Journal of Signal and Information Processing, 2015, 6, 73-78
Published Online May 2015 in SciRes. http://www.scirp.org/journal/jsip
http://dx.doi.org/10.4236/jsip.2015.62007

Scientific Research Publishing

Non-Intrusive Context Aware Transactional Framework to Derive Business Insights on Big Data

Siva Chidambaram[1], P. E. Rubini[2], V. Sellam[2]

[1]Department of Computer Science Engineering, Sri Muthukumaran Institute of Technology, Chennai, India
[2]Department of Computer Science Engineering, SRM University, Chennai, India
Email: sivachidambaram87@gmail.com, iniya29@gmail.com, sellamveera@gmail.com

Received 22 February 2015; accepted 1 April 2015; published 2 April 2015

Abstract

To convert invisible, unstructured and time-sensitive machine data into information for decision making is a challenge. Tools available today handle only structured data. All the transaction data are getting captured without understanding its future relevance and usage. It leads to other big data analytics related issue in storing, archiving, processing, not bringing in relevant business insights to the business user. In this paper, we are proposing a context aware pattern methodology to filter relevant transaction data based on the preference of business.

Keywords

Context Aware, Pattern Recognizer, Big Data

1. Introduction

There are varieties of diversified portfolio of applications getting deployed in the Enterprise Infrastructure space, and each application has a different trend of arrival patterns which generates machine data that need to be captured and processed to gain business insights. The problem starts from collection and filtering and processing of the data becomes difficult due to the rate in which the data getting generated is huge [1]. This requires an efficient way to interpret and bring relevance to the particular context the business deals with. The preferred way to look at this issue is to bring in relevance when the data are getting generated real time so that only the relevance and needed machine data are getting captured by leaving the unwanted machine data.

The purpose of this Context Aware Transactional framework is to categorize the patterns and filter the machine data based on the relevance of each transaction getting performed. Based on the filtered data, the enterprise can

How to cite this paper: Chidambaram, S., Rubini, P.E. and Sellam, V. (2015) Non-Intrusive Context Aware Transactional Framework to Derive Business Insights on Big Data. *Journal of Signal and Information Processing*, **6**, 73-78.
http://dx.doi.org/10.4236/jsip.2015.62007

concentrate on how to effectively bring out the business insights within the application by not spending too much on the cost aspects with respect to the data storage and processing [2].

The objective of this paper is to propose a methodological approach to implement a non-intrusive component which can be plugged into the existing enterprise infrastructure layer to bring out all the insights business wants by capturing only the relevant business oriented machine data [3]. This paper is organized as follows. Section 1 gives an overview of context aware filtering. Section 2 gives the proposed architecture and the multiple phases involved. In Section 3 approaches and the API for context aware filtering are described [4]. Section 4 and Section 5 consist of experimental analysis and the amount of data saved during the data collection gain with respect to storing, archiving, and processing in the context of the component proposed [5].

2. Non-Intrusive Context Aware Transactional Framework

Context aware filtering is the process of recognizing the machine data based on the pattern. The patterns are application specific based on specifics like business rules, database access related, and external interfaces. The methodology is built in such a way that this component can be deployed as non-intrusive into any of the enterprise layer to generate data insights. The high level logical steps involved in context aware filtering are shown in **Figure 1**.

2.1. Pattern Builder

In this phase preserving of existing business and technical knowledge are captured will be utilized and key characteristics of the existing application are captured and stored in the master Meta data in the repository.

2.2. Pattern Recognizer

In this phase pattern matching will be applied on the machine data, and the recognized data are retrieved and stored in the desired repositories. The generation of machine data is in multiple phases, so the pattern recognizer will be a logical independent component which can be made as non-intrusive deployment whenever any transaction happens.

