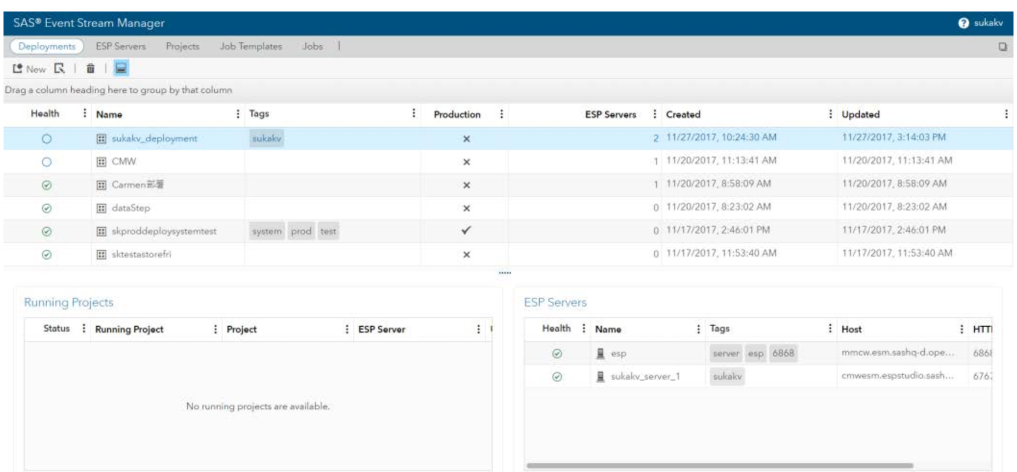

Figure 5. SAS® Event Stream Manager Deployments

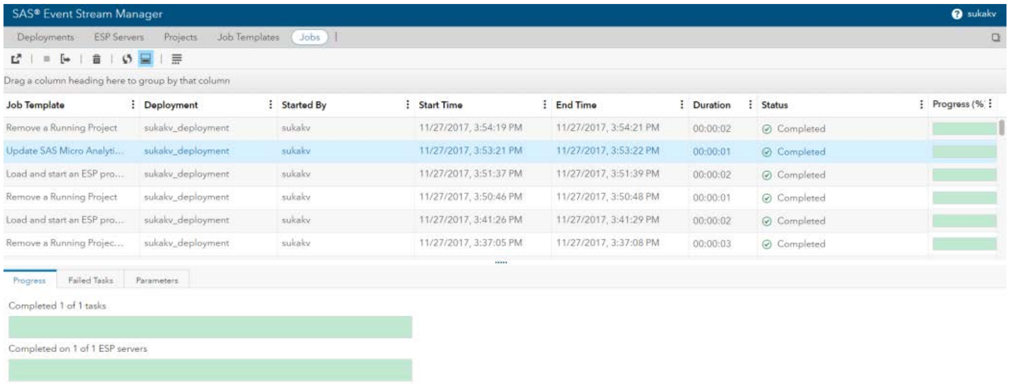

Figure 6. SAS® Event Stream Manager Tasks

300+ NATIVE ENDPOINTS PLUS API CONNECTIVITY

SAS Event Stream Processing natively integrates with more than 300 systems, applications, and endpoints such as the following:

- File and socket
- Web services
- WebSockets
- Network and Syslog sniffers
- Messaging protocols and applications including Java Messaging Service (JMS), Message Queueing Telemetry Transport (MQTT), Apache Kafka, IBM WebSphereMQ, and RabbitMQ
- Data stores like Apache Hadoop, Oracle, Greenplum, Teradata, and Apache Cassandra
- Open-source integration frameworks like Apache Camel and Apache NiFi

In addition, SAS Event Stream Processing provides documented and easy-to-use Python, Java, and C APIs for you to write your custom integration layer if needed.

BUSINESS BENEFITS

This topic focuses on translating SAS Event Stream Processing technical features into tangible business benefits so that we can create a business plan to justify the adoption of SAS Event Stream Processing.

So how do we go from the technical features discussed in previous topics (such as Event Processing, Data Quality, Pattern Detection, Text Analytics, In-Stream Analytics and Machine Learning with Model Supervision, In-Stream Geofencing, and Open-Source Integration) to tangible business benefits (such as Revenue Increase, Customer Satisfaction Increase, Churn Reduction, and Fraud Reduction), improved productivity, reduced operational errors, and others?

WORK WITH THE BUSINESS AREAS TO IDENTIFY PROBLEMS AND NEW OPPORTUNITIES

The first step is to partner with business areas to identify either existing business problems and their associated estimated losses or new opportunities for creating revenue through new products or services that can be offered and their associated estimated gains.

