Paper CM02

Connected Monitoring

Mike Henderson, SAS, Cary, North Carolina, USA

ABSTRACT

Moving from traditional clinical trial monitoring to central monitoring exposes a greater need for interconnectivity between data collection systems. The goal of an agile approach with smooth adaptions to areas of most significant risk is frequently hindered by a lag in a data feed or waiting on an upstream process to combine data from multiple systems for review. In this paper, we show an approach to connecting systems without disrupting them and then utilizing the interconnectivity for greater use of rule-based and statistical methods based monitoring with a quality-focused, risk directed approach for studies, sites, and subjects. There is a particular focus on treating data as streams with interconnectivity accompanied by managing change without disruption to established systems and procedures as well as motivating the need for continual reassessment of all trial data.

INTRODUCTION

Decreasing the latency to clinical trial data sources makes it possible to target monitoring activities better. Enhancements include adopting more analytical methods for detecting anomalies, increasing interactivity between data systems, and being more traceable and adaptable. Additionally, it is easier to set up and use repeatably. Exposure to tools for streaming data processing will enable this approach.

STREAMING VERSUS STATIC

Static data is data at rest. The most common way to reference data sources. When we point to the source being a CSV, excel file, database, or other file types, we are pointing at a resting place for data. A big part of clinical trial data is moving and merging multiple static data sources multiple times throughout trial operations.

Streaming data is data that has not reached it's resting place yet. Most use cases for this come from disciplines with high volume, high-frequency data that needs evaluation in near real-time. Solving problems at this scale has led to tools that automate and connect data of any type and any source. These tools are now mature enough to offer benefits to any application that needs to connect periodic data from multiple sources.

For some data sources, it may not yet be practical to stream from the actual data source. There are still advantages of using streaming methodologies for these sources. Rather than setting up periodic data transfers of static sources, we can set up streams from the first static source into our actual systems. The advantage here is that we can still have dynamic data systems where some sources just update less frequently.

DATA SOURCES

In clinical trial monitoring, we are handing a growing number of data sources. To name a few of our common ones:

- EMR
- Labs
- Devices
- eCOA
- ePRO
- RTSM
- eCRF
- Payment
- CTMS
- Safety
- RWD

Each of these can be from different systems, stored in different formats, updated at different frequencies, maybe even controlled by different vendors.

Our processes today are useful for managing these systems, and they are an integral part of the trial workflow and management. The first consideration for this paper is connecting these systems without interrupting them or disrupting established workflows. The desire is to augment the overall process while adding quality system components to minimize risk to patients and trial integrity.

DECREASE LATENCY

Each of the data sources mentioned above is usually managed by doing a data transfer at some specified frequency. This transfer may mean updating a table somewhere by doing a pull from the database a system writes data in to. Often it means a vendor exports data, transforms it into a portable format, saves it to a location for pickup (maybe even emails it). This file is then processed further by accessing it, further preparing it, combining it with information from other systems, and writing it to a location to be consumed by monitoring tasks.

Our goal is employing the methodologies described later in the paper by connecting to data as close to its point of creation as possible. We wish to do this without introducing security risks or changing the operation of systems. In some cases, we may be able to connect directly to the system generating data and even interact with it dynamically. In other cases, the data may still need to be periodically exported and dropped off at a common location. Either extreme is a move in the right direction and enables us to achieve connected monitoring. As we adapt to this type of system, we see an increasing desire to move the connection closer to the point of data creation, and it is easier to show the value for this as we grow in our implementations of connected monitoring. The key is to start where we can and progress.

MONITORING

The goals of monitoring data may be straightforward. The purpose of this paper is to illustrate the setup of a system that can manage all levels of complexity. Some types of data-based actions that may take place within central monitoring are:

- If a new or changed record is created, update a table
- Apply and filed level check – if a value is out of range, create an alert
- Apply cross-field logic – if a condition is met, or not met, create an alert
- If a record is flagged, create an alert in another system
- If a record is flagged check for similar occurrences in other records
- As new records are created run them through a bank of rules to calculate data integrity and patient safety thresholds
- Do in-flight analytics to trigger alerts for anomalies
- Apply model scores for pre-trained analytical models that can then trigger alerts or reroute data
- Complex combinations of more than one of the items in the list above

