PharmaSUG China 2018 – Paper DM-47

Decision Making Confidence - The Way to Make Decision by BPMN in SAS® LSAF

Emma Liu, SAS R&D, Beijing, China

ABSTRACT

As we known SAS delivers Life Science Analytics Framework (LSAF) solution which implements Business Process Model and Notation (BPMN) to help pharmaceutical companies to joint all the activities of your data management, metadata management and clinical analytics around the clinical trial. On top of that, to make the wise decision on the mix variety of complex processes is very important for the drug development process. This paper is based on BPMN to demonstrate how to automate and streamline your clinical research process, target on design Form Properties with your business data to make the decision by sequence flow on your clinical study result to speed the development of new therapies through optimized operational activities.

INTRODUCTION

The Pharmaceutical industry has been constantly evolving to globalization, consolidation and regulatory compliances. Clinical professionals are shifting to more systematic, data-driven strategies to improve quality outcomes by eliminating compliance issues in to processes. So that design the good modelling of clinical processes and workflows is more and more important in pharmaceutical industry. SAS® Life Science Analytics Framework (LSAF) with implementing of BPMN v2.0 enables your organization to manage the workflow and electronic forms processing dynamically and automatically. This paper demonstrates a use case which is to perform clinical trial design in the study Phase I by BPMN with business process flow. And by implementing Form Property of Process Data and Completion Data in the XML file of BPMN, by design different expression in conditions of Sequence Flow, we will see how to make a process decision via processed oriented models in SAS® LSAF.

USE CASE INRODUCTION – TRIAL DESIGN BY BPMN

Trial Design by BPMN

In the Study Phase 1, design of Platelet response in healthy human subjects which was conducted to assess the safety, Kinetics, and pharmacodynamics of Eltrombopag (nick name), when administered as a once daily oral dose.

We design the Use Case by process flow as Figure1 below by Camunda Modeler which supports BPMN v2.0(See "Appendix" section for BPMN and Camunda Modeler introduction). It will be stored as XML file which finally can be deploy to SAS® LSAF to setup with your business data and execute the process flow.

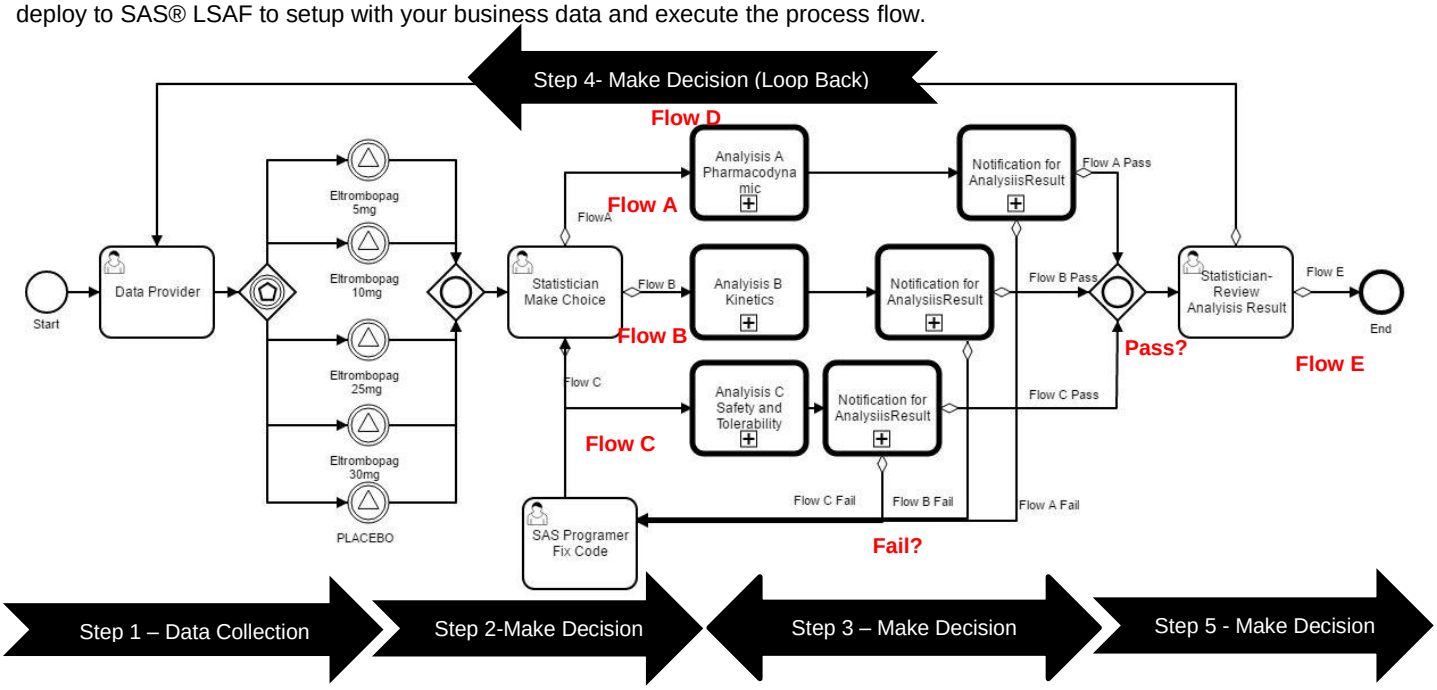


Figure 1. Use Case design by Camunda Modeler

FIVE STEPS - USE CASE TRANSLATE TO BPMN

In summary, the Use Case can be translated to below five main steps to make you understand the whole process.

Step1: **Data Collection**	Data Provider receive the assignment task which is to perform daily data collection of healthy male subjects oral dosing with Eltrombopag (once per day) for 30 days with dose 5mg, 10mg, 25mg, 30mg, and PLACEBO separately.
Step2: **Design Target**	Statistician get the patients data which uploaded by data provider and make decision whether to go Flow A, Flow B, Flow C or all together which is to analyze the trial targets of pharmacodynamics, Kinetics, Safety and Tolerability of Eltrombopag.
Step3: **Fix Code Error**	The Flow A, Flow B and Flow C will perform data analysis separately to generated report of Analysis A, Analysis B, and Analysis C by executing SAS code and send email notification of the SAS code status and analysis result. The process flow will make decision automatically base on the SAS code execution result: • If SAS Code run Pass – It will go to next step which is to continue the process • If SAS Code Fail – SAS Programmer have chance to fix the code error, which is loop back to step2 to fix the SAS Code error and re-run code.
Step4: **Endpoint Day**	Statistician will be received a task assignment asked him to review the analysis result and make the decision of the trial design. Since we need to get patient data daily with the response of different doses of Eltrombopag, so the process should be looped to the beginning when new day comes. We design the Endpoint Day to 30, when Endpoint Day is less than or equal to 30 days, it will make decision to go Flow D which is go to the beginning and restart the whole process
Step5: **Research Finish**	When the Endpoint Day greater than 30 days, we will make decision to finish the process to end our clinical study research. So that it will go Flow E and finish the whole process.

