

# MINING OF AD-HOC BUSINESS PROCESSES USING MICROSOFT SHAREPOINT, NITRO & PROM 6.0

## *An Industrial Practice*

Farhad Naderipour

*Islamic Azad University, Naragh Branch, Naragh, Iran*

**Keywords:** Process mining, Business process management, Business process analysis, Microsoft SharePoint, Nitro, Prom.

**Abstract:** The improvement and re-engineering of business processes is challenging task especially when ad-hoc processes are take into account. Process mining techniques allow identification of process knowledge and characteristics based on so called event logs. Many information systems today provide such log i.e. Microsoft SharePoint which is one of the pioneers of collaborative portals. Although it supports either structured and ad-hoc process orchestration, managing dynamic processes in SharePoint is open to accept new ideas. Since SharePoint registers all change events on its content databases, it can serve as valuable source to mining process context. This paper introduce process mining concepts and techniques and demonstrates the application of Nitro to constructing event logs based on XES standard and finally apply process mining techniques using Prom 6.0 which is newly developed framework for process mining.

## 1 INTRODUCTION

SMEs (Small and Medium Sized enterprises) need to identify, manage and improve their processes to minimize their costs and stay competitive in their market. This effort requires knowledge of the business processes and gathering such knowledge takes long time and cost. Moreover, the process models revealed using traditional top-down process discovery methods often turn out to be based on what process owners and managers think that what should be done rather than what actually is done. Process mining analyzes the organization in a bottom-up manner. It uses historical data in IT Systems to discover the process model and other process information automatically.

Structured processes and ad-hoc processes (also called dynamic processes) are two ends of the business process spectrum in organizations. Structured processes have defined scope and known scenarios and can be easily identified and managed by most of traditional and modern workflow management systems. Ad-hoc processes however are those where the execution order of activities cannot be determined upfront and are led by end-user at run-time with no underlying process

definition. Email systems traditionally have been played a great role to manage this kind of processes.

The essence of dynamic processes makes process model discovery harder and also disclose critical process figures like resource usage, elapse time, and average queue time. Therefore, concepts and tools aiming at ad-hoc business processes are of the high importance.

This paper demonstrates that this is actually possible to extract information about business processes using the collaboration portal system, Microsoft SharePoint, and process mining tools such as Prom. The result is a generic approach for mining ad-hoc business processes and a concrete tool linking Microsoft SharePoint, Nitro and Prom.

## 2 PROCESS MINING

Process mining is the name of a new technology and research area to discovering process knowledge and pattern from event logs. It is mostly used to discover the real process scenarios and activity paths, conformance auditing, process performance analysis, process instance or case prediction, process improvement and social network analysis.

Table 1: Event log example.

Case ID	Task Name	Event Type	Resource	Date Time
R153-23-20-8	Search for potential vendors	Completed	Clark	2009/05/10 11:33
R153-23-20-9	Search for potential vendors	Completed	Clark	2009/05/10 11:33
R153-23-20-9	Quote price	Completed	Barbara	2009/05/12 11:44
R153-23-20-9	Get approval	Completed	Barbara	2009/05/12 12:22
R153-23-20-8	Quote price	Completed	Barbara	2009/05/12 12:23
R153-23-20-9	Search for potential vendors	Completed	Clark	2009/05/13 08:39
R153-23-20-8	Get approval	Completed	Clark	2009/05/13 08:39
R153-23-20-9	Quote price	Completed	Barbara	2009/05/24 07:12
R153-23-20-8	Cancelled	Completed	Barbara	2009/05/24 07:12
R153-23-20-9	Get approval	Completed	Barbara	2009/06/13 08:16
R153-23-20-9	Purchased	Completed	Christian	2009/06/27 08:32

```

<log>
  <trace>
    <string key="concept:name" value="R153-23-20-8" />
    <event>
      <string key="TaskName" value="Search for potential vendors" />
      <string key="EventType" value="Completed" />
      <string key="Resource" value="Clark" />
      <date key="DateTime" value="2009-05-10T11:33:00.000+03:00" />
    </event>
  </trace>
</log>

```

Listing 1: XES log example.

Nowadays, ERP systems, CRM systems, workflow management systems and collaborative portals like Microsoft SharePoint are widely used to support business process in organizations. Majority of them log events based on process instances and user actions for auditing or other purposes. This event logs and audit trails are data sources for process mining.

process model is mined in Figure 1 and can be used to understand the process domain and support the compliance.

