Glassbeam Search: Big Data Analytics & Object Oriented UX framework

Jyotirmaya Mahapatra
User Experience Designer
Clarice Technologies Limited
Jyotirmaya.mahapatra@claricetechnologies.com
+91 7259884025

Porus Rathore
User Experience Manager
Clarice Technologies Limited
Porus.rathore@claricetechnologies.com
+91 9886691636

Swati Gupta
User Experience Designer
Clarice Technologies Limited
swati.gupta@claricetechnologies.com
+91 9740078868

Pramod Sridharamurthy
Product Manager IT
Glassbeam INC.
Pramod.sridharamurthy@glassbeam.com
+91 9341327829

ABSTRACT

Glassbeam Support is a cloud based Machine log data analytics and search application. It allows Data centers, Networking servers, OEMs and IT Enterprises to quickly search for data sources and insights related to any machine.

Due to the complexity of data within log files, the existing interface supported a lot of features, which were not aligned to actual user behavior. This made the product complex and various features were left unexplored. The existing system presented itself with inefficient navigation and performance latency issues. Glassbeam required a complete re-design of 'Search', which could be intuitive and scalable to different kinds of log files across domains.

To achieve this, an 'Object Oriented' user experience design framework was used where a 'Central Object' around which the entire application and functionalities could be aligned.

The redesigned application reduced the time to narrow down to problems and troubleshoot issues in minutes instead of days. The overall usability of the system was enhanced and it supported a broader number of use cases. It empowered enterprises with end-to-end log data analytics, support their support, RnD and Sales forecasting activities.

Author Keywords

Big Data Analytics; Machine Logs; Log Attributes; SPL Semiotic Parsing Language; Log Search; Instance Viewer; Visualization; Object Oriented UX; Information Architecture

APCHI '13, September 24 - 27 2013, Bangalore, India
Copyright 2013 ACM 978-1-4503-2253-9/13/09…$15.00.
http://dx.doi.org/10.1145/2525194.2525263

ACM Classification Keywords

HCI; Object Oriented UX; Big Data Analytics

General Terms

Machine Log Analytics, Design

1. INTRODUCTION

Machine Log files form the core of the application as they contain recordings of the entire machine related information. System experts use Machine log files to do predictive analysis of system usage patterns and monitoring system health. Logs are essential to understand the activities of complex systems [1]. Many data centers, OEM[1]s and IT Enterprises use log maintenance services to support their customers incase of any machine related issues.

Glassbeam search is one such SAAS[2] based log maintenance product. It provides SPL (Semiotic Parsing Language) based service for analyzing machine-generated data. SPL translates unstructured data into something, which is meaningful to the human eye and automated methods [2]. Glassbeam cloud database, stores raw and parsed information from multi-structured/unstructured log files. It contains all the details of machines extracted from these sources, as well as the linkages and correlations between the different sections [3] as shown in Fig 1.a.

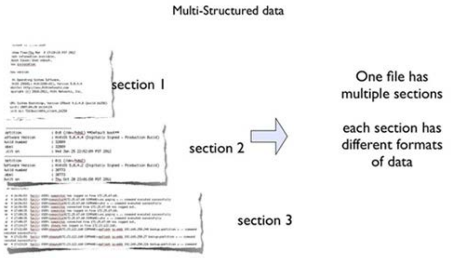

Multi-Structured data
section 1
section 2
section 3
One file has multiple sections
each section has different formats of data

fig.1.a. Multi-structured log data

[1] Original Equipment Manufacturers

[2] Software as a Service

An unstructured log file contains multiple sections and each section contains different formats of data as shown in Fig.1.a. Due to this information complexity, most users find it extremely difficult to search and do an analysis on machines.

In a real world scenario whenever a user is looking for a solution to Support Cases[3] related to a machine, he has multiple options to search for it. e.g. Event timestamp, Error codes, Machine properties etc. And the fact that the system stores millions of historical logs from all the machines makes it very difficult for the users to make quick discovery. Users had to rely upon a combination of log analytics applications, IDE(Integrated Development Environment) and reader applications(e.g. text pad) to complete their key tasks.