DECISION MAKING IN USE CASE

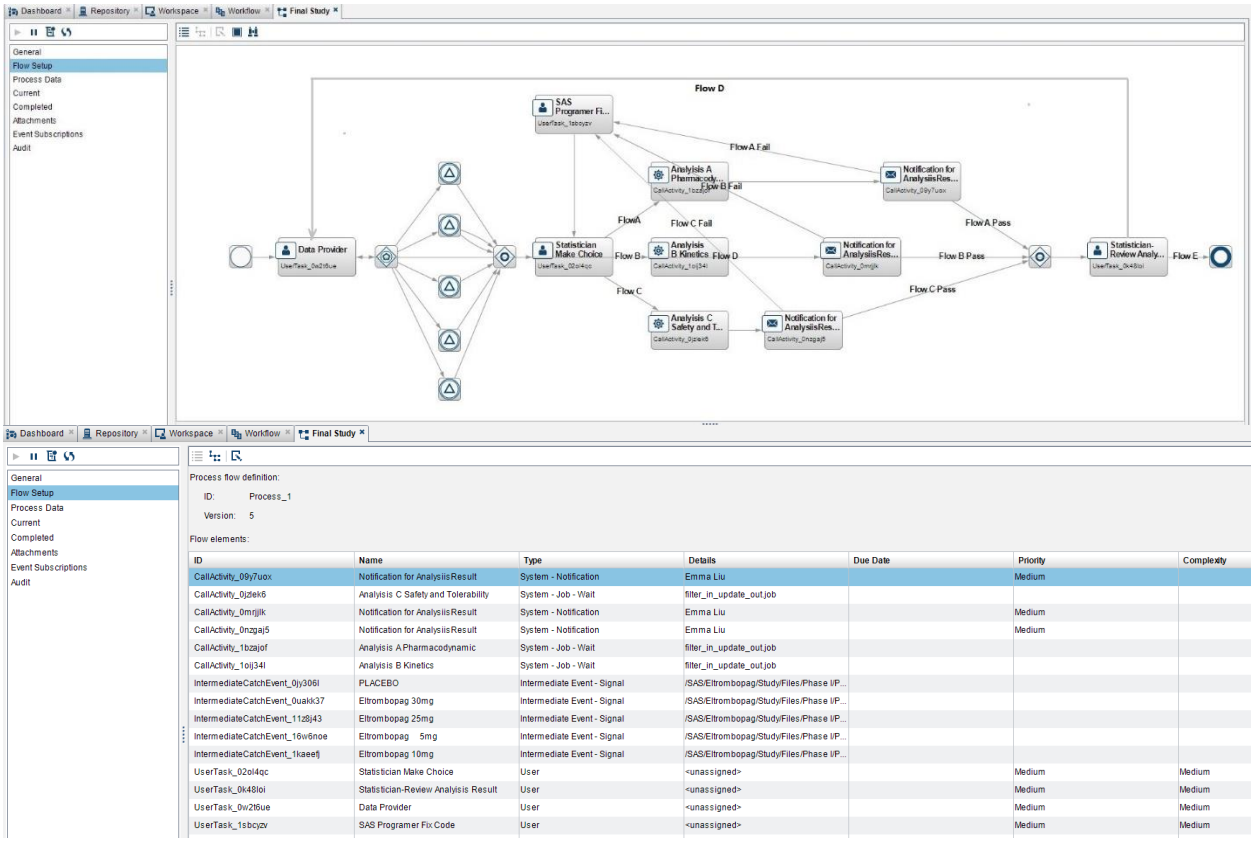
- At Step 2, we see statistician is free to make the decision to choose which trial design for analysis in the target of pharmacodynamics, Kinetics, Safety and Tolerability, this is control by Completion data.
- At Step 3, we see SAS Code execution result can make the decision to go which follow automatically, this is controlled by System Job Status.
- At Step 4 and Step 5, we see the whole process can be looped daily with process being re-run and continue to generate analysis report daily when new day comes of data updates, this is being controlled by Endpoint Day in the Process Data to make the decision.

In the conclusion, how to make the decision and control the process to go which flow is verify important and directly control your whole business process. Let's see next section of how to implement this Use Case to SAS® LSAF and how to make the decision in your process flow by Completion Data, System Job Status and Process Data in SAS® LSAF.

HOW DECISION MAKING WORKS IN USE CASE OF SAS® LSAF

SAS® LSAF is one of SAS's core solutions in pharmaceutical industry which is to perform data processing, data analysis related works in the clinical warehouse which implementing SAS Base and BPMN standard as Process Flow. In SAS® LSAF, user can see the BPMN elements in the diagram view and list view to get the whole outline of your process flow (Display 1). User can setup the Process Flow with your business data and SAS Job for executing SAS code. Each element will be converted to a Task in SAS® LSAF and take responsibility in different parts. Below are the common Tasks in the Process Flow of SAS® LSAF, also our Use Case are using them as example.

- User Task - Is for work assignment to different team member in your organization.
- Signal Task -Is for setup the location in the Repository of SAS LSAF® to store you research data.
- System Job Task- Is for design SAS code and SAS code execution.
- Email Notification Task - Is for send email to any person which want to be noticed.



ID	Name	Type	Details	Due Date	Priority	Complexity
CallActivity_09y7uox	Notification for AnalysiisResult	System - Notification	Emma Liu		Medium	
CallActivity_0jzlek6	Analysis C Safety and Tolerability	System - Job - Wait	filter_in_update_out.job			
CallActivity_0mrjjlk	Notification for AnalysiisResult	System - Notification	Emma Liu		Medium	
CallActivity_0nzgaj5	Notification for AnalysiisResult	System - Notification	Emma Liu		Medium	
CallActivity_1bzajof	Analysis A Pharmacodynamic	System - Job - Wait	filter_in_update_out.job			
CallActivity_1oij34l	Analysis B Kinetics	System - Job - Wait	filter_in_update_out.job			
IntermediateCatchEvent_0jy306l	PLACEBO	Intermediate Event - Signal	/SAS/Eltrombopag/Study/Files/Phase I/P...			
IntermediateCatchEvent_0uakk37	Eltrombopag 30mg	Intermediate Event - Signal	/SAS/Eltrombopag/Study/Files/Phase I/P...			
IntermediateCatchEvent_112ibj43	Eltrombopag 25mg	Intermediate Event - Signal	/SAS/Eltrombopag/Study/Files/Phase I/P...			
IntermediateCatchEvent_16w6noe	Eltrombopag 5mg	Intermediate Event - Signal	/SAS/Eltrombopag/Study/Files/Phase I/P...			
IntermediateCatchEvent_1kaeetj	Eltrombopag 10mg	Intermediate Event - Signal	/SAS/Eltrombopag/Study/Files/Phase I/P...			
UserTask_02oI4qc	Statistician Make Choice	User	<unassigned>		Medium	Medium
UserTask_0k48loi	Statistician-Review Analysis Result	User	<unassigned>		Medium	Medium
UserTask_0w2t6ue	Data Provider	User	<unassigned>		Medium	Medium
UserTask_1sbcyzv	SAS Programer Fix Code	User	<unassigned>		Medium	Medium

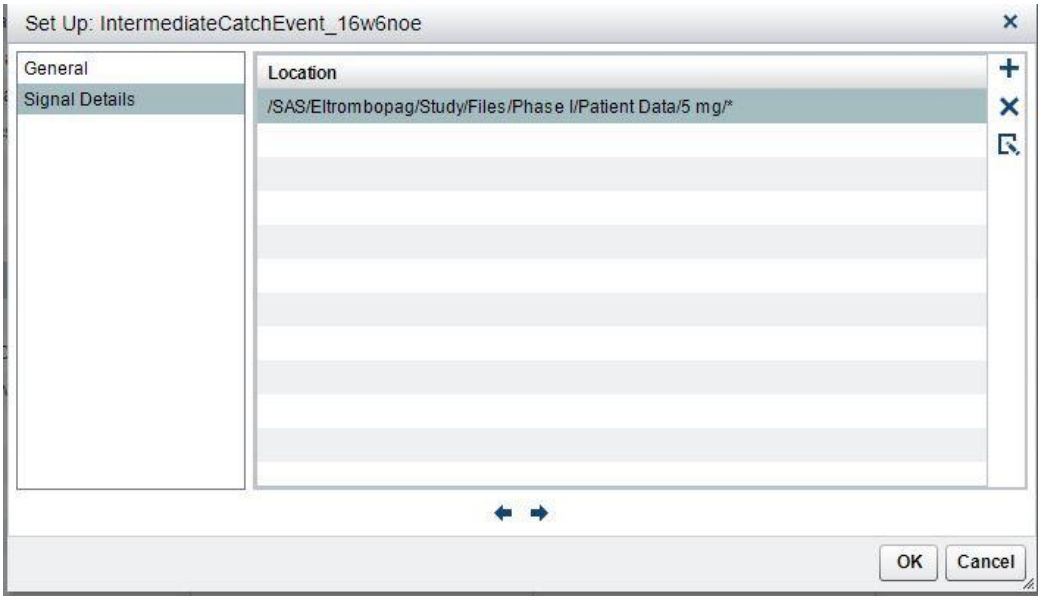
Display 1. Diagram View VS. List View (Tasks in LSAF)

