Published in Proc. of the IEEE International Workshop on Management in Cloud Computing (MCC 2011), July 2011, Washington

Cloud-based Event-processing Architecture for Opinion Mining

Stella Gatziu Grivas
Information and Knowledge Management Unit
University of Applied Sciences NW Switzerland
stella.gatziugrivas@fhnw.ch

Marc Schaaf
Information and Knowledge Management Unit
University of Applied Sciences NW Switzerland
marc.schaaf@fhnw.ch

Michael Kaschesky
E-Government Unit
Bern University of Applied Sciences Switzerland
michael.kaschesky@gmail.com

Guillaume Bouchard
Data Mining and Machine Learning Group
Xerox Research Center Europe Grenoble France
guillaume.bouchard@xerox.com

***Abstract*—The viability of cloud computing for information-intensive tasks arising in real-time opinion mining and sentiment analysis of large online text streams is described. We show how a smart distributed architecture enables an efficient and scalable design for opinion mining on internet-based content that answers key challenges, such as integrating heterogeneous data sources and adapting to events through dynamic system configuration. In particular, we present a novel approach of semantic complex event processing in a cloud environment capturing different levels of information, such as event data (e.g. content from various heterogeneous, distributed sources) as well as associations identified during the opinion mining and sentiment analysis process (e.g. dynamic co-reference resolution).**

Keywords: Cloud Services, Operations and Management, Event-driven Architecture, Policy-based Service Management

I. INTRODUCTION

There is an increasing need for decision makers to capture the public opinion about their product or actions, and several systems have been developed to automatically extract and interpret Web content related to a specific concept or topic [15]. Today, opinion extraction software is mostly relying on a small list of potential data sources to support the analysis of a restricted set of topics. The success of these small/moderate-size systems shows the potential feasibility of more global large scale opinion mining systems on heterogeneous data sources that would be able to analyze the opinion of a countries population on some general questions, such as counting the people having money issues, knowing what the population thinks about the latest news events, etc. Some recent studies demonstrated that such systems could potentially replace, or at least complement, traditional polls [21]. However, such systems do not exist today due to several existing challenges:

- The architecture has to be scalable: millions of web sites have to be crawled in real time to detect all the potential opinions expressed online; the software architecture has to be carefully designed to enable a real time text analysis, opinion extractions and data visualization. One of the key challenges is that the natural language processing workflow is highly interconnected due to implicit or explicit user feedbacks. This means that every time a new entry is detected on e.g. a blog, a large amount of text understanding procedures (and in particular cross-document co-references) have to be launched again. A naïve brute-force reset of the databases is naturally infeasible and smarter ways of updating the extracted content have to be found.
- As the set of information sources that has to be monitored changes, the architecture has to be able to adapt dynamically to changing sources and changing event rates from the different sources. This requires the agility of the processing system.
- There is a question of trust: if global opinion mining systems are used for polls, the various lobbying organizations will try to take advantage of this situation by creating artificially more information on the web to bias the results of the systems in the direction that is advantageous for them.

In this paper, we advocate the fact that a smart distributed architecture enables a convincing design of an opinion mining architecture that leads to a principle answer of the previous challenges. In particular, we derive a novel approach of semantic complex event processing in a cloud environment capturing different levels of information like event data (content gathered from various heterogeneous, distributed sources) as well as associations identified during opinion mining and sentiment analysis.

The paper is organized as follows: We first introduce a motivating example that illustrates the need for large scale distributed software architecture. We then discuss existing approaches for event processing and then detail our approach for event processing suitable for opinion mining and sentiment analysis.

II. MOTIVATING EXAMPLE

Our main goal is to design a generic opinion extraction system that can analyze online opinions about many possible

topics. Most of existing opinion tracking system are based on the analysis a given expertise domain. However, for the policy making domain that we focus on, policymakers have many different requests on a wide variety of topics. In this section, we introduce an example scenario where a decision maker is interested in a specific issue (how the JAVA software standard is to be governed, i.e. opensource of proprietary). The objective of the proposed system is to analyze existing opinions related to the issue that have been expressed over online and display a summary of these opinions in a graphical interface.

A. Application scenario JAVA governance

Decision-making on the governance of the global software standard JAVA used to be a closed-book exercise involving the largest players. Small software firms and individual developers had to accept what the JAVA Governing Board decided. But this community with high internet-affinity informed and communicated via online media and forums to address this issue and request changes leading to opensource the JAVA standard. The full study has been reported elsewhere [16] and is here presented to illustrate the benefits of automated opinion mining using a cloud-based event-processing architecture.