### 3 EXTENSIBLE EVENT STREAM (XES)

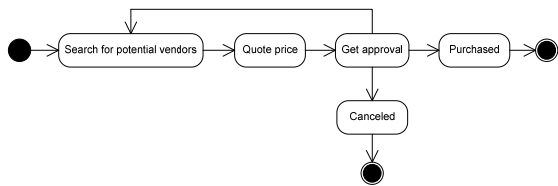


Figure 1: Mined process from Table1.

Table 1 shows a sample log which is fetched from Microsoft SharePoint databases. This log shows task and event transition in purchase order processing systems for two sample purchase orders. Each order considered as case which its activity transitions are logged in SharePoint Database and, has its own scenario and activity path. For instance, though “Get approval” state has been done twice for “R153-23-20-9” purchase order, it is only performed once for the other order. Applying process mining algorithms and techniques on Table 1 data, actual

XES (pronounces as excess), which stands for eXtensible Event Stream, is an open standard for storing and managing event log data and audit trails provided by any kind of systems. Since every system provides log entries based on its own architecture and logging mechanism which is lead to isolated solutions for logging, XES introduces an extensible schema to shape a generally acknowledged format for the event logs. As a result, audit trails can be exchanged simply among different tools and application domains. The UML diagram shown below, describes the Meta model of XES standard (Gunther, 2009). Log object at the root of the schema represents the process related information like purchase order processing, using a XBOX, etc. Each log contains arbitrary number of process instances, called trace, which has identical information about the process. Table 1 shows two traces, R153-23-20-8 and R153-23-20-8 of a purchase order process. Events are every single

activity that has been executed during the run of a process (Gunther, 2009). Each event associates to a trace object. Quote price, Get approval and purchased are sample events from Table 1.

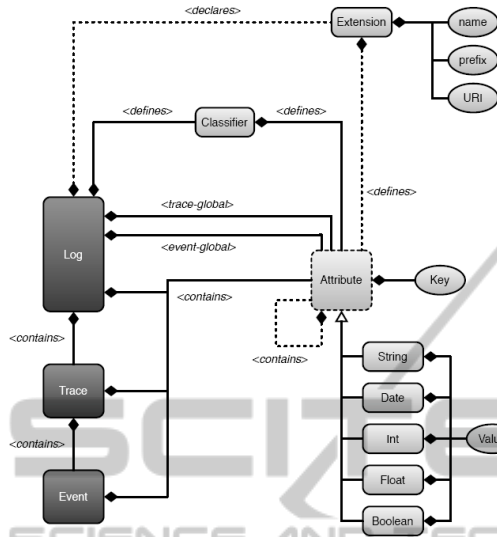


Figure 2: XES Meta model.

By now, the structure of an XES document is defined. However the log, trace and event objects have no information themselves. To store information, String, Date, Float, Integer and Boolean attributes can be define for each object. Listing 1 shows an example XES log.

#### 4 MICROSOFT SHAREPOINT AND AD-HOC PROCESSES

Microsoft SharePoint offers an easy to use workflow engine and designer to streamline business processes in a collaborative environment. Although the workflow patterns are highly customizable in Visual Studio IDE, handling of dynamic processes is far from being truly feasible in such environment.

While managing structured business processes in a practical user-friendly environment is one of the striking features of Microsoft SharePoint, Approaches toward managing unstructured process especially are slightly experimental as well as open to accept new best practices.

As mentioned earlier, if the business model prescribes the activities and their execution constraints in a complete fashion, then the process is structured. Extra budget request, for instance, might check a threshold amount, say 5000 USD, to decide whether a complex or simple approval routine is

required (Weske, 2007). Decision options for this kind of process have been define at design time.

To increase process efficiency and flexibility, processes are designed in less rigid manner. As result, dynamic or ad-hoc processes have been emerged. Activities or steps in such processes can be executed in any order and even may be repeatable either. Number of iteration in process steps is unpredictable and highly dependent on scenario at hand. Involved users and communication path between them is not formally defined. New process steps also can be defined during the execution.

SharePoint lists and libraries contain list forms that allow users to display, edit, and add items with user-defined fields to a list or library. To handle a purchase order process as example of dynamic process in SharePoint environment it is possible to use custom lists to store order information along with status field of type lookup to enable users to change order status by their own free will. By changing Order Status, users forward and backward each order freely up to finalizing each order instance.