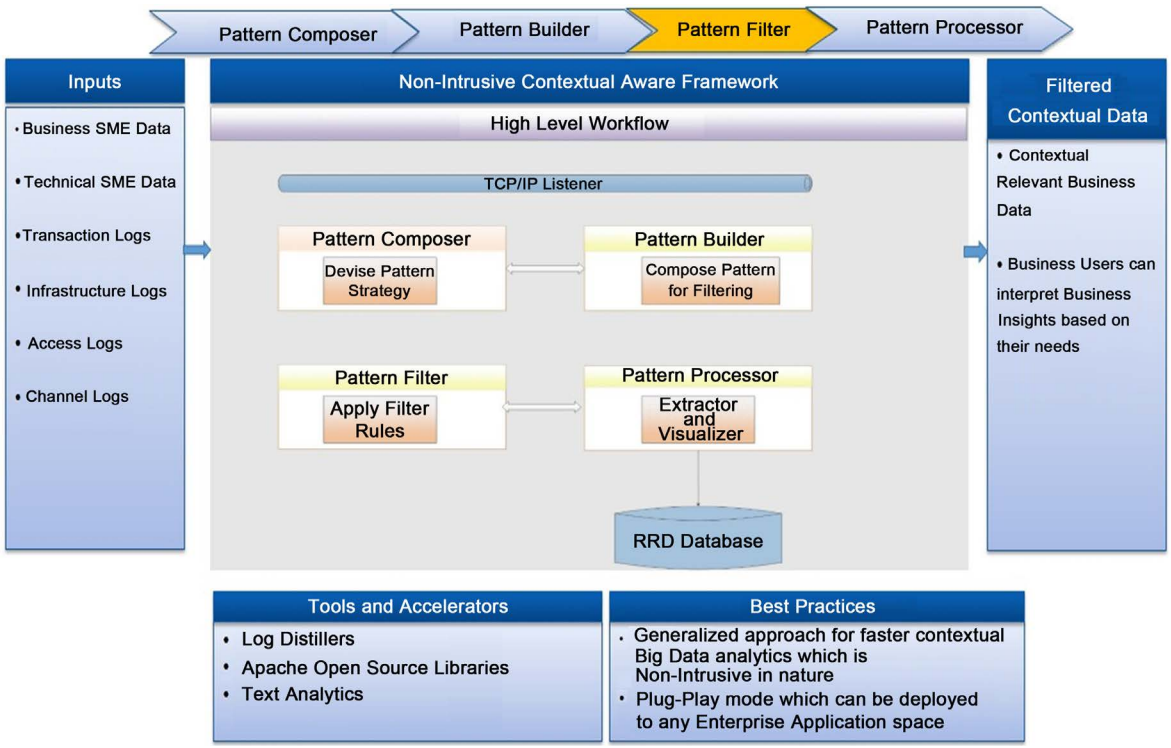

Figure 1. High level logical steps involved in context aware transaction framework.

2.3. Pattern Filter

In this phase the pattern filter gets applied on to the pattern recognized machine data which properly filters the relevant and store it in the database for further processing.

2.4. Pattern Extractor & Visualizer

In this phase the pattern extractor and visualizer helps the enterprise to devise the strategy based on the business rule to extract data.

3. Proposed Architecture Frameworks

The high level proposed architecture is explained in **Figure 2** and the components involved in creating the Context aware filtering are explained in the following sections.

3.1. Channel Listener

This component will act as a listener component to the channels. The channel sends the request based on the request the listener component intelligently forms the triggering point for the Pattern builder to trigger its operations. It acts as a signal sender for the next component to act upon.

3.2. Pattern Composer

This component will also act as intelligent interpreter and filters out the rules present across applications.