FIVE STEPS - USE CASE IMPLEMENT TO SAS® LSAF

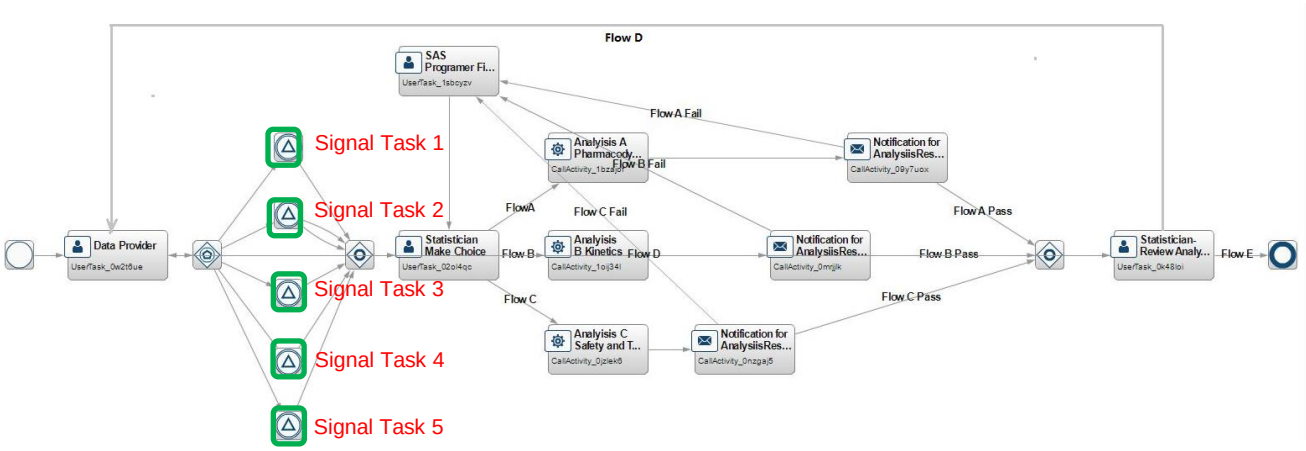
As above Tasks can be designed in the same process flow in this Use Case which is to perform data processing and data analyzing of in the collaborative, automatically and streamline Process Flow. Let's see more details of how the five steps - Use Case works in SAS® LSAF to make decision in the process.

Step1 – Data Collection

Step1: Data Collection	Data Provider receive the assignment task which is to perform daily data collection of healthy male subjects oral dosing with Eltrombopag (once per day) for 30 days with dose 5mg, 10mg, 25mg, 30mg, and PLACEBO separately.
SAS® LSAF Action	Signal Task - Being setup with the specific Location in SAS® LSAF Repository (Display2), when the files are being upload to the Location, the Signal Task will be triggered out. That means when male subjects (patient data) of dose 5mg, 10mg, 25mg, 30mg, and PLACEBO have updates, Data Provider will upload the patient data to the Location, so that below 5 signal Tasks will be fired up. (Display 3) • Signal Task 1 - Dose 5mg • Signal Task 2 – Dose 10mg • Signal Task 3 – Dose 25mg • Signal Task 4 - Dose 30mg • Signal Task 5 - Dose PLACEBO



Display2. Signal Task and Location Setup

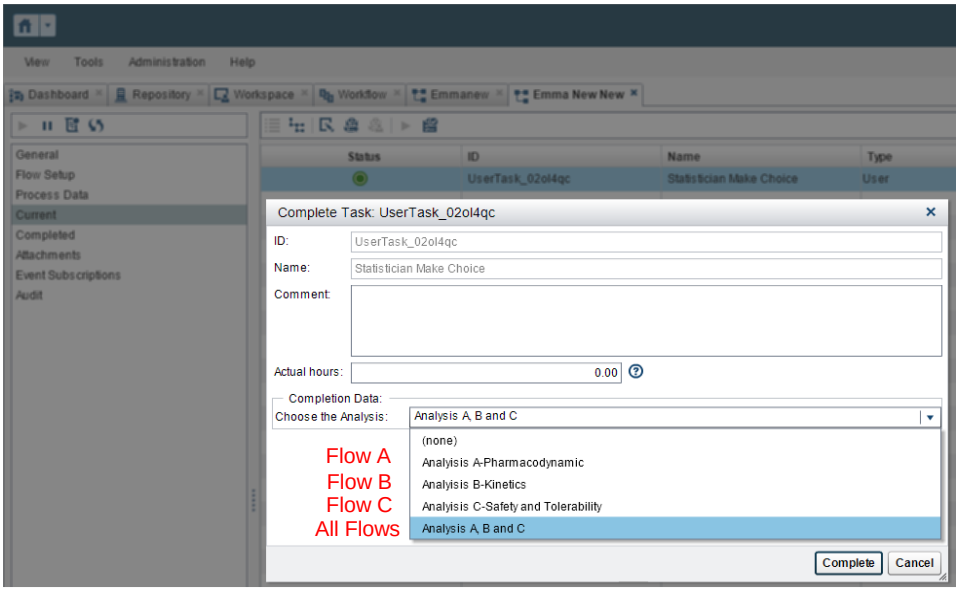


Display 3. Diagram View & Signal Task in SAS® LSAF

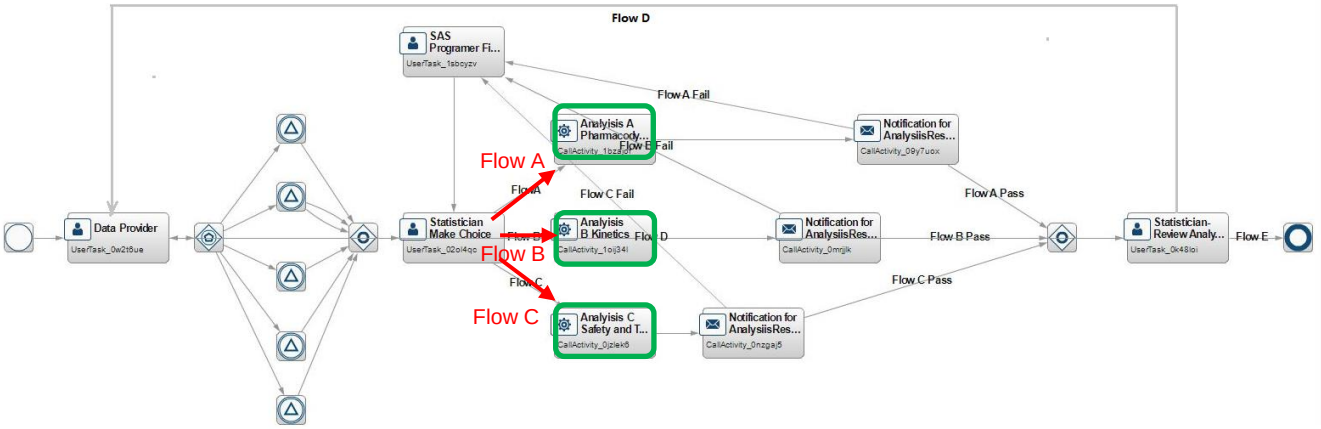
Step2 – Design Target (Completion Data)

Step2: Design Target	Statistician get the patients data which uploaded by data provider and make decision whether to go Flow A, Flow B, Flow C or all together which is to analyze the trial targets of pharmacodynamics, Kinetics, Safety and Tolerability of Eltrombopag.
SAS® LSAF Action	**User Task** - Statistician get assignment of User Task, start it and complete it with sets of Completion Data whether to choose which analysis report depend on your targets to make decision (Display 4).
	System Job Task – After make decision, it with executing different SAS Code in the server side to generate different reports which is to go Flow A, Flow B, Flow C or all together. (Display 5)
	Completion Data - The fields that appear in SAS® LSAF User Task (Display 4) and are only modifiable outside SAS® LSAF. These are being configured as global variables being setup in XML as "form properties". The global variables for setting on conditions in “sequence flows” can control the

	process. For how to set up “form properties" on conditions in “sequence flows”, we will talk about in the next section.



