Development of ITSM BMC Log File Analyzer

Suma Patil[1] Mrs.Anala.M.R[2] Nelakanti Gangadhara Sanya[3]
[1]M.tech Student [2]Associate Professor [3] Project Manager
[1,2]Department Of Computer Science
[1,2] R.V. Collage of Engineering, Bangalore, India
[3]Unisys Global Services, Whitefield, India

Abstract— Every organization consists of servers for processing huge amount of data. These servers are built with proprietary technologies, so the only way to debug any application issues is to go through the log files generated and find out which workflow fired/not fired. Problem with this approach is that these log files grow very quickly depending on user activity and type of logging. Many times these log files grow to 200-300 Mbytes within 4-5 minutes. Also, since these are server side logs, they contain workflows executed for all users and not just the user who is having issues. To physically go through such huge log files is very painful and consumes lot of time as well. In the process productivity of developer also gets affected.

To resolve this we propose an application that can parse through these log files and trace the execution of workflows for given user and present the same in a graphical & intuitive interface. This would greatly reduce time & effort spent by developers in troubleshooting application issues.

Key Words: workflow, log file, graphical interface, BMC.

I. INTRODUCTION

BMC Remedy AR server maintains a record of all its transactions in log files. These log files are generated by AR server which is built with proprietary technologies.

BMC Remedy AR server maintains different types of log files such as filter logs, SQL logs, escalation logs etc.

ITSM uses BMC's Remedy AR (Action Request) System to provide a consolidated Service Process Management platform for automating and managing Service Management business processes. Incidents will be raised in this BMC Remedy by the clients. All the transactions that are performed in BMC Remedy will be logged in its files.

If any error occurs in this BMC Remedy the only way to debug its application issue is to go through the log files generated by BMC Remedy AR Server. These log files grow very depending upon the user actions. To physically go through the log file is really a tedious process for the developer to debug the issues and consumes lot of time. In corporate environment time management is very important.

Hence this application developed is capable of parsing log files that are generated by BMC Server and trace the execution of workflows. Workflow management was originally a tool for the organization, analysis and Rearrangement of business processes. Information technology has gone through one of the most significant changes in the last decades while having proliferated and penetrated into business processes in each and every business field and has become a determining component. In parallel the application of workflow has improved as well. Further application fields include a wide range of sectors ranging from technical design via solving of various computing problems to several application of the processing of information data.

II. NEED FOR A LOG FILE PARSER

A log file is a recording of everything that goes in and out of a particular server. It is a concept much like the black box of an airplane that records everything going on with the plane in the event of a problem. The information is frequently recorded chronologically, and is located in the root directory, or occasionally in a secondary folder, depending on how it is set up with the server. The only person who has regular access to the log files of a server is the server administrator, and a log file is generally password protected, so that the server administrator has a record of everyone and everything that wants to look at the log files for a specific server. Many times these log files grow to 200-300 Mbytes within 4-5 minutes. Also, since these are server side logs, they contain workflows executed for all users and not just the user who is having issues. To physically go through such huge log files is very painful and consumes lot of time as well. In the process productivity of developer also gets affected.

To resolve this we need log file parser that can parse through these log files and trace the execution of workflows for given user and present the same in a graphical & intuitive interface. This would greatly reduce time & effort spent by developers in troubleshooting application issues.

III. STATEMENT OF THE PROJECT PROBLEM

The BMC application is used by around 300000+ users and at any given point there could be up to 3000 users using concurrently .Since these servers are built with proprietary technologies, only way to debug any application issues is to go through the log files generated and find out which workflow fired/not fired.

Problem with this approach is that these log files grow very quickly depending on user activity and type of logging. Many times these log files grow to 200-300 Mbytes within 4-5 minutes. Also, since these are server side logs, they contain workflows executed for all users and not just the user who is having issues. To physically go through such huge log files is very painful and consumes lot of time as well. In the process productivity of developer also gets affected.

IV. PROPOSED SYSTEM

To resolve this we propose to develop an application that can parse through these log files and trace the execution of workflows for given user and present the same in a graphical & intuitive interface. This would greatly reduce time & effort spent by developers in troubleshooting application issues. The necessary methods required to parse the huge log file and workflows generated by them by checking which of them(workflow) are fired or not fired are represented graphically so that it will be easy for the user as it saves time of going through the log files to debug the error.