Content analysis of online discussions on JAVA governance covers the period 1998 to 2006. 1998 was the year in which opensource application servers entered the JAVA field and August 2006 was when the opensource model was adopted for governing the JAVA standard.

The opensource players include consulting firms Microstate and Lutris and startup JBoss. The incumbent players primarily include Sun (now Oracle), BEA, and IBM. Around the time that Microstate announced opensourcing its application server, BEA introduced its proprietary application server. In May 1998, IBM introduced its proprietary application server while collaborating with opensource group Apache.

The formation of the JAVA application server field thus started in 1998 with both opensource and proprietary solutions shaping the field from the very beginning. By August 2006, all major players endorsed opensource JAVA software. In between, conflict and fierce contestation of competing paradigms characterized online discussions and mobilization. The top-level decisions of firms regarding the adoption of opensource approaches can be directly linked to online discourses.

Figure 1 presents two opposing communication networks on the issue at different points in time. One communication pattern is around the 'proprietary logic' while the other is promoting the 'opensource logic'. At the top, the communication network in 2002 is depicted, showing that JAVA opensource software was an issue that attracted some interest. At the bottom, the same communication network is depicted in 2004, showing a massive increase in interest and engagement.

The analysis of JAVA open source status was done by manually identifying the main players in the online discussion and extracting their opinions expressed in online forums during several years. The comments were manually selected using multiple filters and categorized into a predefined ontology describing the type and the status of the opinion. The total amount of manual annotation work is huge: it required the reading and understanding of several thousands of paragraphs speaking about the same topic. In this paper, we describe an architectural approach that enables the automation of the process thanks to opinion mining techniques and the reuse of labeled sentences in related domains; this goes far beyond the content analysis in the JAVA governance case: 1) by tracking opinion formation in real-time, 2) by dynamically adapting to changing user requests (e.g. different topics or timeframe), and (3) by intelligent selection of services (e.g. input source, NLP components) based on user requests or other detected changes relevant for opinion mining.

III. OPINION MINING AND SENTIMENT ANALYSIS

Opinion mining and sentiment analysis is an active area

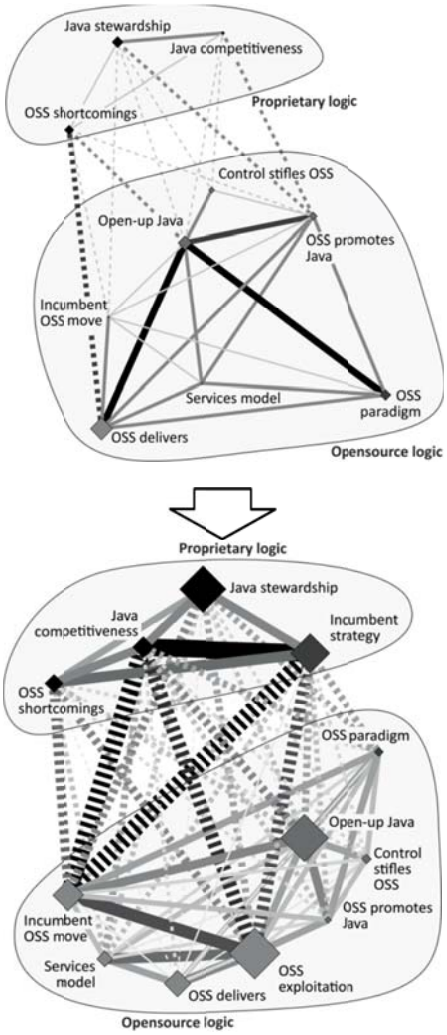

Figure 1: Ongoing detection of opinions and communication network topologies for tracking the evolution of opinions [16]

of research. Often based on a combination of statistical methods (known as 'machine learning') with dedicated background information, such as dictionaries, opinion mining techniques with a good accuracy can be developed relatively quickly by using labeled examples and sentiment words as features: After an initial training phase based on a supervised classification of regression technique, the polarity of the opinion expressed in free texts can be automatically estimated, enabling large scale analyses of opinions [23]. In politics, public opinion is one of the crucial measures of interest classically relying on surveys. With the rapid growth of online media and user-generated content, it has been demonstrated that relatively simple methods can be used to estimate the political orientation of people [20] or of legislative speeches [17]. The existing techniques to extract the opinions from texts in natural language can be very effective on well targeted task on a single domain, but fail to really capture the opinions when the domain when there are multiple topics or when the domain is too broad. There is today a real need to design methods that are robust to domain changes. To solve this issue, one has often to rely on the "crowd-sourcing trick", i.e. enabling several users to benefit from the automatic feedback given by other users. In the following, we discuss the main challenges in opinion mining that one has to face when dealing with very generic texts covering several topics.