Glassbeam required a unified search solution using which users could deep dive into the log files of any machine and do further analysis.

2. APPROACH

A 'Central Object' is the key component in the system around which various interaction patterns can be defined. Typically this component can be used within various information categories to design the system.

'Log Files' were identified as the 'Central Object' as explained above. It was important to understand the mental model of users around log files while solving the Support Cases. To do this, common industry support cases were taken into consideration.

2.1 Support Cases – Attribute Identification

2.1.a Support Case 1- Customer from Tele-communication Domain [5]:

Search for all logs where the "115.115.175.206" is the request source, host name is "10-164-151-16 " (production server). Select "115.115.175.206" from the Apache Access Log IP and "10-164-151-16" from host name facet list and the search query is 'Time to Respond'=1819

To perform this activity user would first look for the search string '*'Time to Respond*'***(Event Attribute)*** for host name '*10-164-151-16*' and do a full text search. However at this point his task doesn't finish. He will then filte for all the source Machine_IDs '*115.115.175.206' **(Machine Attribute)*** in the search result list with a event attribute value '1819' ***(Value Condition)*** for a specified time frame or frequency ***(Time & Frequency)***. Once he has identified the logs, he would perform further activities like downloading the list or check for reasons of that particular event.

2.1.b Support Case 2 – Customer from Data Network Domain [6]:

Search for all queries that use the oneweek_performance_dashboard_data_tmp column and that were executed on 19th June 2013 and that had an execution time of greater than 3 seconds, the search query is 'Binary log Date'=130619 AND 'Binary log Exec Time'>3 AND 'Binary log Query'="UPDATE oneweek_performance_dashboard_data_tmp"*

To perform this activity a user would first search for all the logs having a SQL connection with '*oneweek_performance_dashboard_data_tmp'* (**Event Attribute**) on the 19th of June (**Time & Frequency**). He would then check for *'Binary log Exec Time'>3 AND 'Binary log Query'="UPDATE oneweek_performance_dashboard_data_tmp"* as (**Value Conditions**). Once he has identified all the case he will drill down to the actual program making initiating the event.

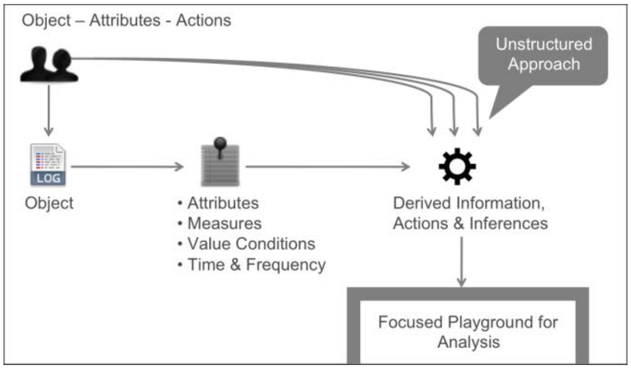

Object – Attributes - Actions
Unstructured Approach
LOG
Object
• Attributes
• Measures
• Value Conditions
• Time & Frequency
Derived Information, Actions & Inferences
Focused Playground for Analysis

Fig 2.1.a. Object Attributes Actions

This approach helped us identify the following **Object Attributes**

a. **Machine/Event attributes** (such as Machine IDs, case ID, Error code etc.)

b. **Measures** (These machine specific attributes like CPU load, portal name, event fields etc.)

c. **Value Conditions** (These are the various grouping and correlation expressions based on the numerical or logical value of any measure)

d. **Time and Frequency** (The point of occurrence or the frequency of any event)

[3] Requests raised by Customers for planning, installation, training, troubleshooting, maintenance, upgrading, and disposal of a product.

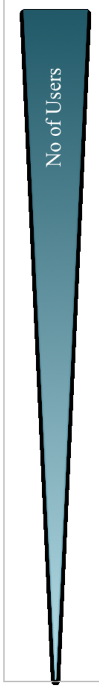
These attributes are correlated and directly picked from the central object i.e. LOG FILES. To substantiate our finding, a model was prepared where the entire machine–related information were classified into the above-mentioned categories.