The application developed has the capabilities:

A. Capable of parsing proprietary log files generated by BMC Remedy Applications-

B. Capable of parsing log files up to 1 GB of size.

C. Capable of identifying workflow execution per user.

D. Display execution of workflows in a graphical & intuitive manner.

E. It has the ability to highlight workflows on User Interface(UI) and shows following details about the workflow when highlighted:

– Whether selected workflow is executed or not.
– Whether selected workflow is enabled or disabled.
– Workflow execution condition.
– Value of parameters set while executing this condition.
– Actions performed if this workflow is executed.

The proposed system working architecture is as shown in figure 1. Here the user extracts the logs from BMC AR server by connecting to it, the extracted log files are then parsed and represented in a form which is useful to the developer in troubleshooting the issues.

Figure 2 represents the flow diagram of the log file analyzer where the AR server maintains the log files in it.

The user extracts the logs from AR Server and gives this file as input to the parser, the parser then parses the log files based on user transaction like if he wants to parse the file based on Thread ID, or based on User and represents them in a graphical manner (tree structure), a form which is useful for the developer in debugging the issues and hence reduces the pain of the developer instead of going through the whole log file which is a very tedious process.

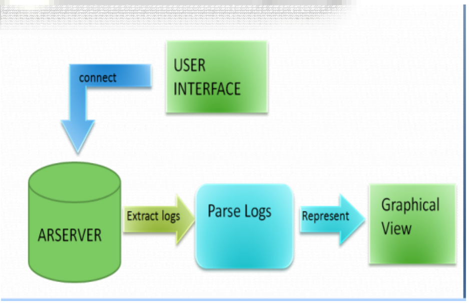


Fig. 1: Working architecture diagram of BMC Remedy Log Analyser.

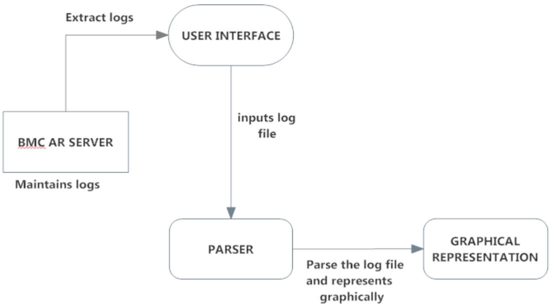


Fig. 2: Flow diagram for BMC Remedy Log Analyser.

V. RESULT

The main page of the tool is shown in the figure 3, where the user provides his provides the log file generated by BMC AR Server. The log file is selected by clicking on browse button through a file chooser.

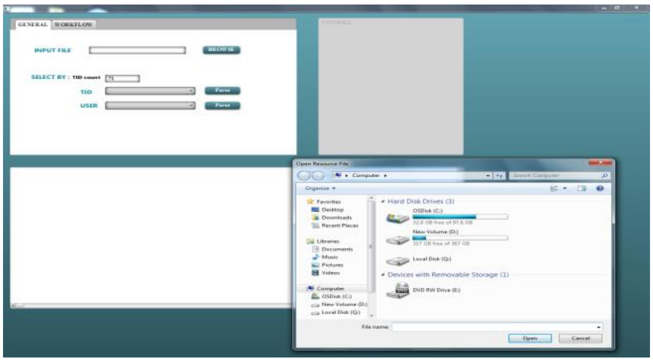
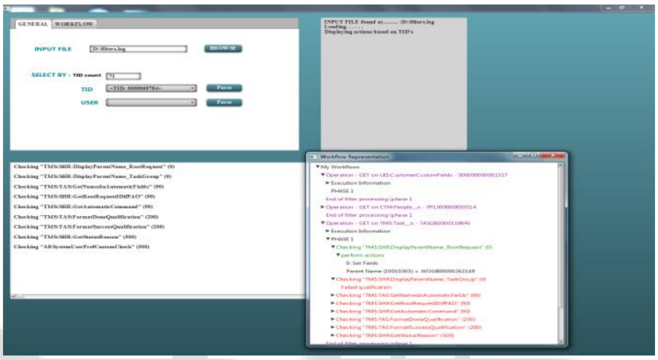


Fig. 3: Main screen.