MANAGING QUALITY

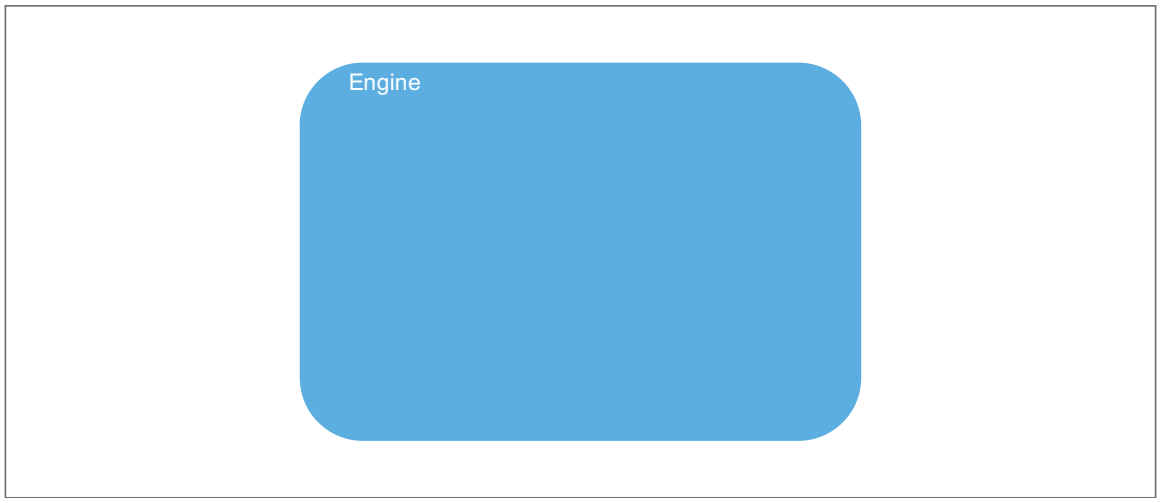
Some monitoring actions may be marked as high-risk indicators for either patient safety or trial integrity. Other actions may be directed to faster trial workflow while creating logging and management and preventing waiting time and human error. In either case, it is important that the system for connected monitoring be able to connect data with minimal latency and make pre-prescribed actions simple to specify, deploy, manage, update, and trace.

EVENT STREAM PROCESSING

To achieve our connected monitoring goals, we process data from various sources. The application of logic to data from one or more sources is called a query. Query is a general term that may be as simple as rerouting records from one system to another or more complex, like applying a signal detection algorithm as data moves. In the following sections, terms are introduced for doing these queries under an event stream processing application.

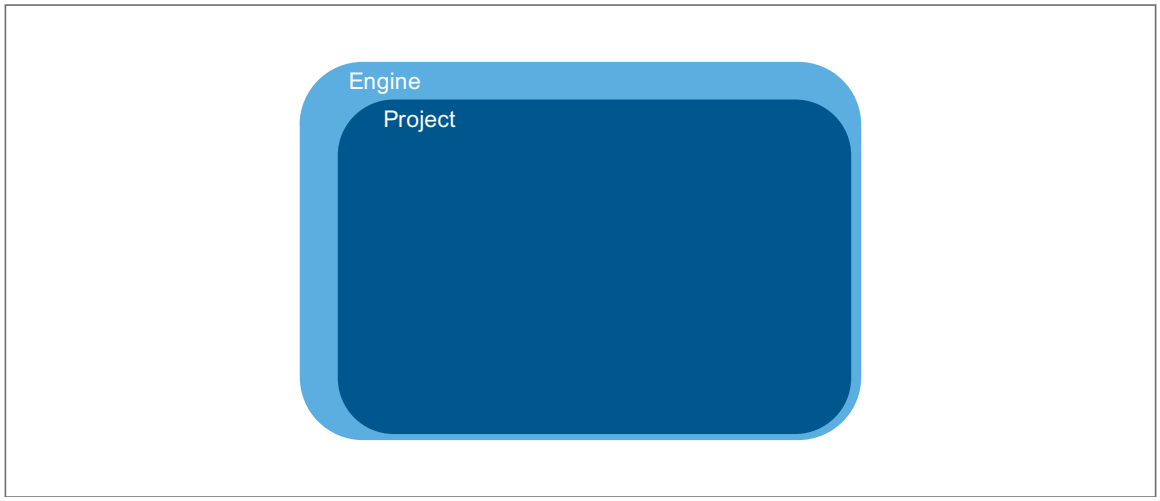
ENGINE

At the top of the hierarchy is an engine. This is the execution environment where our monitoring queries run.