Purchase Order Processing						
New	Actions	Settings	View: <span>سجل</span>		Quantity	LastStatus
R225-25-2-8 New	225	0805CS-151XJLC	ParNo	ParDescription	250	Waiting for Delivery
R225-25-2-8 New	225	0805CS-420XJLC	Ind-62H+5%+0.44-1.33GHZ-0805(-40°C to +125°C)	300	Waiting for Delivery	
R225-25-2-7 New	225	0805CS-960XJLC	Ind-66H+5%+0.5A-1.66GHZ-0805(-40°C to +125°C)	250	Waiting for Delivery	
R225-25-2-6 New	225	0805CS-270XJLC	Ind-27H+5%+0.5A-2.58GHZ-0805(-40°C to +125°C)	250	Waiting for Delivery	
R225-25-2-5 New	225	0805CS-220XJLC	Ind-22H+5%+0.5A-2.66GHZ-0805(-40°C to +125°C)	250	Waiting for Delivery	
R225-25-2-4 New	225	0805CS-180XJLC	Ind-18H+5%+0.6A-3.10GHZ-0805(-40°C to +125°C)	200	Waiting for Delivery	

Figure 3: SharePoint custom list to handle purchase order process.

In addition to custom fields with different data types, lists support version tracking. It logs each change in list data as a separate record in database. When a field value is changed in a list item, the system logs the new value, modifier, date and time of the change for future tracings.

Versions saved for R225-25-2-9			
All versions of this item are listed below with the new value of any changed properties.			
Delete All Versions			
No.	Modified		Modified By
2.0	2010/12/19 11:56 ق.م.		Naderipour, Farhad
		LastStatus Canceled	
1.0	2010/12/19 11:40 ق.م.		Naderipour, Farhad
		Order Number R225-25-2-9	
		Row 2,166	
		OrdreNo R225-25-2	
		CostCenter 225	

Figure 4: SharePoint version tracking feature.

Although Microsoft enables user to track all change histories through SharePoint interface, these trails can be found in WSS content database of a SharePoint site which is heart of the system either.

In view of the fact that for process mining purpose a specific format of event log would be used, having direct access to change histories would be helpful. Unfortunately, reaching these logs needs digging into data tables of WSS content database. Although WSS content database contains many tables, this section focuses on only AllUserData, AllLists and UserInfo which hold necessary data for process mining tools. Table 2 shows brief information about tables used in this paper (Ethan, 2007).

Table 2: Inspecting the SharePoint content database.

AllUserData	Holds information about all the list items for each list.
AllLists	Holds information about lists for each site.
UserInfo	Holds information about all the users for each site collection.

Table 3: AllUserData table.

tp_Id	Identifier for the list item, uniquely identifying it within the AllUserData table.
tp_ListId	List Identifier of the list or document library containing the list item.
tp_SiteId	Identifier of the site collection containing the list item.
tp_Version	A counter incremented any time a change is made to the list item, used for internal conflict detection. Due to the mapping of application properties to the generic columns schema in this table, changes to application schema as well as property values can affect a version increment.
tp_Author	Identifier for the user who created the list item.
tp_Editor	Identifier for the user who last edited the list item.
tp_Modified	A date and time value specifying when this list item was last modified.
tp_Created	A date and time value specifying when this list item was created.
nvarchar#	Columns for application-defined fields that hold values of type nvarchar. The 64 columns are named nvarchar1 to nvarchar64. If the column does not contain data, this value MUST be NULL.
int#	Columns for application-defined fields that hold values of type int. The 16 columns are named int1 to int16. If the column does not contain data, this value MUST be NULL.
float#	Columns for application-defined fields that hold values of type float. The 12 columns are named float1 to float12. If the column does not contain data, this value MUST be NULL.

The detail description of whole content database is beyond the scope of this article. Instead, let's focus on the AllUserData table as a place where all list item data is stored. This important table has 192 columns and all list items including their history are stored in this single table. Table3 describes the more important columns. For a complete list of columns in the AllUserData table, see MSDN documentations at: [http://msdn.microsoft.com/en-us/library/dd358229\(v=prot.13\).aspx](http://msdn.microsoft.com/en-us/library/dd358229(v=prot.13).aspx).

As can be seen, there are group of columns, including nvarchar#, ntext#, int#, in which the values of SharePoint lists are stored. Every time you create a new column in a list, it is automatically mapped to a "free" column of the desired type in the AllUserData table. If you add column "Order No" of type single line to your list for instance, an unused database column in the range from nvarchar1 to nvarchar 64 will be assigned to store this field data (Krause et al., 2010).