Display 4. Set Completion Data in User Task to make the decision

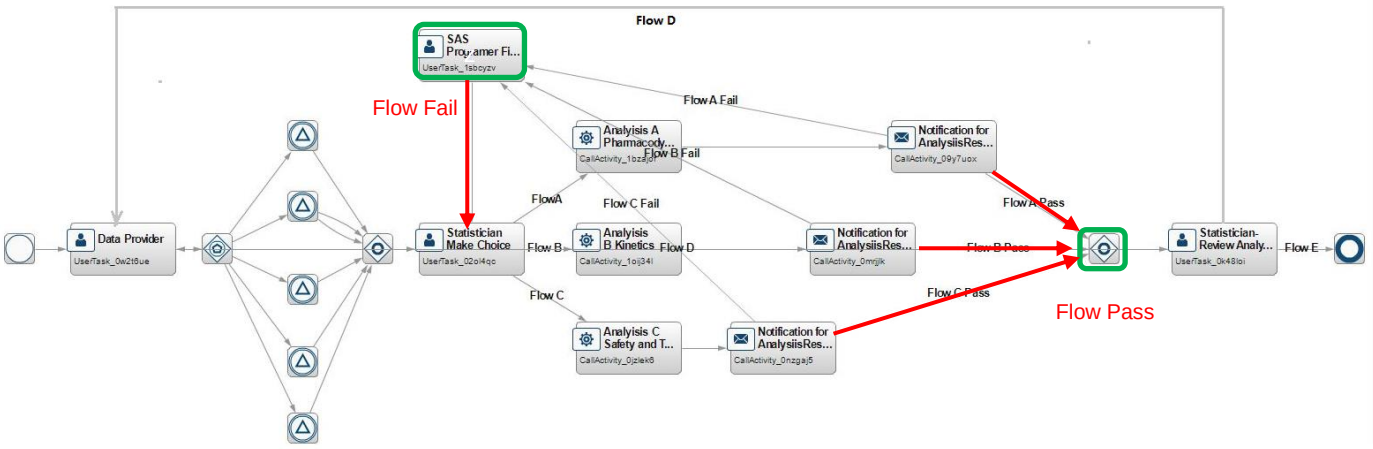


Display5. Make decision and run System Job task

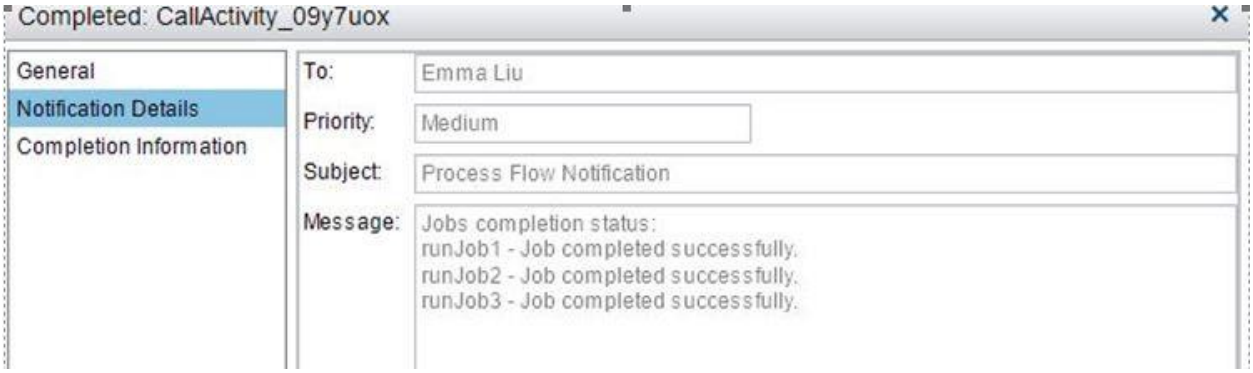
Step3 – Fix Code Error (Automated System Job Status)

Step3: **Fix Code Error**	The Flow A, Flow B, Flow C will perform data analysis separately to generated report of Analysis A, Analysis B, and Analysis C by executing SAS code and send email notification of the SAS code status and analysis result. The process flow will make decision automatically base on the SAS code execution result: • If SAS Code run Pass – It will go Step4 which is to continue the process • If SAS Code Fail – SAS Programmer have chance to fix the code error, which is loop back to step2 to fix the SAS Code error and re-run code.
SAS® LSAF Action	**System Job Status** – When System Job Task completes, it will return the Job run status with the value from -1 to 9. These values are being defined in the SAS Service which represents for the different job run status. For example, 0 represents job run successfully; 2 represents SAS Error. (In the next Section “Data Example in This Use Case” list each job run status.) • If SAS Code run Pass – It will return value 0, and the flow automatically go Flow Pass which is to continue the process. (Display 6)

	• If SAS Code Fail with Errors – It will return value 2, SAS Programmer have chance to fix the code error, which go to Flow Fail and loop back to step2 to fix the SAS Code error and re-run code. (Display 6) These are also being configured as global variables for setting on conditions in "sequence flows" can control the process. For how to set up conditions in "sequence flows", we will talk about in the next section.
	Notification Task – You can send email or message to any one in your organization who you want to notice in the process flow. The Job run status and the analysis result can be added to the message body. (Display 7)