A. Topic modeling and Sentiment Analysis

Even if opinions are correctly extracted from texts, they need to be aggregated and summarized to be properly analyzed. Creating single-document summaries of reviews is recognized to be a difficult task [22]. The general approach consists in first clustering the texts by topics and then organizing the texts by the type of sentiment for each topic [26]. The most popular topic identification technique is Latent Dirichlet Allocation [4] and its application to sentiment summarization [7].

Some researchers applied text categorization methods to extract sentiment at the document level. Pang and Lee have classified movie reviews with bag-of-words features and an SVM classifier. Later they adopted a hierarchical approach, using a classifier to first find subjective sentences and a second one to determine their polarity [23]. Other approaches focus on identifying opinions inside sentences at the sub-sentence expression level. Wilson et al. introduced OpinionFinder, which employs several different classifiers to identify subjective sentences, speech acts and direct subjective expressions, opinion sources and opinion polarities [33]. Godbole et al. use a simple rule-based approach, utilizing custom sentiment lexicons, identifying negations and using named-entity and coreference resolution [14]. Breck et al. use conditional random field classifiers to identify direct speech events [5], while Ding et al. use lexicons to extract specific product features from customer reviews to automatically generate opinion summaries [10]. Lun-Wei et al. also generate opinion summaries from news and blog articles in addition to extracting opinion polarity, degree and correlated events [19].

A recent analysis of the political opinions expressed in tweets highlights the potential of text streams as a cheap substitute and supplement for traditional polls based on people interviews, such as Gallup presidential polls [21].

The approach presented in this paper is based on 1) an information retrieval system identifying relevant tweets, 2) an opinion detection algorithm based on counting positive and negative words, 3) a predictive model based on a moving average time series model. Our approach, although based on more advanced NLP techniques, will use a similar method based on 1) identification of the documents related to the topic of interest, 2) opinion detection algorithm and 3) construction of the dynamic model of the opinion diffusion, which can be used for simulation and prediction and is described in the next section.

B. Deep Linguistics and Interlinking Analysis

Previous opinion mining techniques are mostly based on word count and rarely use advanced NLP techniques, such as a syntax analyzer. However, many opinions are expressed in an ambiguous way (a survey is given in [22]). We analyze the various ways of introducing syntactic information in an opinion mining system and make use of deeper linguistic analysis than has typically been used in opinion mining systems. Building on [28] we use lexical semantic information together with a data-driven approach based on natural language processing as input to a Bayesian machine learning method. The classification and analysis of opinions will go beyond traditional sentiment analysis based on positive and negative opinions, to a more functionally based analysis, potentially providing "opinion" types such as explicative, agreement and disagreement. We further specify and investigate methods to incrementally enhance the granularity of opinion capturing and representation to include rhetorical structure and arguments.

In the expression-level approach to sentiment analysis, it is essential to consider the contextual behavior of subjective words and the effect of so-called polarity shifting expressions [9]. The special challenges of analyzing sentiment in messages in social networks [13] will be an important aspect of the project. Co-reference resolution methods will also be used to increase detection coverage of entities for sentiment analysis [30].

In deep interlinking analysis, the focus is on techniques for detecting, extracting and disambiguating entities and topics on the Web and on identifying and visualizing relations between entities and topics. We go beyond the current state-of-art not only by considering propositional content for topic detection, but also by considering time, location, provenance, cross-references (links) etc. To this end, hierarchical and incremental variants of clustering are investigated, which allow the organization of clusters along a number of orthogonal and coherent dimensions, e.g. time, source, and opinions. Moreover, we make use of derived features, such as named entities or detected opinions, and contextual features, such as time, source, and embedded images, to arrive at more meaningful clusters.

C. Transfer Learning and Active Learning of Opinions

Every time the system is used, a new opinion classifier is built based on the data labeled by the user. However, once the analysis has been done, these labeled data are not used anymore. Designing an efficient transfer learning algorithm will enable the system to take advantage of the previously learned classifier to improve the learning curve of a new classifier. Transfer learning and domain adaptation techniques are important for enabling the efficient reuse of data. Multi-task learning methods can be used in this context, especially through a joint learning of the hierarchy of topics and the opinion detection classifiers.