The first thing to do is to identify the relevant data in AllUserData table by query the table for ID of the list needed. Since lists information are stored in AllLists table, as shown in listing 2, filtering this table could return the GUID of the lists.

```
SELECT tp_ID
FROM AllLists
WHERE tp_Title = 'You lists title'
```

Listing 2: Query AllLists to get list GUID.

Then, using returned tp\_Id from the AllLists table in query criterion of AllUserData can fetch the list item data (See listing 3).

```
SELECT tp_ID AS [Item ID], tp_ListId AS [List ID], tp_Version AS [Version], tp_Author AS [CreatedBy], tp_Editor AS [Modified By], tp_Modified AS [Modified], tp_Created AS [Created], nvarchar1 AS [Order No], nvarchar9 AS [Description], nvarchar23 AS [Status]
FROM AllUserData
WHERE (tp_ListId = '39158042-39dc-4951-9400-03f8f45893b3')
```

Listing 3: Query list item data.

The only remaining challenge is to get the name of the mapped database columns from SharePoint list. To overcome this issue, you need to filter AllUserData for a List ID and then find related columns. For example, Status filed data of Purchase Order Processing list is mapped to nvarchar23.

After applying Listing 3 commands, the results may not be completely satisfactory. Because, this

article looks for only status change trails through the whole historical data but SharePoint preserves and shows any changes occurred on list items. To eliminate rows with repetitive status data, the prior query must be trimmed like Listing 4, to reach a neat status changes for each list item or case.

```
SELECT
UDMain.tp_ID AS [Item ID],
UDMain.tp_ListId AS [List ID],
Count(UDMain.tp_Version) AS [Version],
UDMain.tp_Editor AS [Modified By],
Min(UDMain.tp_Modified) AS [Modified],
UDMain.nvarchar1 AS [Order No],
UDMain.nvarchar9 AS [Description],
UDMain.nvarchar23 AS [Status]

FROM AllUserData AS UDMain
WHERE
(UDMain.tp_ListId = '39158042-39dc-4951-9400-03f8f45893b3')
And (UDMain.tp_RowOrdinal = 0)
GROUP BY
UDMain.tp_ID,
UDMain.tp_ListId,
UDMain.tp_Editor,
UDMain.nvarchar1,
UDMain.nvarchar9,
UDMain.nvarchar23
ORDER BY UDMain.tp_ID
```

Listing 4: Pure status change data.

As mentioned earlier, UserInfo table contains information about all portal users including their full name. To replace user ID data in Modified By column with user full name, the table user info have to be joined with AllUserData.

## 5 CONVERTING SHAREPOINT DATA TO XES USING NITRO

In previous sections, the important role of constructed event log, XES, and fundamental approach for querying content database to reach raw material for building the event stream have been elaborated. Now, it is time to serialize SharePoint list item histories according to XES data model (See figure 2).

This conversion involves programming activities which could be usually time-taking. Instead, using Fluxicon’s tool, Nitro, can seed-up the process. Nitro maps and converts variety of event-based data to generally acknowledged XES or MXML event logs. For further information about Nitro visit <http://fluxicon.com/nitro/>.

The query results of SharePoint content database saved in Microsoft Excel or exported as CSV file serves as input for Nitro. Load the excel file data in Nitro and map the columns as follow:

Table 4: Nitro column mapping.

Column Name	Map to	Description
Order No	Case ID	Process instances are introduced with Case ID.
Modified By	Resource	The resource that performed an activity in each process instance is making known here.
Modified	Timestamp (Completion of activity)	The finish time of the activity is mapped here.
Status	Activity	-

The Start of activity column which is second option of Timestamp can be set to Ignored Column in this case. Now it is time to start conversion and export data as either XES (for Prom 6.0) or MXML (for Prom 5.0) format.

## 6 MINING THE PROCESS USING PROM 6.0

ProM 6.0 is an open source framework for implementing process mining algorithms and techniques in a standard environment. By importing event logs, practitioners can examine the process using various ready to use plug-ins provided by the ProM. New functionalities also can be added to the framework. Detailed information on ProM 6.0 can be obtained from <http://www.processmining.org>.

Many business process improvement and re-engineering projects spend considerable time on discovering running processes in organizations. It is sometimes hard to verify that running business processes comply with regulations and procedures as they were originally governed by top management. Prom 6.0 makes it possible to visualize process using running information systems in organizations. Based on purchase order processing system log converted by Nitro, “Mine SPD (Simple Precedence Diagram) Model” plug-in shapes the process diagram.