3.3. Pattern Builder

This component retrieves the composed pattern from pattern composer and builds a searchable pattern format

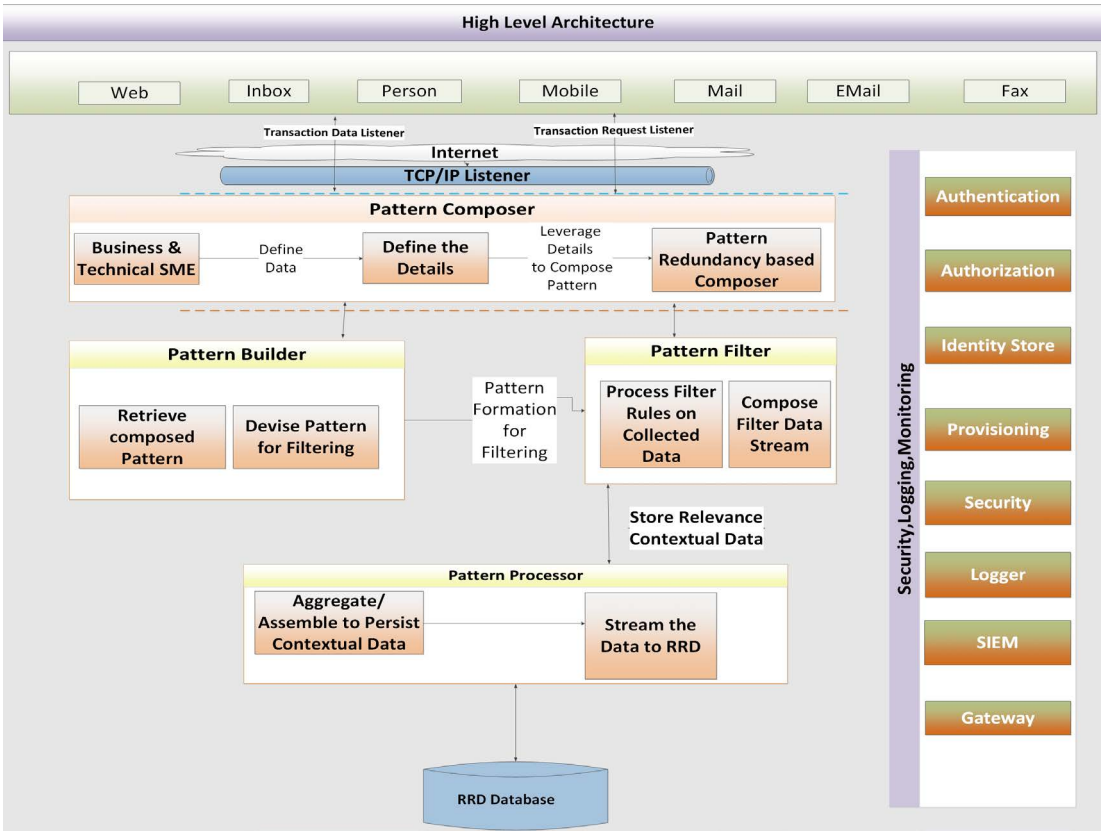

Figure 2. Context aware filtering high level architecture.

which can be directly applied on to the enterprise contextual data getting captured by the Channel Listener component. This component also deals with the intelligent interpretation of the contextual data from the Enterprise with multiple dimensions and variety. The smartness is built into the component itself and different scoring algorithms based implementation is leveraged to achieve the same.

3.4. Pattern Filter

This component applies the filter rules and it has to interact much with the infrastructure component. The filtered data after the appliance of rules will be streamed to Pattern processor component

3.5. Pattern Processor

This component retrieves the filtered contextual data and parses efficiently to aggregate and assemble the data as per the requirements

3.6. Infrastructure Component

This component provides Authorization, Authentication, Logging, Security etc. and it's visible to all the other components in this framework. This leverages most of the open source libraries for its operation.

3.7. Transaction Log Parser

A portion of transaction log file used for the experimentation has been shown in **Figure 3**.

3.8. API Details

The API and the corresponding functions are explained in **Table 1**. These APIs are used in the process of data filtering.

3.9. High Level Logical Details

Properties details and context aware details to filter the relevant data to connect to twitter and also the extraction based on filtering option given in context Data Pattern parameter is defined in **Figure 4**.