It was found out that this model holds true for most of the Support Cases (as shown in Table.1).

The model has been explained in detail in sections below.

Based on the usage pattern and Object Attributes the following key features in the system were designed.

- Glassbeam Search to facilitate Log Data Search
- Log Instance viewer to facilitate Machine Discovery & RCA[4]
- Workbench for Report Visualization



User Role	Objectives	Machine Attributes	Event Measures	Event Values	Time & Frequency	Derived Information
L1/L2 Support Engineer	Search for occurrences where error code = "Software asset failure" AND fault description contains "UNKNOWN_opcode"	✓	✓	✓	✓	✓
	Show events with "DSP in recovery mode" in event text of logger files.		✓	✓	✓	✓
	Show events where status is ”Error Code = 26” and !=42.	✓	✓	✓	✓	✓
	Show all occurrences where Buffer Allocation Failure has value > 0 in section .Show data path maintenance counters	✓	✓	✓	✓	✓
L3 Support / R&D	Search for most frequently occurring “Asset Crash” in the last three months. Identify Machines facing this problem.	✓	✓	✓	✓	✓
	What was the threshold “No of Concurrent Users” in Servers at Bangalore facility during the last failure.	✓	✓	✓	✓	✓
Product Manager	What was the total number of Support cases raised across product line. Which product is corrupted and needs to be updated.	✓	✓	✓		✓
	What was the total network downtime in the Development facility of ABC Tech, Mysore.	✓	✓	✓		✓
Sales and Marketing	How many user are using XYZ Suite Version 3.0. He wants to update them about the latest Version i.e. Version 3.1 in the market.	✓	✓			✓
	How many Machines have graphics package G800 active on them. List down the Machine IDs if hardware needs to be updated i.e. Graphics Card < version GX800.	✓	✓	✓		✓

Table.1 Log File Information sets and Use Cases

3. OBJECT ORIENTED UX FRAMEWORK

The ‘Object Oriented’ framework allowed us to map the derived Object Attributes to the Usage Pattern. This informed the Interaction Design model such that Primary information sets as defined via Object Attributes was brought upfront in the UI.

3.1 Key Usage Pattern

From the user research it was identified that whenever users are looking to solve a support problem they search for the source log file and download them. Once they have identified the machine and downloaded the log file, they open the log file using any IDE or text pad to do further analysis. It was identified; the user always required a specific instance of log file opened up for doing a detailed analysis. The current model of user involvement with the log database is shown in Fig 1.b.

[4] Root Cause Analysis

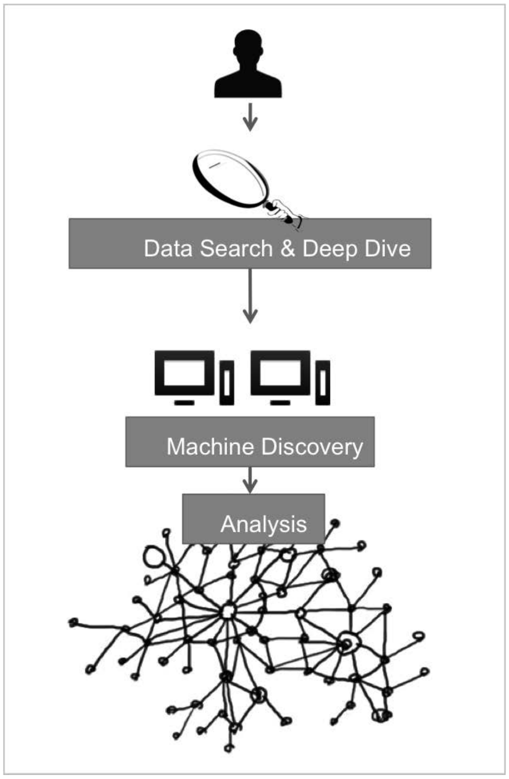
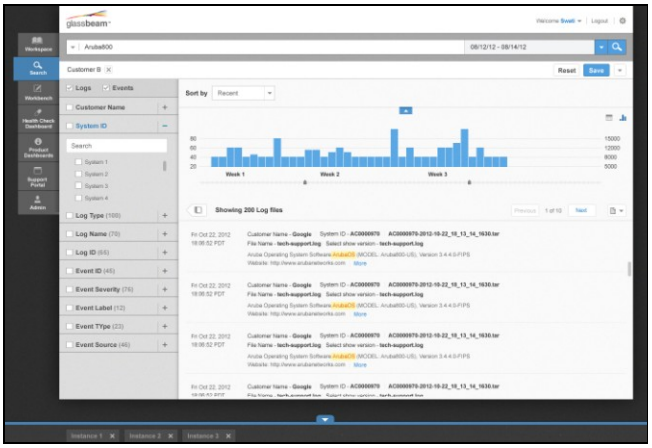