Display 6. Notification Task & User Task

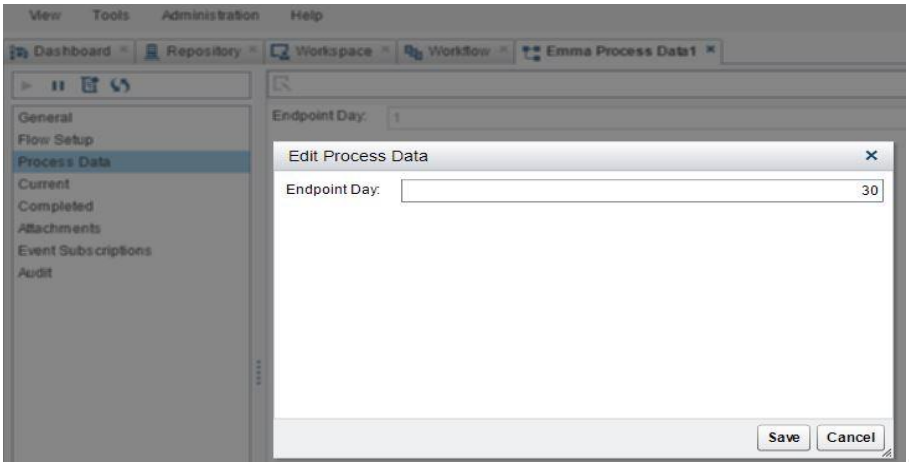


Display 7. Notification Task Details

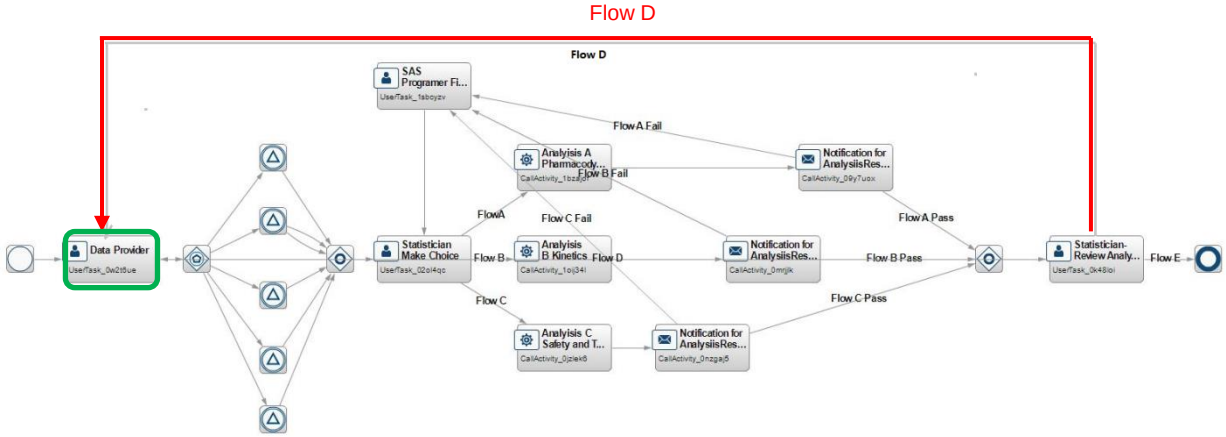
Step4 – Endpoint Day (Process Data)

Step4: Endpoint Day	Statistician will be received a task assignment asked him to review the analysis result and make the decision of the trial design. Since we need to get patient data daily with the response of different doses of Eltrombopag, so the process should be looped to the beginning when new day comes. We design the Endpoint Day to 30, when Endpoint Day is less than or equal to 30 days, it will make decision to go Flow D which is go to the beginning and restart the whole process

SAS® LSAF Action	The Endpoint Day can be setup in LSAF by Process Data to make decision to control process. We defined it with the expression- if it less than or equal to 30, it will go to Flow D which is back go beginning to loop the process (Display 9). So that the process flow is re-started from beginning when day increasing from 1 to 30.
	Process data - The fields that appear on the Process Flow Setup in left side bar (Display 8). Same as Completion Data, these are being configured as global variables being setup in XML as "form properties". The global variables for setting on conditions in "sequence flows" can control the process. For how to set up "form properties" on conditions in "sequence flows", we will talk about in the next section.



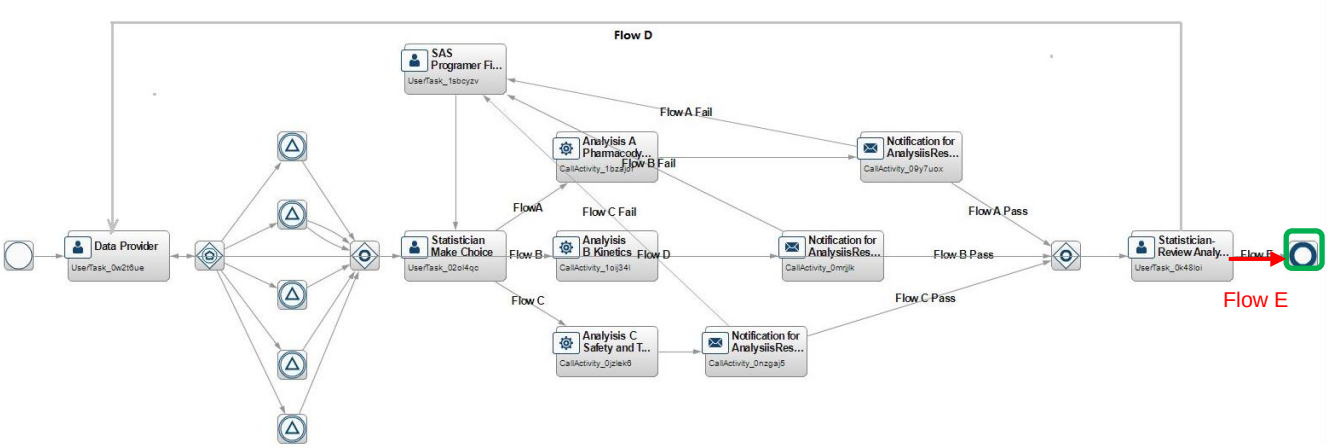
Display 8. Setup Process Data in LSAF- Endpoint Day less than or equal to 30



Display 9. Endpoint day less than 30 will go to the beginning (When new days increasing)

Step5 – Research Finish (End)

Step5: Research Finish	When the Endpoint Day greater than 30 days, we will make decision to finish the process to end our clinical study research. So that it will go Flow E and finish the whole process.
SAS® LSAF Action	When setup the Process Data of Endpoint Day greater than 30, it will go to Flow E which is finish the whole process of your clinical research (See Display 10)



Display 10. Endpoint day greater than 30, go to the end finish this research.

SUMMARY1: RESULTS ILLUSTRATION FOR THE ANALYSIS RESULT

At step 2, each Completion Data is to choose and run the different System Job Task which is to perform SAS code execution in the server side of SAS® LSAF. Finally, we got below different Analysis results represent this clinical study research result to represent Pharmacodynamic, Kinetics, Safety and tolerability in the clinical study Phase I.

- **Analysis A** - Pharmacodynamic, increases are apparent at 30, 50 and 75 mg. Values in graph indicate mean and 1 SD. (Figure2)
- **Analysis B** - Kinetics of platelet response in healthy male subjects following 30 days oral dosing Eltrombopage. The platelet number began rising at 5 days and peaked at day 15. Values in graph indicate mean and 1 SD. (Figure 2)
- **Analysis C** - Safety and tolerability, most frequently reported adverse events in the phase 1 study of Eltrombopag in healthy male subjects. (Figure 3)

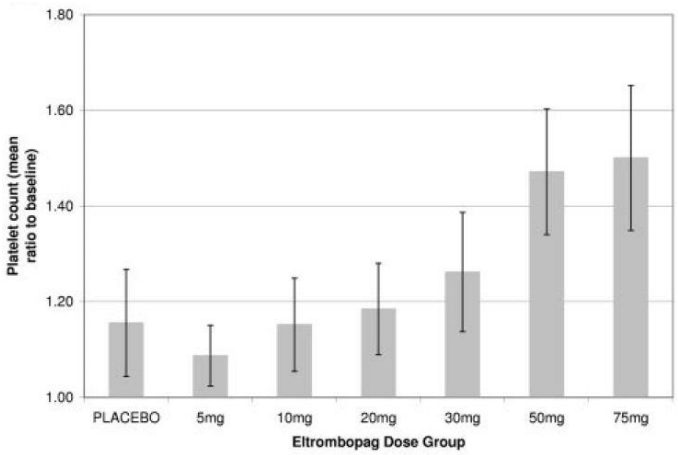


Figure 2. Analysis A

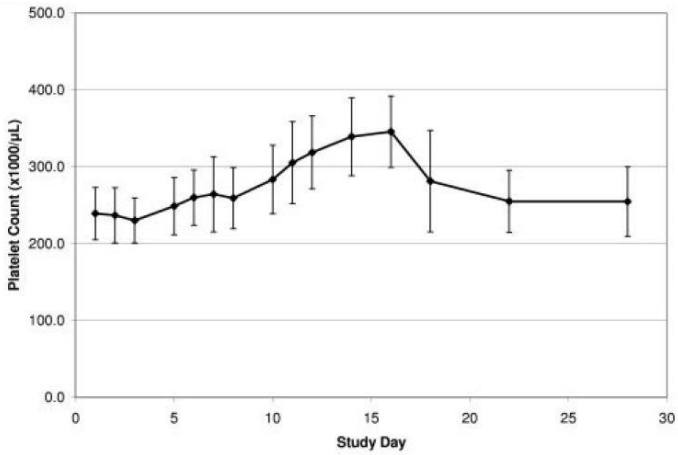


Figure 3. Analysis B

Adverse event	5-mg dose	10-mg dose	20-mg dose	30-mg dose	50-mg dose	75-mg dose	Placebo
Headache	2	4	0	1	1	3	2
Abdominal pain/cramp	0	1	1	0	0	1	0
Sore throat	1	1	0	0	0	1	0
Tiredness	0	0	0	3	0	0	0
Loose stools	0	0	0	0	0	0	2
Minor nose bleed	0	1	0	0	0	1	0
Bruise	0	1	0	0	0	0	0
Subjects exposed	9	9	9	9	10	9	18

Figure4. Analysis C

SUMMARY2: DATA TO CONTROL THE PROCESS

- Completion Data - We see to choose which analysis report is control by Completion data.
- System Job Status – We see the process can be controlled automatically by System Job Status.
- Process Data - We see the whole process can be looped daily is controlled by Process Data.