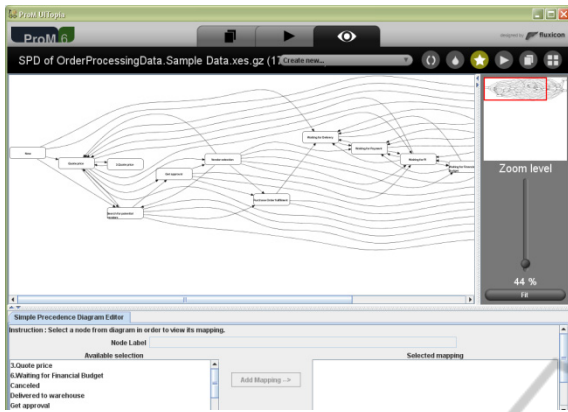


Figure 5: SPD model mined from sample event log.

Social Network Analysis is another analytical data that can be fetched from ProM. Social Network Analysis answers which employee receives more work, who sends more workload to others and who is isolated in organization or process boundaries. Figure 6 illustrates simple SNA report derived from order processing example.

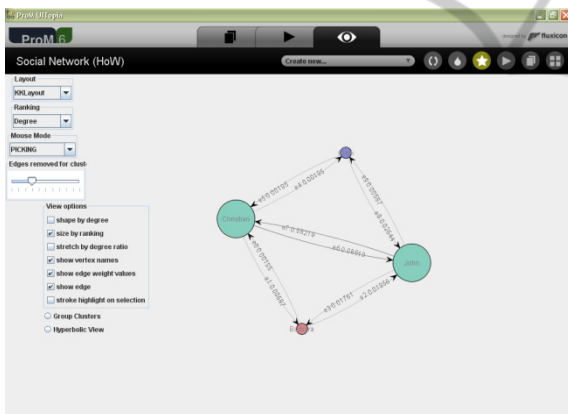


Figure 6: Social network diagram mined from sample event log.

There are still much more information, like work load analysis and dotted chart analysis, related to each process that can be obtained from event trails. To see the distribution of events over time for instance, Dotted chart analysis add-in plots and spreads event occurrences over time line (see Figure 7). It shows density of events, performance metrics, gives an overall insight into a process and reveals interesting process patterns.

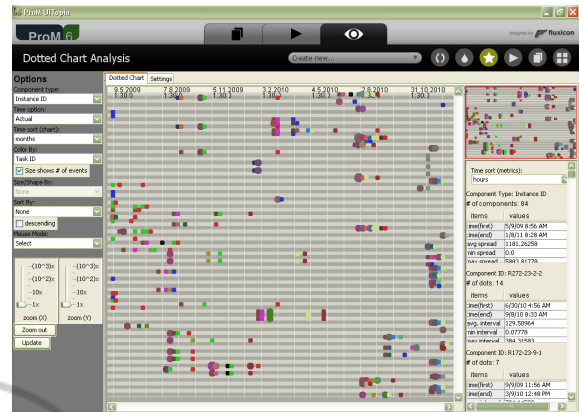


Figure 7: Dotted chart representing event distribution.

## 7 CONCLUSIONS

To sum up, process mining is very helpful in situations where process steps are logged in IT systems. Using Microsoft SharePoint, a process-aware collaborative portal, gathering and logging process steps could be surviving fast practice in case where no information is available in IT systems. The resulting data stored in SharePoint content databases formed in XES XML format by Nitro that can be read by process mining tools such as Prom 6.0 which is introduced in this paper.

## REFERENCES

H. M. W. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, *XES Tools*. Technische Universiteit Eindhoven, Department of Mathematics and Computer Science.

H. M. W. (Eric) Verbeek, R. P. Jagadeesh Chandra Bose, *ProM 6 Tutorial*. August 2010.

C. W. Gunther, 2009. *XES Standard Definition*. Fluxicon Process Laboratories.

Mathias Weske, 2007. *Business Process Management: Concepts, Languages, Architectures*. Springer.

Ethan, 2007. *Inspecting the SharePoint Content Database*. <http://vspug.com/ethan/2007/09/16/inspecting-the-sharepoint-content-database/>

Joerg Krause, Christian Langhirt, Martin Döring, Alexander Sterff, Bernd Pehl, 2010. *SharePoint 2010 as a Development Platform*. Apress.