The parsing of log file is done based on Thread ID and also based on Users present in the log file. As seen in Figure 4, parsed graphical representation of log file based on Thread ID is done.



INPUT FILE: D:\filters.log

TID count: 71

TID: <TID: 0000069784>

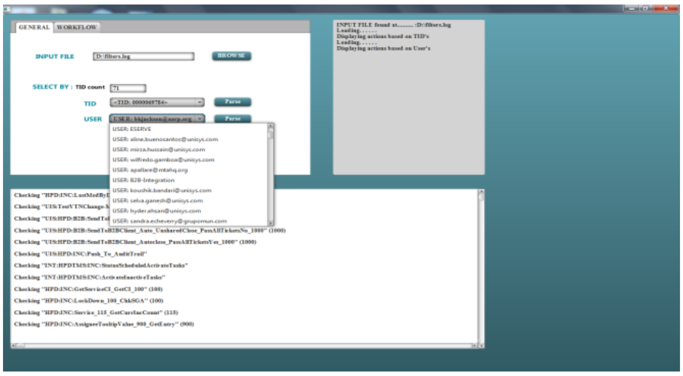
```
INPUT FILE found at....... :D:\filters.log
Loading......
Displaying actions based on TID's
```

```
Checking "TMS:SHR:DisplayParentName_RootRequest" (0)
Checking "TMS:SHR:DisplayParentName_TaskGroup" (0)
Checking "TMS:TAS:GetNamesInAutomaticFields" (90)
Checking "TMS:SHR:GetRootRequestIDifPAO" (90)
Checking "TMS:SHR:GetAutomaticCommand" (90)
Checking "TMS:TAS:FormatDoneQualification" (200)
Checking "TMS:TAS:FormatSuccessQualification" (200)
Checking "TMS:SHR:GetStatusReason" (900)
Checking "ARSystemUserPrefCustomCheck" (800)
```

```
My Workflows
  Operation - GET on UIS:CustomerCustomFields - 000000000001517
    Execution Information
    PHASE 1
  End of filter processing (phase 1
  Operation - GET on CTM:People_o - PPL000000893014
  End of filter processing (phase 1
  Operation - GET on TMS:Task_o - TAS000000119849
    Execution Information
    PHASE 1
      Checking "TMS:SHR:DisplayParentName_RootRequest" (0)
        perform actions
          0: Set Fields
          Parent Name (10010305) = WOG000000262149
      Checking "TMS:SHR:DisplayParentName_TaskGroup" (0)
        Failed qualification
      Checking "TMS:TAS:GetNamesInAutomaticFields" (90)
      Checking "TMS:SHR:GetRootRequestIDifPAO" (90)
      Checking "TMS:SHR:GetAutomaticCommand" (90)
      Checking "TMS:TAS:FormatDoneQualification" (200)
      Checking "TMS:TAS:FormatSuccessQualification" (200)
      Checking "TMS:SHR:GetStatusReason" (900)
  End of filter processing (phase 1
```

Fig. 4: Input screen to select workflow based on Thread ID.

By selecting the user, the workflow for that particular user can be represented graphically. Figure 5 shows the list of users present in the selected log file.