The setting of the current approach is unique since we do not know *a priori* what type of opinion will be relevant. This will depend on the specific application scenario selected by the policymaker, and might change every time he or she decides to use or modify the system parameters. Therefore, the learning phase has to be part of the global architecture of the system. Rather than constructing a black-box that takes as input several content sources and outputs a list of opinions for each extracted text, we will develop a collaborative interactive system where the user (e.g. policymaker or assistant or citizen) is asked to label only a small portion of the extracted sentences in order to help the system to learn to extract correct opinions in a transparent way. Because the system continuously learns from user feedback, an accurate labeling of millions of texts will be obtained in few hours. The use of probabilistic methods and user feedbacks will also help to estimate the reliability of the opinion mining system, as it is done today in modern machine translation systems [29].

IV. Cloud Computing and Event Processing

In opinion mining massive amounts of content is gathered from various heterogeneous, distributed sources, such as online discussion forums, blogs, news sites or social network sites. Each of these sources provides continuously new content which needs to be processed in a reasonable time frame to enable opinion mining and sentiment analysis in near real-time. The vast amount of content that needs to be gathered has to be condensed to reduce the amount of separate information pieces that have to be processed to allow the overall system to perform extended analysis with a reasonable amount of resources. Effective pre-processing of gathered content is thus required to reduce the load on the following processing steps and to store only relevant pieces of information in databases and event histories. In addition, content gathering and processing must be able to scale in case of a general increase of published content, for example, before and after major events such as elections or natural disasters.

A. Background

Cloud Computing is an emerging paradigm as recently shown by its first place in Gartner's top 10 strategic technologies list [12]. It promises that resources like computing capacity and storage or services like databases or messaging systems can rapidly be acquired and released based on the current requirements of the deployed application. The provided infrastructure is capable of scaling upwards and downwards rapidly to adjust to the current requirements. A combination of several well-known technologies like virtualization and concepts like service oriented architecture enables cloud computing. However it also provides a great risk of vendor lock-in [3] because currently most cloud providers have their own specialized services which cloud based applications have to use. With regard to event or message-based communications, examples for such proprietary mechanisms are Amazon simple message queue service or Amazons simple notification service.

An Event Driven Architecture (EDA) is an architecture style for the creation of distributed systems with the aim to create efficient and agile applications [6]. An *event* is the center of the communication and is used to control the components which act autonomous, and are fully decoupled. Only the semantic of the events must be understood by all components. The core principle of event processing is that events are sent via computing middleware to all subscribers who act on it. The other components decide themselves which type of events they will subscribe or on which they will respond.

Event Driven Architectures are tightly linked to Complex Event Processing (CEP). CEP is an emerging enabling technology to apply context-aware knowledge from and against large amounts of event data in near real-time. CEP engines can process multiple streams of event-oriented data to identify meaningful events in real time, based on the business rules framework. They analyze low-level events to discover event patterns (e.g., by aggregating them or combining them) and to act on them [8][18]. The huge amount of events to be processed is thus reduced within several processing stages. Complex Event Processing has already been recognized as a natural extension of SOA under the term SOA2.0 or Advanced SOA (according to Gartner report: Context Delivery Architecture: Putting SOA in Context, 23 October 2007). Event processing is also recognized as an important feature for the cloud, be it just in form of messaging or in more elaborated event- or rule-driven behavior [27]. Complex event processing in a cloud environment has gained increasing attention [18][34].

Building on semantic technologies (e.g. on Action/Event/Process Ontologies like OWL-S, WSMO, SWSL, and PSL for Semantic Web Services) there is now the prospect of combining event processing and semantic technologies into the area of Semantic Complex Event Processing (SCEP). The event processing engines can understand what is happening during the processing of the event streams which is relevant to the events and (process) states. Semantic event processing uses for the signaling of an event the knowledge base (e.g., defined as ontology modeling of the relationships between the different events) and can also derive knowledge which is not explicitly described in the event definition. SCEP-enabled event processing engines will have (1) a description of what is happening in terms of events and event patterns based on situations and process states (like in CEP), as well as (2) a

higher-level plan for the (re)actions and activities they can invoke, which can lead to (monitored) follow-up events.

Current research includes rule-based Event Processing Languages (EPLs) for the Web [2], such as Reaction RuleML, which employs reaction rules that have evolved from existing rule-based technologies such as Production Rules and Event-Condition-Action (ECA) rules. They use SCEP-generated facts and knowledge to derive further decisions and trigger automated reactions. Moreover, such EPLs can exploit the declarative expressive power of semantic rules as a means to specify knowledge in a way that is understood by 'the business' and is executable by CEP rule engines. First ideas for semantic event processing [11][1] focus on one common super event ontology which generalizes and ignores the differences of the various application domains. There is a lack of fine grained formal event models which capture the semantics of the specific application domains and use this knowledge to derive further decisions and trigger reactions.