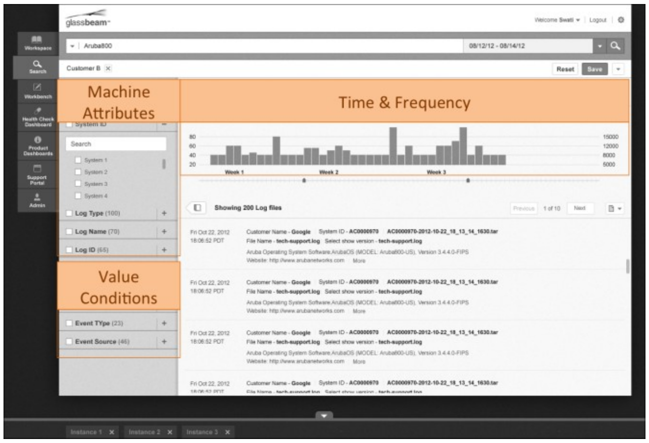
Fig 1.b. Usage pattern with log file

3.1.a Glassbeam Search

All the Object Attributes were provided as the different facet layers a user could apply after he has done a full text search. Based on the attribute measure, the options for filtering the search were provided. [7]



Fri Oct 22, 2012 18:06:52 PDT Customer Name - Google System ID - AC0000970 AC0000970-2012-10-22_18_13_14_1630.tar
File Name - tech-support.log Select show version - tech-support.log



Fri Oct 22, 2012 18:06:52 PDT Customer Name - Google System ID - AC0000970 AC0000970-2012-10-22_18_13_14_1630.tar
File Name - tech-support.log Select show version - tech-support.log

Fig.3.a Glassbeam Search

3.1.b Log Instance Viewer

Once the user finds the exact log file he could open the entire log file from within the application in a log instance viewer. This facilitated him to do multitasking and provided him with a playground to do all his analysis.

He could view related logs based on the key Machine or Event attributes.

He could perform a full text or a regular expression search based on the Value Conditions of Measures.

The system allowed him to compare different instances of log file based on the time attribute. [7]

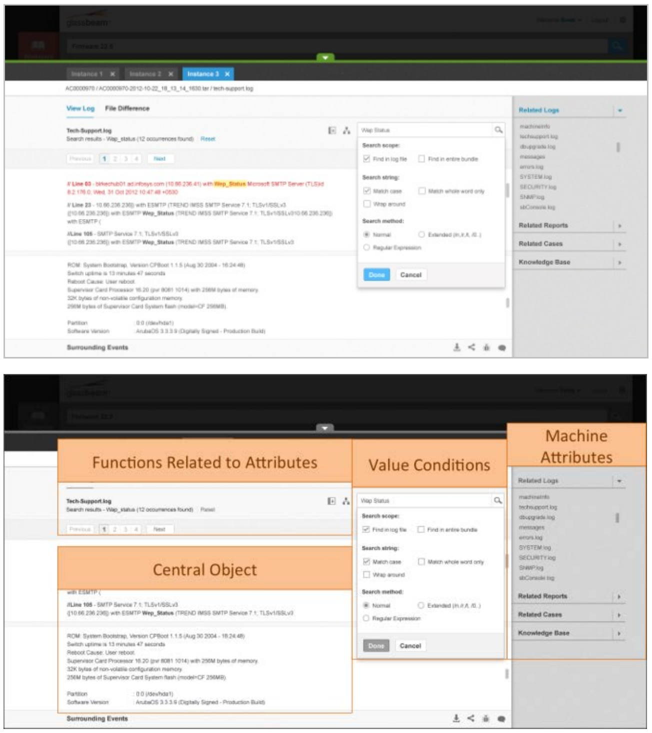


Fig.3.b Glassbeam Log Instance Viewer

3.1.c Glassbeam Workbench

The Object Attributes allowed user to create reports out of history log data. He could now choose the Machine easily and plot visualizations against any Measure. The Value Conditions and Time attributes helped him to further drilldown easily. [8]