INPUT FILE: D:\filters.log

TID count: 71

TID: <TID: 0000069784>

USER: bkjackson@aarp.org

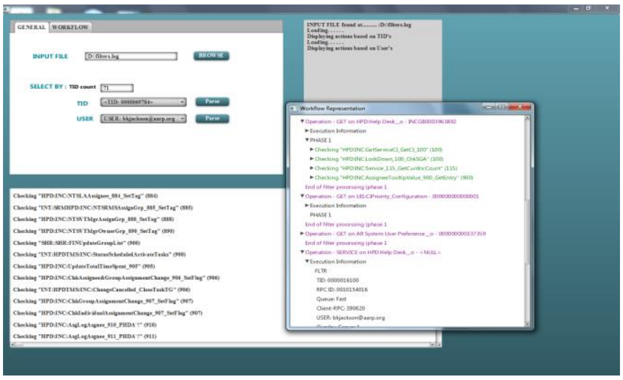
```
USER: ESERVE
USER: aline.buenosantos@unisys.com
USER: mirza.hussain@unisys.com
USER: wilfredo.gamboa@unisys.com
USER: apallare@mtahq.org
USER: B2B-Integration
USER: koushik.bandari@unisys.com
USER: selva.ganesh@unisys.com
USER: hyder.ahsan@unisys.com
USER: sandra.echeverry@grupomun.com
```

```
INPUT FILE found at....... :D:\filters.log
Loading......
Displaying actions based on TID's
Loading......
Displaying actions based on User's
```

```
Checking "HPD:INC:LastModByD
Checking "UIS:TestVTNChange-N
Checking "UIS:HPD:B2B:SendTo
Checking "UIS:HPD:B2B:SendToB2BClient_Auto_UnsharedClose_PassAllTicketsNo_1000" (1000)
Checking "UIS:HPD:B2B:SendToB2BClient_Autoclose_PassAllTicketsYes_1000" (1000)
Checking "UIS:HPD:INC:Push_To_AuditTrail"
Checking "INT:HPDTMS:INC:StatusScheduledActivateTasks"
Checking "INT:HPDTMS:INC:ActivateInactiveTasks"
Checking "HPD:INC:GetServiceCI_GetCI_100" (100)
Checking "HPD:INC:LockDown_100_ChkSGA" (100)
Checking "HPD:INC:Service_115_GetCurrIncCount" (115)
Checking "HPD:INC:AssigneeTooltipValue_900_GetEntry" (900)
```

Fig. 5: Input screen to select workflow based on User.

Here in Figure 6, the workflows for selected user are displayed in the new window. The area below displays the list of all filters names that are triggered for the selected user workflow.



```
INPUT FILE  D:\filters.log
TID count  71
TID  <TID: 0000069784>
USER  USER: bkjackson@aarp.org
```

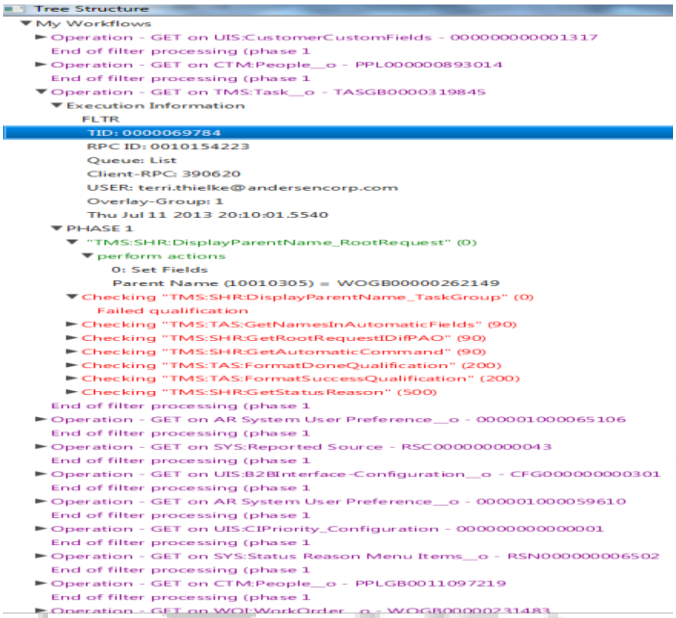
```
INPUT FILE found at....... :D:\filters.log
Loading......
Displaying actions based on TID's
Loading......
Displaying actions based on User's
```

```
Checking "HPD:INC:NTSLAAssignee_884_SetTag" (884)
Checking "INT:SRMHPD:INC:NTSRMSAssignGrp_885_SetTag" (885)
Checking "HPD:INC:NTSVTMgrAssignGrp_888_SetTag" (888)
Checking "HPD:INC:NTSVTMgrOwnerGrp_890_SetTag" (890)
Checking "SHR:SHR:FINUpdateGroupList" (900)
Checking "INT:HPDTMS:INC:StatusScheduledActivateTasks" (900)
Checking "HPD:INC:UpdateTotalTimeSpent_905" (905)
Checking "HPD:INC:ChkAssignee&GroupAssignmentChange_906_SetFlag" (906)
Checking "INT:HPDTMS:INC:ChangeCancelled_CloseTaskTG" (906)
Checking "HPD:INC:ChkGroupAssignmentChange_907_SetFlag" (907)
Checking "HPD:INC:ChkIndividualAssignmentChange_907_SetFlag" (907)
Checking "HPD:INC:AsgLogAsgnee_910_PHDA`!" (910)
Checking "HPD:INC:AsgLogAsgnee_911_PHDA`!" (911)
```