Current domain-specific approaches have developed application and task ontologies for stock applications based on modular subontologies [32]. Another domain-specific approach aims at the development of a framework for different levels of description from event log data, through individual trajectories of activity and interactions between them in a dynamic network of associations that are created by and further shape these interactions [31].

B. Proposed Approach for Event Processing

For the event processing, we adopt the well-proven and clearly defined semantics of event-condition-action (ECA) rules as known from Active Database Management Systems (Active DBMS) and extend them to cloud computing. Moreover, the architectural design is going to be based on proven principles and patterns from service oriented architectures (SOA) which provide an abstraction from the heterogeneity of the underlying communication technologies and infrastructure. This allows different event processing components to work independently of a cloud provider's specific communication technologies and enables semantic interoperability.

As a first step, we propose a new architectural concept with the integration of the architectural approaches SOA and EDA with CEP and the well-defined semantics of active database systems in a cloud environment. In a next step we extend our architectural concept with the principles of semantic technologies. This enables the creation of dynamic processes that can adapt themselves based on derived knowledge to react to changes by adaption of the processing system itself. For this purpose we will integrate access to a central ontology within the runtime containers to allow the deployed components to access the ontologies and related model information in a unified way across the whole processing system.

To provide the required level of abstraction form the underlying technologies and architectural principles, we introduce a runtime container that allows the access to the underlying systems in a unified manner. The container will provide the means necessary for the event based communication with other components and the service based access to central information sources (e.g. the domain specific model / the data repository for foreign data sources). Event processing components can be deployed within this container.

V. System Architecture and Run-Time Platform

In the application scenarios described above, massive amounts of content is gathered from various heterogeneous, distributed sources, such as online discussion forums, blogs, news sites or social network sites. Each of these sources provides continuously new content which needs to be processed in a reasonable time frame to enable opinion mining and sentiment analysis in near real-time. The vast amount of content that needs to be gathered has to be condensed to reduce the amount of separate information pieces that have to be processed to allow the overall system to perform extended analysis with a reasonable amount of resources. Effective pre-processing of gathered content is thus required to relieve the following processing steps and to store only relevant pieces of information in databases and event histories. In addition, content gathering and processing must be able to scale in case of a general increase of published content, for example, before and after major events such as elections or natural disasters.

Furthermore, the most basic requirement is collection of raw data and its suitable modeling and storage in a data repository. Due to the diversity of the sources, the frequent updates, and the volatility of the incoming data size, this is a challenging task. Data from blogs, web sites and forums have to be collected in near real-time and they have to be readily available for opinion mining and sentiment analysis. To this end several Extract-Transform-Load (ETL) techniques and tools must be employed.

A very important issue that arises from the nature of the data, and has not been dealt before in the context of ETL, is the preservation of the privacy of the sources. Whereas most of the data sources are public, there are sources where the identity of the users must be protected. Decision makers and other end-users have their own private sources, such as data from transactions (e.g. tax related data) or data from questionnaires and interviews. The fusion of such private data with public data of the same person adds value to opinion mining and sentiment analysis but must provide privacy guarantees to individuals associated with the data.

We propose a platform for a staged processing approach. This platform will provide the required environment for components of opinion mining and sentiment analysis. It will further provide the means for the communication between these components and to access central model data. Thus it provides the integration with the other components in a loosely coupled fashion.

A. Architecture, standards, component specifications and event processing

To provide the infrastructure for processing massive amounts of content, a staged approach is suited. The first processing stage involves gathering and pre-screening raw content and is located in close proximity to the actual content

sources and possibly distributed across geographic regions (e.g. in university IT centers). The second processing stage is located in dedicated data centers and works with aggregated and filtered content that is processed for opinion mining and sentiment analysis. Such a dynamic staged infrastructure reflects the principles of cloud computing. The first stage of the processing system is thus located in public clouds where a rapid scaling based on current load is possible. The second stage is located in private clouds where privacy is ensured and access monitored while still enabling the sharing of resources to adapt to changing loads.

Because of the dynamic nature of the internet, content collection must dynamically adapt to take the most current and relevant content sources around a specific product or issue into account. The processing infrastructure allows adding or removing content sources by distributing the required configuration information to the associated components and by deploying additional components that handle new input sources. Components will have to monitor different types of data sources. For some sources a subscription mechanism will be possible but for many other sources a specialized polling and extraction mechanism is required that needs to be integrated within the processing system to provide the information in the correct type.