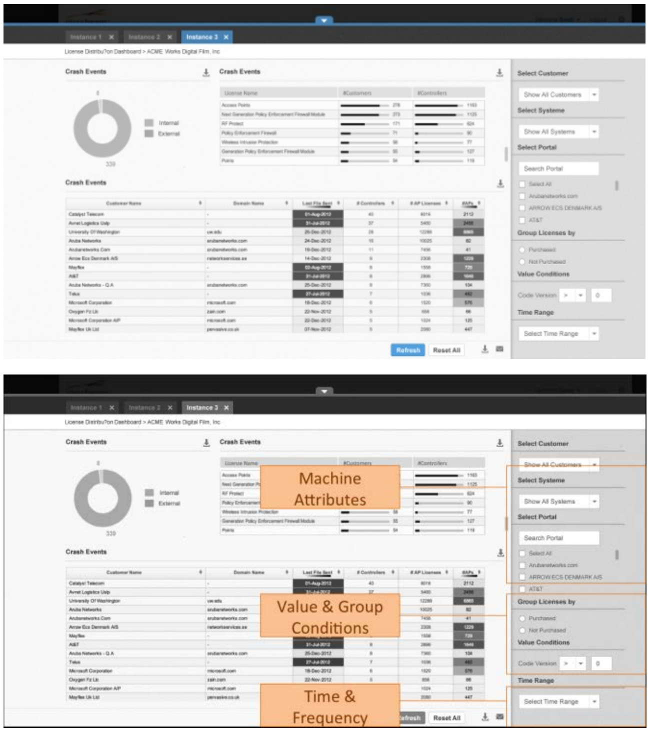


Fig.3.c Glassbeam Workbench

4. CONCLUSION

The redesigned system allowed users to deep dive into actual logs and files from within the application and find relevant sections. They could now easily compare multiple observations of log files and quickly converge on changes. A defined log file instance allowed them to quickly isolate issues by exploring actual data rather than just search. This reduced the dependency on other systems and actual time to support.

The object oriented approach helped in designing a scalable framework, which could work for all types of log files across different domains with varying complexities.

5. REFERENCES

[1] "Computer data logging Wikipedia, the free encyclopedia." 2007. 14 Jun. 2013 <http://en.wikipedia.org/wiki/Computer_data_logging >

[2] "Glassbeam plots course for its SaaS analysis service with ..." 2013. 22 Jun. 2013 <http://www.glassbeam.com/wpcontent/uploads/2013/04/Glassbeam_plots_course_for_its_SaaS_analysis_service_with_uns1.pdf >

[3] "Big data analysis on machine data Glassbeam." 2013. 24 Jun. 2013 <http://www.glassbeam.com/bigdataanalysisonmachinedata/ >

[4] "Object oriented user interface Wikipedia, the free encyclopedia." 2007. 25 Jun. 2013 <http://en.wikipedia.org/wiki/Objectoriented_user_interface >

[5] "Polycom- Home." 2013.26 Jun. 2013 < http://www.polycom.co.in/>

[6] "ArubaNetworks – Enterprise Mobility & Remote Networking Solutions" 2013.26 Jun. 2013 < http://www.arubanetworks.com/>

[7] "Data Sheet – Glassbeam Search." 2013. 27 Jun. 2013 <http://www.glassbeam.com/wpcontent/uploads/2013/02/SEARCHDataSheet.pdf >

[8] "Glassbeam workbench drag and drop analytics on machine log ..." 2013. 30 Jun.2013 <http://www.youtube.com/watch?v=hIuzNBvK_Lg >