Workflow Representation

```
▼ Operation - GET on HPD:Help Desk__o - INCGB0003963892
  ► Execution Information
  ▼ PHASE 1
    ► Checking "HPD:INC:GetServiceCI_GetCI_100" (100)
    ► Checking "HPD:INC:LockDown_100_ChkSGA" (100)
    ► Checking "HPD:INC:Service_115_GetCurIncCount" (115)
    ► Checking "HPD:INC:AssigneeTooltipValue_900_GetEntry" (900)
  End of filter processing (phase 1
▼ Operation - GET on UIS:CIPriority_Configuration - 000000000000001
  ► Execution Information
    PHASE 1
  End of filter processing (phase 1
► Operation - GET on AR System User Preference__o - 000000000037359
  End of filter processing (phase 1
▼ Operation - SERVICE on HPD:Help Desk__o - <NULL>
  ▼ Execution Information
      FLTR
       TID: 0000016100
       RPC ID: 0010154016
       Queue: Fast
       Client-RPC: 390620
       USER: bkjackson@aarp.org
```

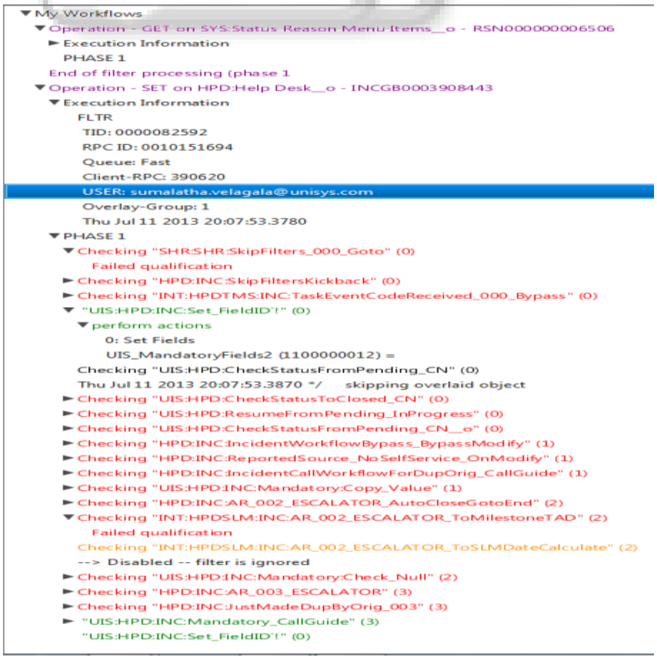
Fig. 6: Workflow for the selected user is displayed.