It is not always easy to have this type of information ready and available, and, even when it is available, different areas and levels of the company might not agree with the estimated losses or gains metrics and might struggle in defining the priorities for the problems and new opportunities. Let's worry about this later. One suggested approach follows:

- Work with the business areas to identify existing business problems and/or new opportunities for creating revenue.
- Once you have several candidate problems and opportunities, work closely with the business areas to try to estimate the losses and gains associated with each problem or opportunity. They should consider both direct and indirect gains and losses.

MAP EVENT STREAM PROCESSING TECHNOLOGY FEATURES TO BUSINESS BENEFITS

The next step is to map SAS Event Stream Processing features to resulting business benefits. Below is a list of common benefits that can be used. Note that these might not apply to your specific problems or opportunities but can be used as a starting point.

Direct Business Benefits

Technical features like Event Processing, Pattern Detection, Text Analytics, In-Stream Analytics and Machine Learning, and In-Stream Geofencing can map almost directly to business benefits such as Revenue Increase, Customer Satisfaction Increase, Churn Reduction, and Fraud Reduction. The idea here is that we can continuously monitor our business (including customers and transactions) and apply manipulation and analytics (ranging from basic filters and aggregations), which sometimes by itself can bring tremendous business value, to more sophisticated techniques like pattern detection, geofencing, and analytical models in real time.

The real-time part is very important because here is where we usually can add the most value, and it is usually forgotten. For example, when we say that just applying a simple filter (for example, a failed transaction) or detecting a simple pattern in data (for example, the customer contacted us three times in less than one hour) can bring value, it is because of the real-time factor, and that makes sense. If we look to failed transactions or at an insistent customer two days later, chances are that this opportunity is already lost because a lot of time has passed. But if we look at it and act on it right away – seconds after it happens – chances are we can use that problem or opportunity in our favor in some of these ways:

- Offering new products or services, increasing revenue.
- Offering discounts for keeping customers when they start to think about leaving us – reducing churn.
- Solving problems, indirectly increasing customer satisfaction and reducing churn and

consequently maintaining revenue.

- Detecting fraudulent activities and preventing it from happening – reducing fraud.

Fast Time to Market

The graphical or programming project development technical feature can map to this business benefit. The idea here is that SAS Event Stream Processing projects can be created using either an easy-to-use graphical interface or Python code. This can boost productivity because each developer can use what he or she is most comfortable with and, yet the result will be the same – that is, a project will be executed by SAS Event Stream Processing the same way, using the same constructs and logic that we previously discussed.

This can directly boost productivity and might also indirectly cause a positive impact on other business metrics such as revenue, because if we can produce more projects with fewer resources, chances are we will do these development cycles more often. This applies both for the first project development cycle as well as for any subsequent project changes cycles.

In addition, the 300+ endpoints plus API connectivity can also map to this benefit. SAS Event Stream Processing can connect natively with several sources. And even if a specific source is not supported out-of-the-box, there are easy-to-use provided APIs that can be used to build a custom connector or adapter. We configure or build that once and all the sources and events will be available for usage. It's as simple as that. So, chances are that after a few development cycles, we will be plugged in to every source of data that our business has, increasing time to market.

Reduction of Model and Project Deployment Operational Errors

The in-stream analytics technical feature can also map to this business benefit.

The idea here is that you can use trained analytical models without needing any recoding. Just drag a box, select the model, and you are good to go. This might sound trivial, but there are many scenarios where models are not plug-and-play, that is, they need to be recoded in several disparate systems that support our business. SAS Event Stream Processing eliminates this need, reducing the chances of operational errors occurring.

Another technical feature that maps to this benefit is graphical project deployment. The idea is that regardless of the method that you choose to create you project in SAS Event Stream Processing, you can centralize the deployment using a graphical interface. This helps to keep things organized and in one place, helping in avoiding IT operational errors.

Model Lift Improvement

The in-stream model supervision technical feature can map to this business benefit.

This can automate business decisions related to the modeling cycle process itself – like which model to run, when to retrain this model, when to change this model with another one – and doing this over time can cause a model lift improvement.

Unified Real-Time Data Integration and Data Quality

Technical features like 300+ endpoints plus API connectivity, event processing, and data quality can also map to this business benefit, when applicable. Keep in mind that the goal here is not to replace every ETL process that we have in our business. But, for some cases, it can make sense to use SAS Event Stream Processing as a single source of data integration.

If we think that before SAS Event Stream Processing adoption we are limited using periodic batch ETL processes to get our data, and after we adopt SAS Event Stream Processing we have real-time access to that data, we can think of retiring some of the ETL processes and make SAS Event Stream Processing responsible for writing this data to historical databases. We can apply transformations and data quality in real time to these events before writing them out to the database.