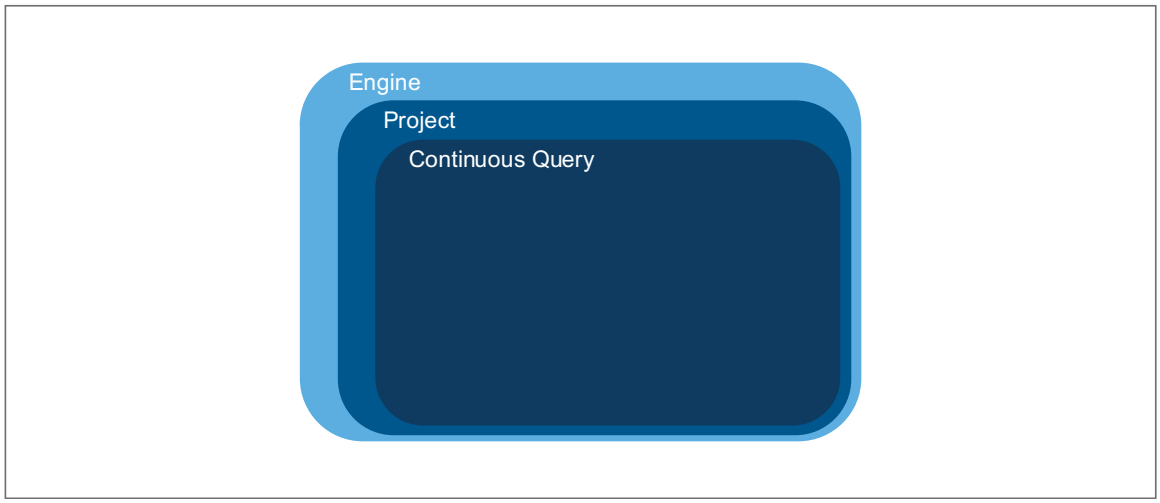
PROJECT

Each of our monitoring queries are contained within a project. An engine can contain one, or more likely, many projects for clinical trial monitoring.



QUERY

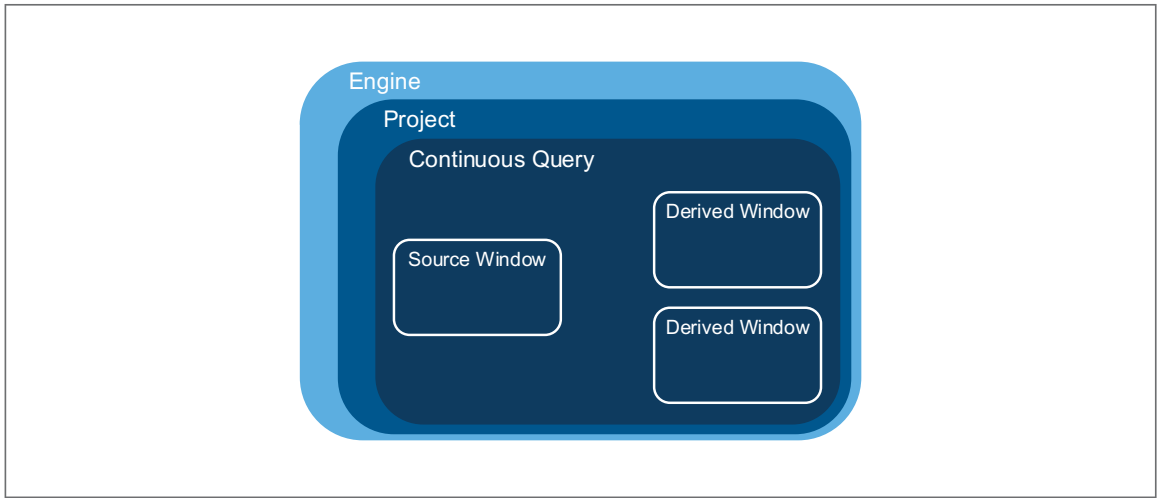
Each monitoring task, however simple or complex, is broken down into individual queries. A single project can contain one or more queries. We can also connect different projects, which allows us to organize the logic of a connected monitoring system in a logical and controlled manner.