4. Report Analysis

The report analysis for visualization of contextual relevant data and also the percentage of savings before and after context aware filtering is shown in **Figure 5** and **Figure 6** respectively.

Figure 3. Sample transaction log file.

```
OAuthConsumerKey=dafjsdfljsadfjsdafjsdafj
OAuthConsumerSecret=532462748234sdhfjsdafhas32486742342
OAuthAccessToken=2342348fsdfhsdafj32472342349234723497234897
OAuthAccessTokenSecret=fsdjfjsda23984283423781jlkfsjfksdjksdajkfsdajkf

contextId=JYF7
contextData=Award,Politics,Sports,Cinema,Email Id
contextDataPattern=Po%ti*
isContextValid=false


filterId=FYJTTTUVR
filterData=Email Id
filterDataPattern=Po%ti*
isfilterValid=false
```

Figure 4. Context data pattern parameter.

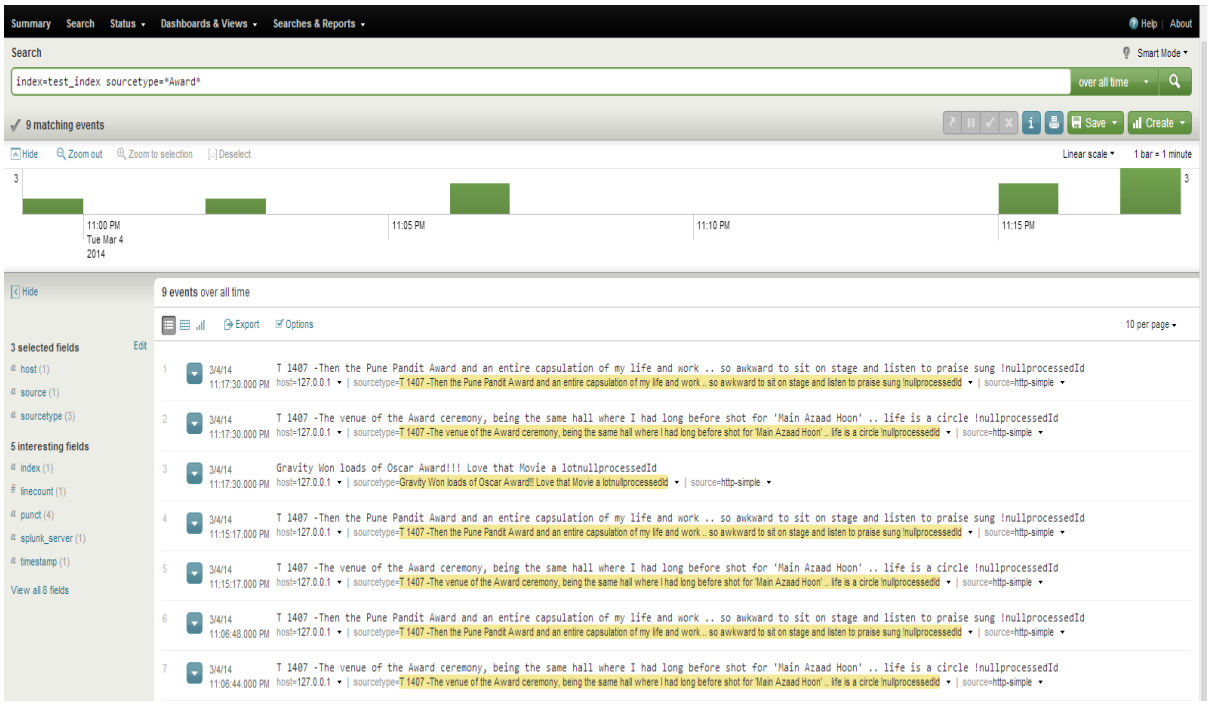

Figure 5. Visualization of contextual relevant data.