Tree Structure

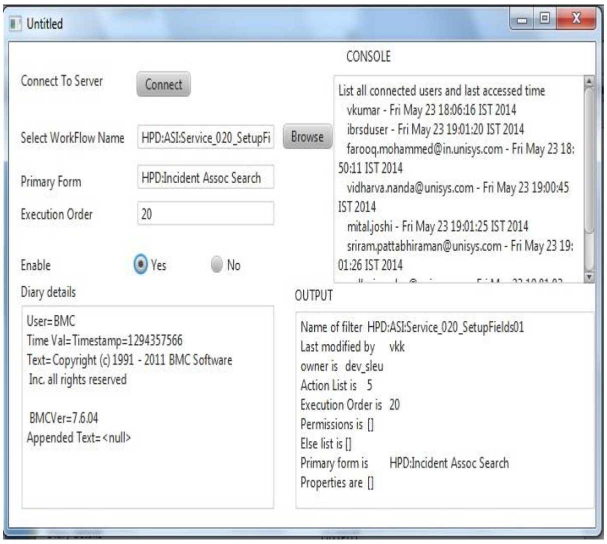
```
▼ My Workflows
  ► Operation - GET on UIS:CustomerCustomFields - 000000000001317
    End of filter processing (phase 1
  ► Operation - GET on CTM:People__o - PPL000000893014
    End of filter processing (phase 1
  ▼ Operation - GET on TMS:Task__o - TASGB0000319845
    ▼ Execution Information
        FLTR
         TID: 0000069784
         RPC ID: 0010154223
         Queue: List
         Client-RPC: 390620
         USER: terri.thielke@andersencorp.com
         Overlay-Group: 1
         Thu Jul 11 2013 20:10:01.5540
    ▼ PHASE 1
      ▼ "TMS:SHR:DisplayParentName_RootRequest" (0)
        ▼ perform actions
            0: Set Fields
            Parent Name (10010305) = WOGB00000262149
      ▼ Checking "TMS:SHR:DisplayParentName_TaskGroup" (0)
          Failed qualification
      ► Checking "TMS:TAS:GetNamesInAutomaticFields" (90)
      ► Checking "TMS:SHR:GetRootRequestIDifPAO" (90)
      ► Checking "TMS:SHR:GetAutomaticCommand" (90)
      ► Checking "TMS:TAS:FormatDoneQualification" (200)
      ► Checking "TMS:TAS:FormatSuccessQualification" (200)
      ► Checking "TMS:SHR:GetStatusReason" (500)
    End of filter processing (phase 1
  ► Operation - GET on AR System User Preference__o - 000001000065106
    End of filter processing (phase 1
  ► Operation - GET on SYS:Reported Source - RSC000000000043
    End of filter processing (phase 1
  ► Operation - GET on UIS:B2BInterface-Configuration__o - CFG000000000301
    End of filter processing (phase 1
  ► Operation - GET on AR System User Preference__o - 000001000059610
    End of filter processing (phase 1
  ► Operation - GET on UIS:CIPriority_Configuration - 000000000000001
    End of filter processing (phase 1
  ► Operation - GET on SYS:Status Reason Menu Items__o - RSN000000006502
    End of filter processing (phase 1
  ► Operation - GET on CTM:People__o - PPLGB0011097219
    End of filter processing (phase 1
  ► Operation - GET on WOI:WorkOrder__o - WOGB00000231483
```

Fig. 7: Complete view of tree structure workflow based on Thread ID.



```
▼ My Workflows
  ▼ Operation - GET on SYS:Status Reason Menu Items__o - RSN000000006506
    ► Execution Information
      PHASE 1
    End of filter processing (phase 1
  ▼ Operation - SET on HPD:Help Desk__o - INCGB0003908443
    ▼ Execution Information
        FLTR
         TID: 0000082592
         RPC ID: 0010151694
         Queue: Fast
         Client-RPC: 390620
         USER: sumalatha.velagala@unisys.com
         Overlay-Group: 1
         Thu Jul 11 2013 20:07:53.3780
    ▼ PHASE 1
      ▼ Checking "SHR:SHR:SkipFilters_000_Goto" (0)
          Failed qualification
      ► Checking "HPD:INC:SkipFiltersKickback" (0)
      ► Checking "INT:HPDTMS:INC:TaskEventCodeReceived_000_Bypass" (0)
      ▼ "UIS:HPD:INC:Set_FieldID`!" (0)
        ▼ perform actions
            0: Set Fields
            UIS_MandatoryFields2 (1100000012) =
        Checking "UIS:HPD:CheckStatusFromPending_CN" (0)
        Thu Jul 11 2013 20:07:53.3870 */     skipping overlaid object
      ► Checking "UIS:HPD:CheckStatusToClosed_CN" (0)
      ► Checking "UIS:HPD:ResumeFromPending_InProgress" (0)
      ► Checking "UIS:HPD:CheckStatusFromPending_CN__o" (0)
      ► Checking "HPD:INC:IncidentWorkflowBypass_BypassModify" (1)
      ► Checking "HPD:INC:ReportedSource_NoSelfService_OnModify" (1)
      ► Checking "HPD:INC:IncidentCallWorkflowForDupOrig_CallGuide" (1)
      ► Checking "UIS:HPD:INC:Mandatory:Copy_Value" (1)
      ► Checking "HPD:INC:AR_002_ESCALATOR_AutoCloseGotoEnd" (2)
      ▼ Checking "INT:HPDSLM:INC:AR_002_ESCALATOR_ToMilestoneTAD" (2)
          Failed qualification
        Checking "INT:HPDSLM:INC:AR_002_ESCALATOR_ToSLMDateCalculate" (2)
        --> Disabled -- filter is ignored
      ► Checking "UIS:HPD:INC:Mandatory:Check_Null" (2)
      ► Checking "HPD:INC:AR_003_ESCALATOR" (3)
      ► Checking "HPD:INC:JustMadeDupByOrig_003" (3)
      ► "UIS:HPD:INC:Mandatory_CallGuide" (3)
        "UIS:HPD:INC:Set_FieldID`!" (0)
```