For the efficient processing of massive amounts of content, each gathered piece of content constitutes an event that will be processed via an event processing network to correlate the new content with other relevant pieces of information. This pre-processing of content reduces the overall amount to be processed by filtering out unusable or duplicate events and by deducting information from the events to create so-called complex events representing the interlinking of several related raw events. This will reduce the amount of events that need to be processed with each processing step along the way. In addition, events are semantically annotated during content collection and processing. In other words, the event monitors collecting content from the different sources semantically annotate the events. The rules that define how sources are annotated in which way are specified along the configuration of the particular source.

The event processing system as a whole is distributed across the borders of a single cloud. In the first stage of event processing, agents that handle the massive amounts of raw events are deployed in multiple public clouds to realize the initial filtering and the basic pre-processing (Figure 2). Once the first stage is accomplished these public clouds handover the reduced stream of events to a second processing system that might also be located in a cloud environment.

Throughout the processing stages, the rules controlling the processing draw on past results from processing obtained through earlier runs. A distributed database provides access to the processing history which typically consists of static or seldom changing data. This database provides read only access for the processing nodes to allow the distribution and an update process that can be fed with results of all the processing stages to extend the database with the current processing results. This data store will be available in parallel to the other processing steps to allow the rules to access additional (static) data. Thus the data store is also be used for the integration of foreign data sources by using ETL processes.

B. Distributed rule and event model and event routing

To ease the management of the distributed processing, concepts will be created for the central configuration of the different event processing agents which includes the management of their rule bases, the definition of event sources and the assembly of the different agents to form the required processing structure. The created platform will provide access to all those pieces of information that are needed by the different processing components. Thus, mechanisms to distribute the required model parts to the desired parts of the processing network must be specified and integrated into the platform.

For the distribution of the rules an approach where the rules are specified on a general level, without being assigned to the actual stages and processing nodes is favorable, because the specification of the exact rules for each of the processing nodes would result in a management effort that would cripple the possibility of the system to react to changes rapidly. Thus created framework is required for defining how a general specification of rules from the pre-processing as well as for the semantic processing of the events is used to derive the concrete rules for each of the processing nodes and to specify the required communication channels between those nodes. A similar mechanism needs to be found for the specification of the event sources and the assignment of the corresponding semantic parameters form the currently used ontology.

The linking of the different processing nodes will be based on the information in the central domain specific model and is responsible for the routing of the events between the processing components. Furthermore information on the input sources will be available through the model together with information on the output side where the derived intelligence needs to be distributed to the consumer components such as opinion mining, sentiment analysis, visualization and simulation.

The visualization and simulation components may also

request changes in pre-processing that are triggered by end-user modifications, for example, if a new application situation or new scenario becomes the focus of investigation. Thus, the system infrastructure needs to provide a two way communication mechanism to 1) connect the components to the dynamic part of the central domain model where they can apply their change requests and 2) to obtain the required information from the processing network.

Furthermore the ontologies on which event annotations and event processing are based have to be distributed across the different nodes in a dynamic manner to accommodate for changing requirements. Together with the ontology model, the rule base also changes rapidly, not only due to the changing requirements but also because of self-adaptation mechanisms that are triggered by the simulation and visualization components. Thus the distribution of the models and rule bases must be integrated in the provided platform to allow the processing components to access the model and rule base in a unified way.

C. Privacy Preserving Content Retrieval and Storage

The basic goal is the development of anonymization algorithms that will allow the data fusion from public and private sources without endangering the privacy of the users. The key idea is that by employing anonymization methods, the privacy of users is preserved and no third party will be able to infer new information out of existing public information.

The data privately collected from users is usually collected under privacy requirements constraints. An end-user may exploit such data only to the extent that the personal identities of the users remain hidden.

The dangers to privacy come mainly from two factors: a) internal leakage of personal data in the end-user organization; and b) the identification of users based on statistical aggregated data.

A naïve anonymization method is the removal of direct identifiers like the name of a person or the Social Security Number. Such naïve methods have been shown to be vulnerable to linking attacks where third parties exploit available public knowledge, like age and zip code of an individual, to identify the real person associated with the naively anonymized information. The public knowledge that can be used to identify a person in data where the direct identifiers have been removed, is termed quasi identifier (e.g. age and zip code). It is therefore necessary to develop anonymization methods that will protect against linking attacks by suitably transforming the quasi identifiers. There are three basic challenges associated with this transformation:

- information from various data sources are gathered, so linking attacks must take into account complicated combinations of gathered data.
- opinion mining and sentiment analysis requires adequate accuracy, so data anonymization must keep the transformations of the quasi identifiers to a minimum.
- the almost continuous inflow of data requires that the anonymization procedure must be lightweight and fast to avoid creating a bottleneck in loading of the repository.