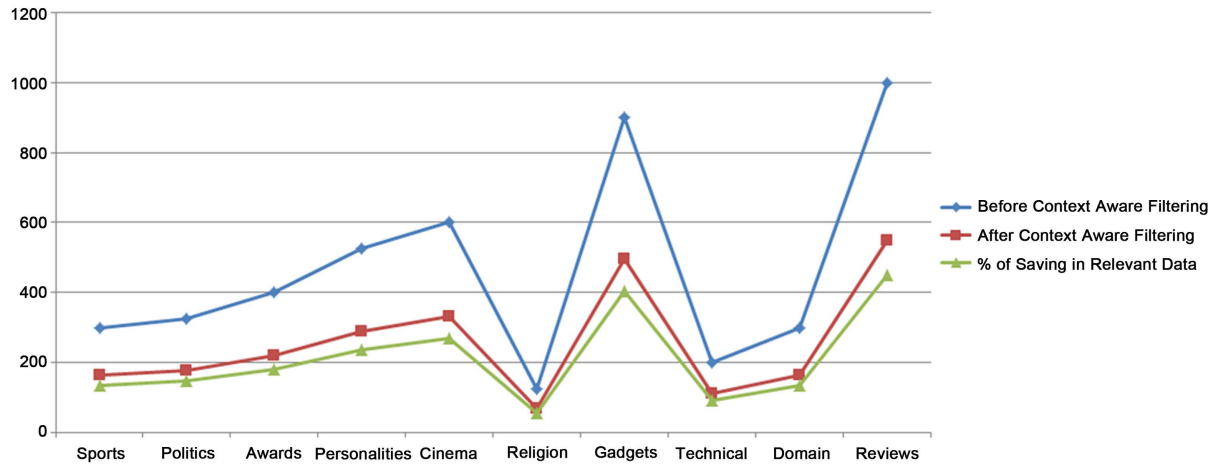

Figure 6. Percentage of savings.

Table 1. API descriptions.

Input: Transaction log file

a) Listen to TCP/IP channels;
b) New pattern repository builder pb;
c) pb. build pattern ();
d) pb. persist patterns ();
e) Data: machine data;
f) Getconnection ();
g) List patterns list: = readall patterns ();
h) New pattern composer pc;
i) List redundant patterns = pc. get redundent patterns (patterns list);
j) New pattern filter pf;
k) List filtered pattern list = pf. filter patterns (patterns list, redundant patterns);
l) Data = pf. apply filter (data, filtered pattern list);
m) Data=aggregate & assemble (data);
n) Persist data (data);
o) Release connection ();
p) New pattern extractor pe;
q) pe. show log report ();

Output: Contextual relevant data consolidated for gaining business insights

5. Conclusion

Most of the analytics application captures all the data for future analytics, but the key aspect is to bring in the context aware filtering on the data getting generated from multiple sources of applications. This eases out the analytics complexity on the enterprise and brings in better prospects towards visualization of data. The complexity and cost factor involved in data management like storing, archiving, backup, recovery, etc. can be reduced by this framework.

References

[1] Duggan, J. and Stonebraker, M. (2014) Incremental Elasticity for Array Databases. *ACM SIGMOD/PODS* (*SIGMOD* 2014).

[2] Kalinin, A., Cetintemel, U. and Zdonik, S. (2014) Interactive Data Exploration Using Semantic Windows ACM. *SIGMOD/PODS* (*SIGMOD* 2014).

[3] Vojnovic, M., Xu, F. and Zhou, J.R. (2012) Sampling Based Range Partition Methods for Big Data Analytics. No. MSR-TR-2012-18.

[4] Sundaram, N., Turmukhametova, A., Satish, N., Mostak, T., Indyk, P., Madden, S. and Dubey, P. (2014) Streaming Similarity Search over One Billion Tweets Using Parallel Locality Sensitive Hashing. *Annual Conference on Very Large Data Bases* 2014 (*VLDB* 2014).

[5] Jun, S.-W., Liu, M. and Kermin Fleming, A. (2014) Scalable Multi-Access Flash Store for Big Data Analytics. 22*nd ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*.