Fig. 8: Complete view of tree structure workflow based on User.



Connect To Server

Select WorkFlow Name: HPD:ASI:Service_020_SetupFi

Primary Form: HPD:Incident Assoc Search

Execution Order: 20

Enable ● Yes ○ No

Diary details

```
User=BMC
Time Val=Timestamp=1294357566
Text=Copyright (c) 1991 - 2011 BMC Software
 Inc. all rights reserved

 BMCVer=7.6.04
Appended Text=<null>
```

```
List all connected users and last accessed time
  vkumar - Fri May 23 18:06:16 IST 2014
  ibrsduser - Fri May 23 19:01:20 IST 2014
  farooq.mohammed@in.unisys.com - Fri May 23 18:
50:11 IST 2014
  vidharva.nanda@unisys.com - Fri May 23 19:00:45
IST 2014
  mital.joshi - Fri May 23 19:01:25 IST 2014
  sriram.pattabhiraman@unisys.com - Fri May 23 19:
01:26 IST 2014
```

```
Name of filter  HPD:ASI:Service_020_SetupFields01
Last modified by    vkk
owner is  dev_sleu
Action List is   5
Execution Order is  20
Permissions is []
Else list is []
Primary form is      HPD:Incident Assoc Search
Properties are  []
```

Fig. 9: Display of workflow information collected from the server

VI. CONCLUSION

This project will be useful in corporate environment for folks working with remedy log files and facing problems in troubleshooting the issues while parsing the logs and for tracing the workflows for different users. This indeed reduces the effort of the developers in solving the issues and reduces time as well.

REFERENCES

[1] Dapeng Liu; Shaochun Xu; Huafu Liu "Using Log Files as Knowledge Bases " Industrial and Information Systems (ICIIS), in 6th IEEE International Conference on Advanced Applied Informatics, 2011 On page(s): 130 – 135

[2] P.W.D.C. Jayathilake, "A Novel Mind Map Based Approach for Log Data Extraction" in 6th International Conference on Industrial and Information Systems, 2011, ICIIS 2011, Aug. 16-19, 2011, Sri Lanka

[3] B. Beizer, Software Testing Techniques, 2nd Edition, 2 Sub. Intl Thomson Computer Pr (T), 1990

[4] "Log4j." [Online]. Available: http://logging.apache.org/log4j/1.2

[5] J. H. Andrews, "Testing using log file analysis: tools, methods, and issues," in 13th IEEE International Conference on Automated Software Engineering, 1998. Proceedings, 1998, pp. 157–166.

[6] J. H. Andrews, "Testing using log file analysis: tools, methods and issues," Proc. 13th IEEE International Conference on Automated Software Engineering, Oct. 1998, pp. 157-166.

[7] James H. Andrews, "Testing using Log File Analysis: Tools, Methods, and Issues"

[8] J.H.Andrews. Theory and practice of log file analysis. Technical Report 524, Department of Computer Science, University of Western Ontario, May 1998

[9] BMC Remedy Action Request System 7.6.04 Concepts Guide

[10] L.K. Dillon, G. Kutty, L. E.Moser, P.M.Melliar-Smith, and Y. S. Ramakrishna. Graphical interval logic for specifying concurrent systems. ACM Transactions on Software Engineering and Methodology, 3(2):131–165, April 1994.