This can help create a unified vision of the data because we will have only one process transforming the data. This can increase the time to get the data and improve the quality of the data.

WORK WITH THE BUSINESS AREAS TO REFINE AND FINISH THE PLAN

The final step is to work again with the business areas to explain the tangible business benefits that SAS Event Stream Processing provides and to refine the plan in conjunction with them.

There will be times that you will be required to provide estimates for loss reductions or direct gains. If you need to do that, try to be conservative in the estimates and try to get validation from the business areas, themselves. They usually know the business and are qualified to create rationales like these.

SUGGEST THE ADOPTION ROADMAP

After you finish the business plan, you can improve it by suggesting the SAS Event Stream Processing adoption roadmap. The suggested approach is to choose one business problem or opportunity to start with.

Although it is possible to cover more than one business problem at a time, this can add complexity to the project, introduce uncertainties, which, together, might increase risk for the project. So, it is usually better to attack one problem or opportunity at a time.

With that in mind, considering that SAS Event Stream Processing was built for speed – being capable of handling millions of events per second with commodity hardware – one strategy that can be used to reduce the subsequent implementation cycles is to get all the data for each involved system or application. For example, consider that the first implementation cycle must get specific data from three systems. You can just get all the data and discard what is not needed for the first cycle, and the next cycles will be able to use that data with no changes needed in the integration – because you are already getting all the data. This results in a decreased time to market.

CONCLUSION

SAS Event Stream Processing has many valuable features, but most of them are technical. This paper explained these features and guided you through a process to translate them into business benefits and then use these translated benefits to create a business plan that can be used to help business areas adopt and leverage SAS Event Stream Processing.

We have talked about five tangible business benefits that are usually seen in SAS Event Stream Processing projects including direct business benefits - such as Revenue Increase, Customer Satisfaction Increase, Churn Reduction and Fraud Reduction, among others, along with fast time to market, reduction of model and project deployment operational errors, model lift improvement, and unified real-time data integration and data quality. In Table 4 we summarize the usual mappings between these technical features and the related business benefits.

Technical Features	Business Benefits
Event Processing, Pattern Detection, Text Analytics, In-Stream Analytics and Machine Learning, In-Stream Geofencing.	Direct business benefits such as Revenue Increase, Customer Satisfaction Increase, Churn Reduction, and Fraud Reduction.
Graphical or programming project development; 300+ endpoints plus API connectivity.	Fast Time to Market
In-stream analytics; graphical project deployment	Reduction of Model and Project Deployment Operational Errors
in-stream model supervision	Model Lift Improvement
300+ endpoints plus API connectivity; event processing; data quality	Unified Real-Time Data Integration and Data Quality

Table 4. Technical Features to Business Benefits mapping

Although they can all apply to a given problem or opportunity, it is usually best to choose the most beneficial ones for each problem or opportunity, as this usually results in a strong overall argument, because you are keeping things simple and going straight to the point.

Finally, the most important recommendation is to partner with the business areas so that they can help come up with the problems and opportunities that can be targeted, and they can estimate losses and gains as accurately as possible. You can tell them that you understand that they might not have all the information needed to calculate an exact number, and that is okay because the most important aspect here is the rationale used, and you can help with the technical versus business benefits mapping. Work closely with them and make them feel like an important stakeholder, because they are. This also leads to better results especially during the presentation to the higher-level stakeholders.

REFERENCES

SAS Institute, Inc. 2019. SAS® Event Stream Processing Customer Documentation. Cary, NC: SAS Institute Inc. Available https://support.sas.com/documentation/onlinedoc/esp/index.html. (accessed February 25, 2019).

SAS Institute, Inc. 2019. SAS® Event Stream Manager Customer Documentation. Cary, NC: SAS Institute Inc. Available https://support.sas.com/documentation/prod-p/esm/index.html. (accessed February 25, 2019).

SAS Institute, Inc. 2019. SAS® Quality Knowledge Base (QKB) Customer Documentation. Cary, NC: SAS Institute Inc. Available http://support.sas.com/documentation/onlinedoc/qkb/. (accessed February 25, 2019).

ACKNOWLEDGMENTS

I would like to acknowledge all the customers that I have had the chance to work with on SAS Event Stream Processing projects and opportunities. These experiences and interactions were fundamental for this paper.

I would also like to especially acknowledge and offer thanks to my technical reviewers and friends Anderson Sforcini, Cristiane Rodrigues and Mark Addy.

RECOMMENDED READING

SAS Institute, Inc. 2019. *Learn SAS® Event Stream Processing*. Available https://support.sas.com/en/software/event-stream-processing.html.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Henrique Consolini Danc
SAS Institute Brazil
+5511972349844
henrique.danc@sas.com
www.sas.com
www.linkedin.com/in/henriquedanc