These data can make the decision and control the process to go which sequence flow. So how to make the decision in your process flow is very important for our trial design. Let's see how to configure your Completion Data, System Job Status and Process Data.

DECISION MAKING SETUP

COMPLETION DATA AND PROCESS DATA WITH FORM PROPERTIES

As we talked in above sections, both Completion Data and Process Data fields are created by defining "Form Properties" in the definition. The Completion Data properties are set on a User Task, the Process Data properties are set on the Start event. Below Activiti Form Properties can be added to the XML in the Camunda Modeler and finally implement to SAS® LSAF with Completion Data and Process Data.

String	<activiti:formProperty id="stringID" name="String" type="string" default="My value"/>
Numeric (Long)	<activiti:formProperty id="longID" name="Long" type="long" default="123"/>
Date	<activiti:formProperty datePattern="yyyy-MM-dd'T'HH:mm:ssZ" id="dateID" name="Date" type="date" default="1975-03-21T12:00:00EDT"/>
Boolean	<activiti:formProperty id="booleanID" name="Boolean" type="boolean" default="true"/>
Enum	<activiti:formProperty id="EnumID" name="Pick One" type="enum" default="A"> <activiti:value id="A" name="Choice A"/> <activiti:value id="B" name="Choice B"/> <activiti:value id="C" name="Choice C"/> </activiti:formProperty>

FORM PROPERTY INTEGRATES TO XML

Camunda Modeler does not support the implementation we are using via its UI, so we can only set and modify them directly in the XML. Below 4 steps demonstrate how to add Activiti Prom Property to the XML structure.

Step1 – Add Package to Top

Add below package to the definitions line at top after xmlns:di=*"http://www.omg.org/spec/DD/20100524/DI"* in XML.

```
xmlns:activiti="http://activiti.org/bpmn"

xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:omgdc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:omgdi="http://www.omg.org/spec/DD/20100524/DI"
```

Step2 – Add Form Property for Completion Data

Add below extensionElements which highlighted with yellow to the User Task in the XML.

```
<bpmn2:userTask id="UserTask_1" name="COMPLETED SUCCESSFUL">
  <bpmn2:extensionElements>
  <Add any types of properties between these two tags>
  </bpmn2:extensionElements>
    <bpmn2:incoming>SequenceFlow_2</bpmn2:incoming>
<bpmn2:outgoing>SequenceFlow_10</bpmn2:outgoing>
</bpmn2:userTask>
```

Step3 – Add Form Property for Process Data

Add below extensionElements which highlighted with yellow to the Start Event in the XML.

```
<bpmn2:startEvent id="StartEvent_1" name="Start">
  <bpmn2:extensionElements>
      <Add any types of properties between these two tags>
  </bpmn2:extensionElements>
  <bpmn2:outgoing>SequenceFlow_1</bpmn2:outgoing>
</bpmn2:startEvent>
```

CONDITION EXPRESSION ON SEQUENCE FLOW

The Completion data and Process Data being setup as Form Property in the XML which are available for setting on conditions by expressions in Sequence Flows to control the process. Let's see what is Sequence Flow and Condition.

Sequence Flow	Sequence Flow shows the order of Tasks to be performed. It is shown as a straight line with an arrow. It might show a default flow, or a conditional flow. Conditional flow can have a condition expression that are evaluated at runtime to determine whether or not the Sequence Flow will be used. In our Use Case, Sequence Flow is to determine whether to go Flow A, Flow B, Flow C, Flow D or Flow E.
Conditions	Conditions are expressions set on a sequence flow to determine if the target of the sequence flow will get activated when the source is completed. The expression could be anything but most commonly we expect it to evaluate a global Process Variable (Form Property). The Process Variable is created from a form property id.

We have defined the Completion Data and Process Data in the XML in step1-3, now we can set up the Expression as the condition in the Sequence Flow by Camunda Modeler UI of below steps:

Step1

Select the Sequence Flow A as example in Camunda Modeler(Display 11)

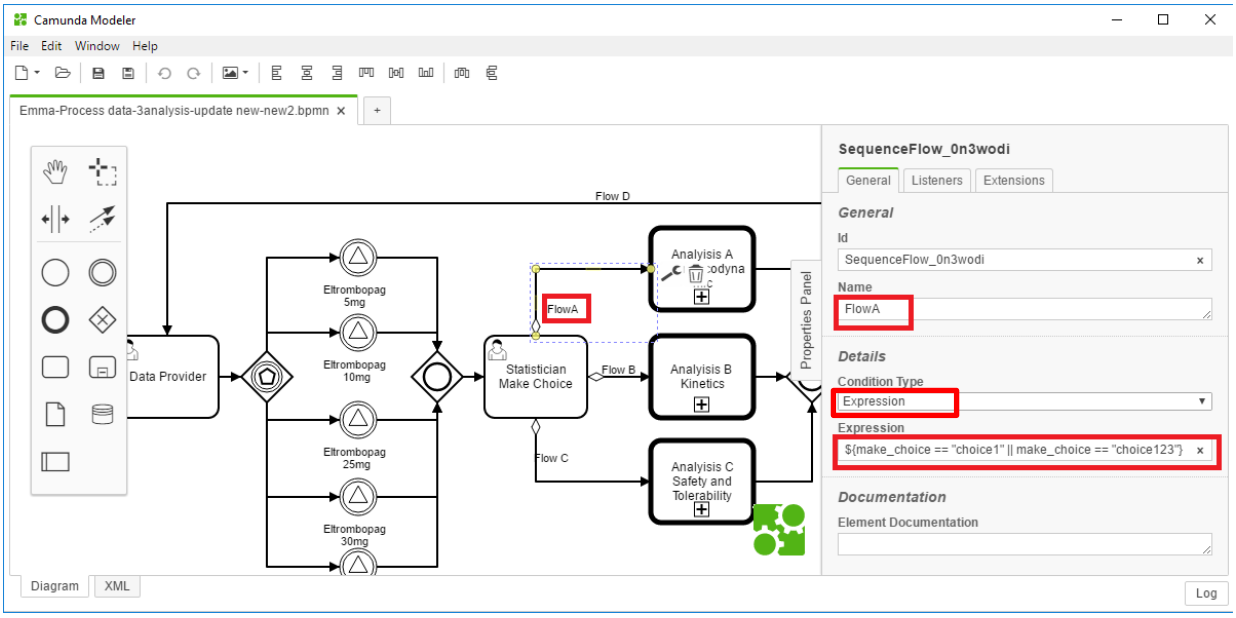
Step2

Choose Condition Type of Expression

Step3A - Add Process Data or Completion Data in the Expression

Add Expression to Sequence Flows. (Display 11) Flow A as example, add the Condition Expression of Flow A to the Expression field. (Condition Expression are being list in next section: Data Example in This Use Case)