VI. CONCLUSION

The presented concept provides a highly dynamic, scalable architecture that allows the processing of vast amounts of information from various quickly changing sources. Furthermore it provides an active, event based, approach for the opinion mining and sentiment analysis that has the capability to provide the results faster than ETL-based approaches to data gathering and processing.

We follow the architectural concept of Event Driven Architectures so that components communicate to each other with events which are sent via computing middleware to all subscribers who act on it. The other components decide themselves which type of events they will subscribe for or on which they will respond. Based on the principles of Service Oriented Architectures we provide an abstraction from the heterogeneity of the underlying communication technologies and infrastructure [15]. We support Complex Event Processing for analyzing low-level events to discover event patterns (e.g., by aggregating them or combining them in a specific context: time, space, semantics) and to act on them [25]. The huge amount of events to be processed is thus reduced within several processing stages.

In a next step we extend our architectural concept with the principles of semantic technologies like in [24]. This enables the creation of dynamic processes that can adapt themselves based on derived knowledge to react to changes by adaption of the processing system itself. The event processing engines can understand what is happening during the processing of the event streams which is relevant to the events and (process) states. For this purpose we will integrate the access to a central ontology within the runtime containers to allow the deployed components to access the ontologies and related model information in a unified way across the whole processing system. By this we provide the necessary information to all processing nodes that is required for the reasoning processes.

To provide the required level of abstraction form the underlying technologies and architectural principles, we introduce a runtime container that allows the access to the underlying systems in a unified manner. The container will provide the means necessary for the *event based communication with other components* and the *service based access to central information sources*. Furthermore the runtime container provides a strong encapsulation of the components with clear interfaces to the other communication parties. Due to this and due to the abstraction from the underlying infrastructure, the event processing components can be deployed dynamically within those containers where the containers themselves can are distributed across several locations like for example different clouds or datacenters with heterogeneous use of the appropriate communication technologies.

We propose the described container concept as a platform for a staged processing approach. This platform provides the required environment for components of opinion mining and sentiment analysis. It can further provide the means for the communication between these components and to access central model data, thus enabling the

integration with the other components in a loosely coupled fashion.

A possible organization of the components for the opinion mining on a large scale could be as follows (Figure 1): The first processing stage involves gathering and pre-screening raw content and is located in close proximity to the actual content sources and possibly distributed across geographic regions. The second processing stage is located in dedicated data centers and works with aggregated and filtered content that is processed for opinion mining and sentiment analysis. The third stage of the system can be located in a private (secure) datacenter and does the final processing and thus produces critical results that have to be secured against unauthorized access. The results from this stage can e.g. be feed into simulation or visualization engines.

Such a dynamic staged infrastructure reflects the principles of cloud computing. The first stage of the processing system is located in public clouds where a rapid scaling based on current load is possible. The later stages are located in private clouds where privacy is ensured and access monitored while still enabling the sharing of resources to adapt to changing loads.

REFERENCES

[1] J. Aasman. Unication of geospatial reasoning, temporal logic, & social network analysis in event-based systems. Proc. of the 2nd Intl. Conf. on Distributed event-based systems, pages 139{145, New York, NY, USA, 2008.

[2] D. Anicic, P. Fodor, S. Rudolph, R. Stühmer, N. Stojanovic, R. Studer. A Rule-Based Language for Complex Event Processing and Reasoning. Proc. of the 4th Intl. Conf. on Web reasoning and rule systems, 2010.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. A view of cloud computing. Commun. ACM 53, 4, April 2010.

[4] David M. Blei, et al. Latent Dirichlet allocation. Journal of Machine Learning Research 3: 993-1022, 2003.

[5] E. Breck, Y. Choi, C. Cardie. Identifying expressions of opinion in context. Intl Conf. on Artifical intelligence, 2007.

[6] R. Bruns, J. Dunkel, Event-Driven Architecture - Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse, Springer Verlag, 2010.

[7] Y. L. Chang and J. T. Chien. Latent Dirichlet learning for document summarization. Proc. of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, 2009.

[8] S.-K. Chen, H. Chang. Complex Event Processing using Simple Rule-based Event Correlation Engines for Business Performance Management. Proc. of the 8th IEEE Intl. Confe. on E-Commerce Technology on Enterprise Computing, ECommerce, and E-Services. 2006.