WINDOWS

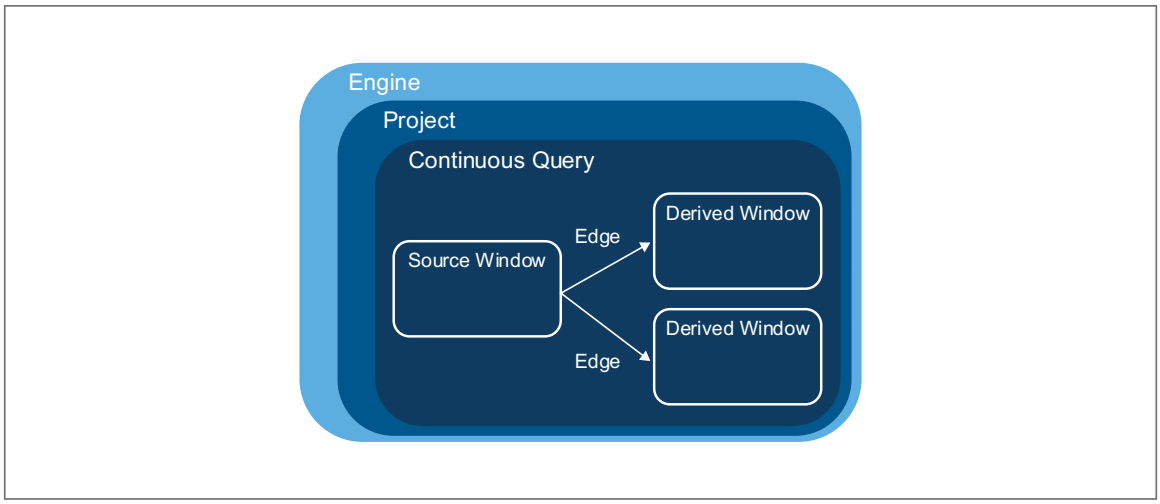
To execute a query, we need to see data from one or more sources. These are called source windows.

The logic for executing our query and the result of our query is encapsulated in another set of windows. These are called derived windows.



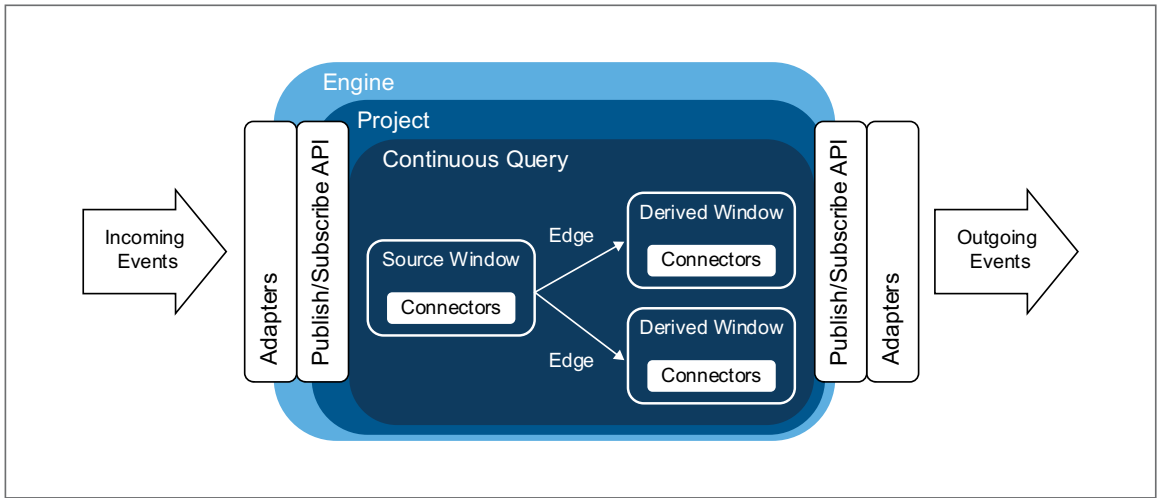
EDGES

The workflow of our query can involve connecting one or more source windows to one or more derived windows. Our logic may require connecting or sequencing multiple derived windows. This workflow gets specified with edges.