- Flow A: ${make_choice == "choice1" || make_choice == "choice123"}

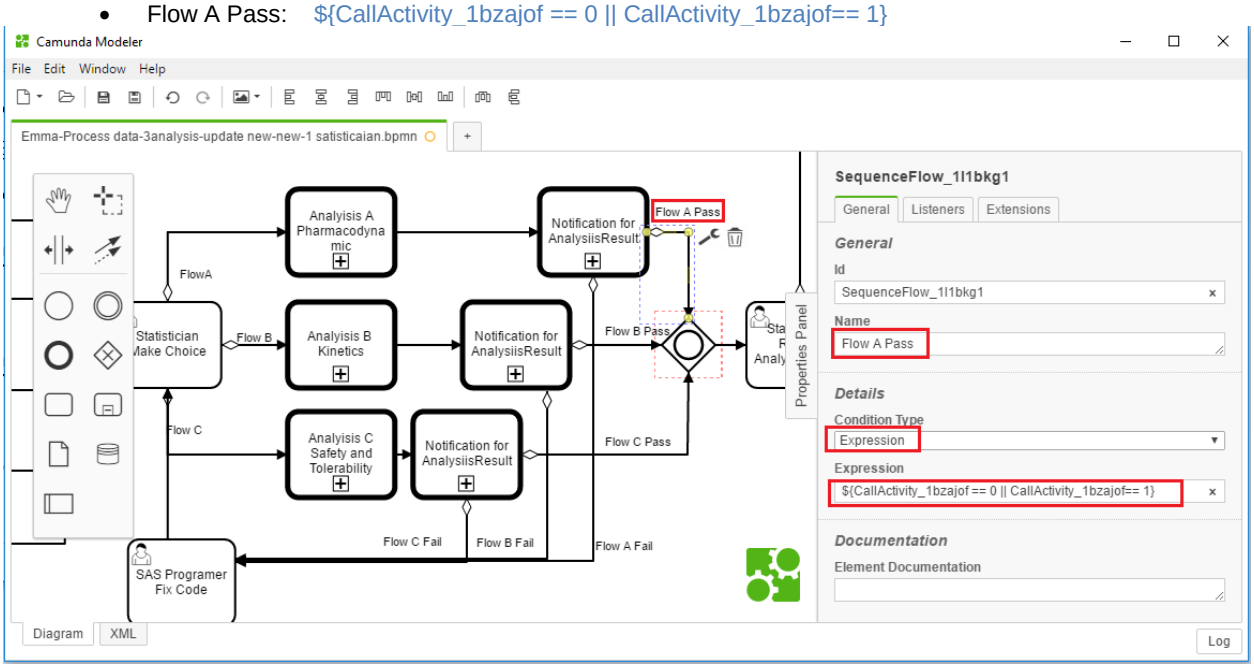


Display 11. Setup Express as condition of sequence flow (Flow A example of generate report of Analysis A)

Step3B - Add System Job Status to the Expression

Add Expression to Sequence Flows. Flow A Pass as example, add the Condition Expression of Flow A Pass to the Expression field. (Display 12)

- Flow A Pass: ${CallActivity_1bzajof == 0 || CallActivity_1bzajof== 1}



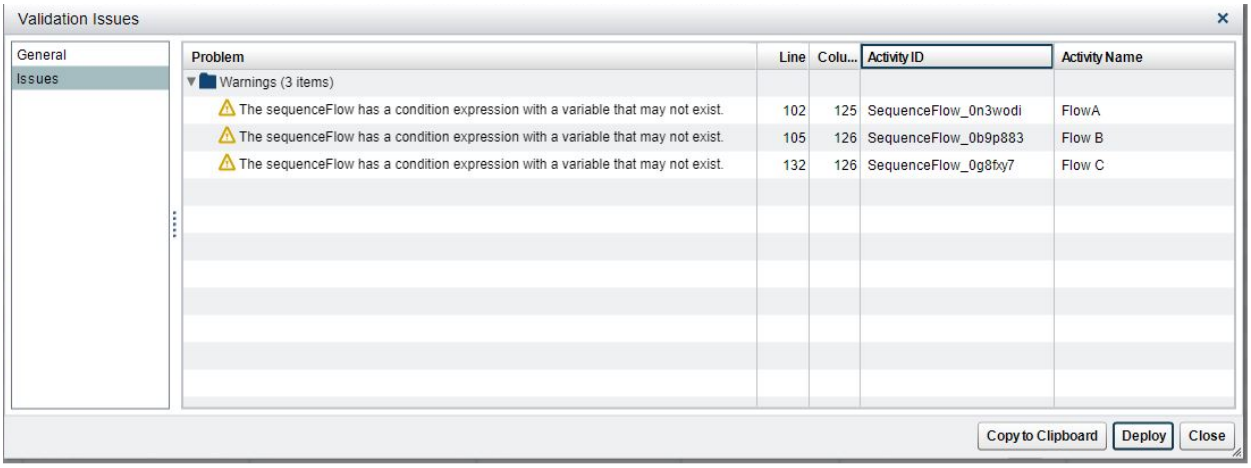
Display 12. Setup Express as condition of sequence flow (Flow A Pass example by System Job Status run pass)

DATA EXAMPLE IN THIS USE CASE

Completion Data	<camunda:formProperty id="make_choice" name="Choose the Analysis" type="enum" writable="true" default="choice123"> <camunda:value id="choice1" name="Analyisis A-Pharmacodynamic" /> <camunda:value id="choice2" name="Analyisis B-Kinetics" /> <camunda:value id="choice3" name="Analyisis C-Safety and Tolerability" /> <camunda:value id="choice123" name="Analysis A, B and C" /> </camunda:formProperty>
Process Data	<camunda:formProperty id="End_Point" name="Endpoint Day" type="long" writable="true" default="1" /> </bpmn2:extensionElements>
System Job Status	System Job status are defined by SAS Service in LSAF, user can directly use those value in the expression: -1=UNKNOWN - No other status returned (this is should only occur in the No Wait job case) 0=COMPLETED_SUCCESSFUL 1=COMPLETED_WARNINGS 2=COMPLETED_SASERRORS 3=COMPLETED_CHECKIN_ERROR 4=COMPLETED_SASWARNINGS_CHECKIN_ERROR 5=COMPLETE_SASERRORS_CHECKIN_ERROR 6=COMPLETED_FAILED 7=COMPLETED_CANCELLED 8=NO_SETUP (No job in setup) 9= INVALID_ACCOUNT
Condition Expression	**Completion Data and Process Data**- Process Variable being defined in the Condition Expression can be created from a Form Property ID; **Flow A:** ${make_choice == "choice1" || make_choice == "choice123"} **Flow B:** ${make_choice == "choice2" || make_choice == "choice123"} **Flow C:** ${make_choice == "choice3" || make_choice == "choice123"} **Flow D:** ${End_Point <= 30} **Flow E:** ${End_Point > 30} **System Job Status** – Process Variable being defined in the Condition Expression can be created from a System Job ID, and System Job Status can be directly defined in the Condition Expression **Flow A Pass:** ${CallActivity_1bzajof == 0 || CallActivity_1bzajof== 1} **Flow A Fail:** ${CallActivity_1bzajof >= 1} **Flow B Pass:** ${CallActivity_1oij34l == 0 || CallActivity_1oij34l== 1} **Flow B Fail:** ${CallActivity_1oij34l >= 1} **Flow C Pass:** ${CallActivity_0jzlek6== 0 || CallActivity_0jzlek6== 1} **Flow C Fail:** ${CallActivity_0jzlek6 >= 1}

TROUBLESHOOTING

Finally, when finish the setup in the XML with Process Data and Completion Data. Then we can deploy the XML file to SAS® LSAF. SAS® LSAF provides XML validation which validate the Errors and Warnings coming from the XML file which compliance checks against BPMN v2.0 standard. Using Display 13 as example, it validates the condition expression format is not expected. So that we can figure out and fix the problem from the XML by the exactly Line, Column and Activity ID number.