[9] Y. Choi and C.Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. Proc. of the Conf. on Empirical Methods in Natural Language Processing: 793-801, 2008.

[10] X. Ding, B. Liu, P. S. Yu. A holistic lexicon-based approach to opinion mining. Proc. of the 1st ACM Intl. Conf. on Web Search and Data Mining, Feb 11-12, 2008, Stanford University, Stanford, California, USA.

[11] O. Etzion. Semantic approach to event processing. Proc. of the Inaugural Intl. Conf. on Distributed event-based systems, pages 139{139, New York, NY, USA, 2007. ACM (DEBS 07).

[12] Gartner Inc. , Gartner Identifies the Top 10 Strategic Technologies for 2011, Oct 2010.

[13] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. Processing: 1-6, 2008.

[14] N. Godbole , M. Srinivasaiah , S. Skiena. Large-scale sentiment analysis for news and blogs. Proc. of ICWSM, Boulder, Colorado, USA, 2007.

[15] P. Goyal, R. Mikkilineni. Policy-based event-driven services-oriented architecture for cloud services operation & management. IEEE 2009 Intl. Conf. on Cloud Computing. Bangalore, India, September 2009.

[16] Kaschesky, M. and R. Riedl. Tracing opinion-formation on political issues on the internet. Proc. of the Hawaii Intl. Conf. on System Sciences, 2011.

[17] M. Laver, K. Benoit, and J. Garry. Extracting policy positions from political texts using words as data. American Political Science Review, 97(2): 311-331, 2003.

[18] D. C. Luckham, The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Boston, MA, USA: Addison-Wesley, 2001.

[19] K. Lun-Wei, L. Yu-Ting and C. Hsin-Hsi. Opinion extraction, summarization and tracking in news and blog corpora. Proc. of AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs, 2006.

[20] T. Mullen and R. Malouf. A preliminary investigation into sentiment analysis of informal political discourse.. AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW): 159–162, 2006.

[21] B. O'Connor, R. Balasubramanyan, B. R. Routledge, N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. Intl. AAAI Conf. on Weblogs and Social Media, Washington, DC, May 2010.

[22] B. Pang and L. Lee. Opinion mining and sentiment analysis. Found. Trends Inf. Retr. 2, 1-2: 1-135, 2008.

[23] B. Pang and L. Lee. Thumbs up? Sentiment classification using machine learning. Proc. of EMNL, 2002.

[24] A. Paschke. A Semantic Design Pattern Language for Complex Event Processing. Proc. of AAAI, 2009.

[25] L. Perrochon, E.J.S. Kasriel, C. Luckham, Enlisting Event Patterns for Cyber Battlefield Awareness, 2006.

[26] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. Computational Linguistics, 28(4): 399-408, 2002.

[27] M. Schaaf, A. Koschel, S. Gatziu Grivas, I. Astrova. An Active DBMS Style Activity Service for the Cloud Environments, Proc. of the 1st Intl Conf. on Cloud Computing, GRIDs, and Virtualization November 2010, Lisbon (IARIA 2010),.

[28] H. Saggion and A. Funk. Extracting Opinions and Facts for Business Intelligence.. RNTI, E-17:119--146. 2009.

[29] L. Specia, m. Turchi, N. Cancedda, M. Dymetman and N. Cristianini. Estimating the sentence-level quality of machine translation systems. Proc. of the 4th Intl. Workshop on Statistical Machine Translation, Athens, Greece, 30-31 March, 2009.

[30] V. Stoyanov and C. Cardie. Partially supervised coreference resolution for opinion summarization through structured rule learning. Proc. of Conf. on Empirical Methods in Natural Language Processing, 2006.

[31] D. Suthers. Interaction, Mediation, and Ties: An Analytic Hierarchy for Socio-Technical Systems. Proc. of the 44th Hawaii Intl. Conf. on System Sciences, 2011.

[32] K. Teymourian, A. Paschke. Towards Semantic Event Processing. In DEBS'09, July 6-9, Nashville, TN, USA.

[33] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, S. Patwardhan. OpinionFinder: A system for subjectivity analysis. Proceedings of HLT/EMNLP, 2005.

[34] G. Wishnie and H. Saiedian. A complex event routing infrastructure for distributed systems. IEEE Computer Society, Vol. 2, pp. 92–95, 2009.

Published in Proc. of the IEEE International Workshop on Management in Cloud Computing (MCC 2011), July 2011, Washington