FLOWS

To stream data in and out of our projects, we can use adaptors to a wide variety of data sources to both read from (subscribe) and pass on (publish) data. This flow is depicted below as adapters and connectors for the flow of data through our project.



EXAMPLES

Using these ideas, we can accomplish significant new benefits and automate many everyday tasks. A few areas I can see this making central monitoring and overall trial conduct more efficient and better are:

- Centrally managed edit checks for CRF fields
- Dynamic edit checks. Rules across many fields, even deploy machine learning derived algorithms trained across previous trials, maybe even re-trained on previous patients in the current trial
- Adaptive edit checks. Use real-world-data to build cohorts for trials and match patients to strata within the cohort to derive anticipated ranges for CRF that adapt as more data for a patient accumulates.
- Actively intervene in high-risk situations by connecting people and systems to data sources in near real-time
- Give feedback to clinical staff at the patient, or visit level to reinforce protocols
- Automate trial adaptions

The following are some common tasks in clinical trial data flow translated into the grammar presented for event stream processing

1. eCRF data transfer processed into SDTM
    - Project: process RAW to SDTM
    - Subscribe to ODM export for EDC system (example)
    - Continuous Query: XML processing
    - Source Windows:
        1. Connect to XML file
        2. Metadata Repository
    - Derived Windows:
        1. Generalized Structure
        2. DM
        3. CO
        4. CM
        5. EX
        6. AE
        7. DS
        8. MH
        9. …
    - Publish to: SAS Datasets, CSV, RDBMS
2. Process a query
    - Project: query alerting
    - Continuous Query: XML processing
    - Source Windows:
        1. Connect to XML file for eCRF
        2. Connect to XML file for query data
    - Derived Windows
        1. Query Status

a) Query Open
    a. Risk Assessment Rules engine
        - Publish to: CTMS system

3. Query Research
    - Connect 1 & 2 together
    - Project: query alerting (2)
    - Source Windows:
        1. Connect to Derived Windows in Project: process RAW to SDTM (1)
        2. Connect to XML file for query data
    - Derived windows
        1. Query Status
            a) Query Open
            b) Derived Window
                a. Collect AE, DM, CO CM, … from Derived Windows in (1)
                    - Publish to: CTMS system

MORE ON WINDOWS

Windows can be any type of data related task. It could be as simple as running a SQL query. More interestingly it could be running an on the fly statistical monitoring task:

- Processing data through a bank of rules to detect errors or risk
- Scoring data with a previously trained model
- Re-training a model with new data
- Apply clustering methods like DBSCAN, K-means
- Apply anomaly detection such as Kalman filter, isolation forest
- Rerouting data to other systems for alerting monitoring processes
- Offline training of models such as neural networks, Robust PCA
- Process text entries into topics and keywords to process for risk association

CONNECTED MONITORING

Once adopted, connected monitoring can enable continuous monitoring. Even allowing near-instant insights in cases where a record is identified for a risk indicator, it could then automatically alert on any other records with similar scenarios. If connections can be achieved into source systems, then the edit checks in eCRF's could be linked to data from other vendors and even have self-adapting rules based on information learned about individual patients. While the primary purpose of connected monitoring should be to reduce risk to the patient and ensure trial integrity, it also has clear benefits to the trial workflow. It could easily lead to shorter wait times and less human interventions in the data systems for clinical trials.

CONCLUSION

Connectivity is a big step forward as we move toward greater use of central monitoring and more interactivity between data sources to achieve the goals of central monitoring. Throughout this paper, descriptions have introduced a new way of thinking about data connectivity while also positioning how to implement this without disrupting well-established processes. Thinking of data sources like data streams and connectivity with the ideas of subscribing and publishing is a step forward toward the ultimate goals of central monitoring. Injecting our monitoring rules for today and advanced approaches as we mature is much easier as we allow the streams to be processed through the rules and analytics engines of choice as this space matures in the years to come.

CONTACT INFORMATION

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

Mike Henderson
SAS
Mike.henderson@sas.com
www.statmike.com (visit for: paper, presentation, discussion)
www.linkedin.com/in/statmike