Display 13. BPMN Validation in LSAF

CONCLUSION

From the Use case, we can see that process automation has helped pharmaceutical companies break down information silos and manage data and analysis containing information about health life science professionals. BPMN provides a mapping between the graphics of the notation and the underlying constructs of execution languages which is XML languages- designed for the execution of business processes, can be visualized with a business-oriented notation of collaboration diagrams and process diagrams. By using Form Properties with your business process data and completion data in SAS® LSAF, by design different expression in conditions of Sequence Flow, we can make the decision on your clinical study result on the right way to speed the development of new therapies through optimized operational activities.

APPENDIX

BPMN INTRODUCTION

Business Process Model and Notation (BPMN) is a standard that provides a graphical notation for specifying business processes in a Business Process Diagram. It is developed by Object Management Group (OMG) and the current version is 2.0. The objective of BPMN is to support business process management, for both technical users and business users, by providing a notation that is intuitive to business users, yet able to represent complex process semantics. (See Figure 5 of BPMN Core)

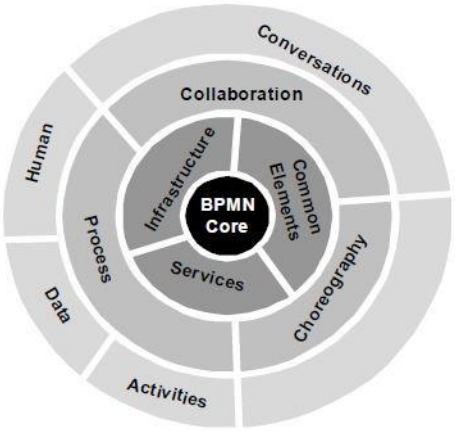


Figure 5. BPMN Core

BPMN ELEMENTS CATEGORY

BPMN depicts below four set of elements categories. The reader of a BPMN diagram can easily recognize the basic types of elements and understand the diagram.

- **Flow Objects:** Events, Activities, Gateways
- **Connecting Objects:** Sequence Flow, Message Flow, Association
- **Swim Lanes:** Pool, Lane
- **Artifacts:** Data object, Group, Annotation

Events

A trigger that starts, modifies or completes a process. Event types include message, timer, error, compensation, signal, cancel, escalation, link and others. They are shown by circles containing other symbols based on event type. They are classified as either “throwing” or “catching,” depending on their function. (Figure6)



Figure 6. Events

Activity

A particular activity or task performed by a person or system. It’s shown by a rectangle with rounded corners. They can become more detailed with sub-processes, loops, compensations and multiple instances. (Figure 7)

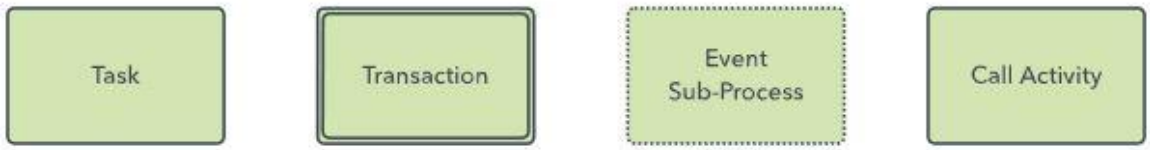


Figure 7. Activity

Gateway

Decision point that can adjust the path based on conditions or events. They are shown as diamonds. They can be exclusive or inclusive, parallel, complex, or based on data or events. (Figure 8)



Figure8. Gateway

Sequence flow

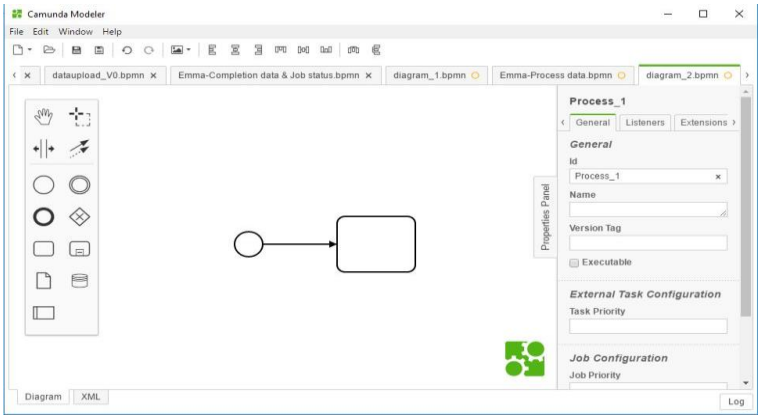
Shows the order of activities to be performed. It is shown as a straight line with an arrow. It might show a conditional flow, or a default flow. (Figure9)



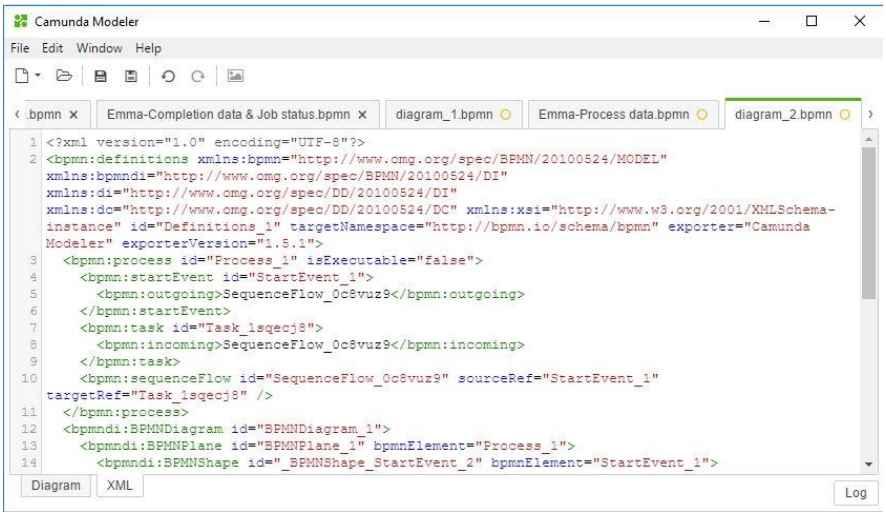
Figure 9. Sequence Flow

CAMUNDA MODELER INTRODUCTION

There are some third-party tools support to build up your business process model by BPMN 2.0. This paper use Camunda Modeler as the demonstration. It is a free a desktop tools for modeling BPMN workflows. Both business analysts and developers can directly drag and drop any elements which supports for BPMN v2.0 to design process flow (Display14). Camunda Modeler allows you to edit all properties necessary for technical execution. And finally, your work flow process is being stored as XML format (Display 15) which can be deploy to SAS® LSAF to perform data processing related works.



Display 14. Camunda Modeler-Business Process Diagram



Display 15. XML format

REFERENCES

- Business Process Model and Notation (BPMN) V2.0 -OMG Document Number: formal/2011-01-03 Standard document URL: http://www.omg.org/spec/BPMN/2.0
- SAS® Life Science Analytics Framework 4.7: User's Guide
- Phase 1 clinical study of eltrombopag, an oral, nonpeptide thrombopoietireceptor agonist<Julian M. Jenkins, Daphne Williams, Yanli Deng, Joanne Uhl, Valerie Kitchen, David Collins, and Connie L. Erickson-Miller, GlaxoSmithKline, Collegeville, PA>

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who review this paper and give valuable comments.
Specially, I would like to extend my sincere gratitude to my supervisor, Han Liu, for his constant encouragement and valuable guidance on my thesis. I am deeply grateful of his help in the completion of this thesis.

CONTACT INFORMATION

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

Name: Emma (Junru) Liu
Enterprise: SAS Research and Development (Beijing) Co., Ltd.
Address: Motorola Plaza, No. 1 Wang Jing East Road
City, State ZIP: Beijing, 100102
Work Phone: (8610) 83193355-3795
Fax: (8610) 6310-9130
E-mail: junru.liu@sas.com
Web: www.sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.