Knowledge-Based Semantification of Business Communications in ERP Environments

Marios Meimaris
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
m.meimaris@medialab.ntua.gr

Michalis Vafopoulos
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
vaf@aegean.gr

Abstract— **The complexity of information in business environments is increasing in rates that are difficult for companies to handle using traditional solutions. The diversity of information and communications that stem from remote and heterogeneous sources creates the need for information and knowledge management in multi-dimensional contexts and platforms, such as web sources, mobile devices and closed databases. As web technologies evolve, the volume, temporality and heterogeneity of available data exhibits a degree of dynamicity that is beyond manual control. Without these limitations, the aggregation of this information can provide new value-driving layers in business environments. Moreover, when this data is semantically structured, knowledge sharing and rule-based management allow for new ways of controlling the flow of information, both internally and externally. We propose, design and implement a set of semantic components extending ERP systems in order to assist semantic web interoperability, to provide a basis for intelligent knowledge management and to unify communication level platforms under shared sets of principles.**

Index Terms—Semantic Web, Linked Open Data, Business Intelligence, Knowledge Management, Ontology

I. Introduction

Modern technologies and design approaches make the web an exceptionally dynamic environment, where data and information is created, consumed, edited and deleted within very small periods of time. This information can be in the form of communication between people and organizations, as in the example of social networking, or in the form of electronic publishing of facts and otherwise static data with varied time scopes. The dynamicity of these types of information makes it necessary for people and systems to be capable of being constantly up to date and ready to respond to challenges, problems and difficulties that result from change.

In the web business world, when it comes to responsiveness such aspects are crucial. Within business contexts, these types of information form a level that will be called the *Business Communications Level*. They can stem from both internal (e.g. databases, employee mobile phones etc.) and external (e.g. social media, web repositories etc.) sources. In the traditional sense, such data is usually neither reusable nor easily retrievable between heterogeneous and remote sources. Data is not always meant for sharing, thus existing within a limited circle of impact. It is even more difficult to take advantage of such data sources in cases where response times have to be minimal, as is the case in the business world.

Aggregating these types of data and information and combining them through a unified schematic structure will become essential for business entities in order to survive and adapt to the required responsiveness. For this reason, it is within our belief that providing platforms with the ability to support interoperability through interchanging diverse data is a step toward an evolved business environment. We have designed and implemented a set of components that follow knowledge-based, semantic web and open data directives, as well as facilitate the extraction and reuse of mobile phone data under a shared ontological substrate, as an extension to traditional Enterprise Resource Planning (ERP) systems. This paper is organized as follows: section (II) will provide the background, section (III) will describe the conceptual modelling, section (IV) will describe the system implementation and section (V) will present the results and a brief discussion of this work's contribution.

II. Background

A. Knowledge Management and Business Intelligence

Generally, Knowledge Management (KM) can be considered as a loosely defined set of methodologies and techniques that aim in externalizing what is considered as internal, or tacit knowledge. Tacit knowledge, as opposed to explicit knowledge, is the type of knowledge that cannot be easily outlined and defined, as it resides within individuals in the form of skill sets, experiences and so on.

A particular subset of Knowledge Management, as is suggested in [1] is *Business Intelligence* (BI), which is considered to be a knowledge level that provides analytical tools and settings for supporting decision-making. Moreover, with the incorporation of new technologies, classical BI functionality is extended to include context identification and socially-aware components, this way taking advantage of the richness of information that stems from social web environments.

In the business world, information is increasingly dynamic. Information flows are constantly changing form and direction, as new sources are created everyday in diverse environments and platforms. Within corporate contexts, these can be found both internally and externally. Internal sources include databases, company knowledge (both tacit and explicit), mobile phone data and any other types of information that are directly controlled by the company. On the other hand, external sources include web 2.0 (social media, blogosphere etc.), semantic web services, open data repositories, the linked open data (LOD) cloud and so on.

B. Internal Sources

1) Databases and ERP Systems

It is usual for companies to hold their data in local databases. More often than not, these are traditional relational databases that use schemata designed and created in company-specific manners. In many cases, they are incorporated in Enterprise Resource Planning (ERP) or similar systems. ERP systems provide a framework for managing business-related resources and assets (both human and non-human) with respect to projects, processes and specific tasks that need to be accomplished. This way, interdepartmental cooperation and coordination can be achieved.

The main benefits of ERP systems are the automation, or semi-automation, of these procedures, as well as the provision of assistance in decision-making tasks. Successful deployment of ERP systems reduces overall operating costs, facilitates better data and information management within the boundaries of an organization and promotes data sharing between organizational sub-parts, and at the same time limits data redundancy [2].

The major drawbacks of ERP systems have to do with openness of data that a particular software package provides, limitations over data sharing and an overall strong coupling between data and software design. Smaizys & Vasilecas [3] argue that the absence of widely accepted standards and formalisms on business rules is what leads ERP system designers to time-consuming solutions which can often be both misleading and discouraging. It is therefore necessary for this data to be structured in ways that allow for sharing and combining with external data under shared and loosely coupled principles.

2) Mobile Devices

The increasing usage of smartphones and their diverse functionalities have led companies to incorporate them into their operation management models. Graf and Tellian [4] talk about the benefits of using smartphones as trackers in Supply Chain Management, achieved with the use of 3G connections and GPS signals. Furthermore, the app market has led companies to develop applications for managing human resources. Modern smartphone capabilities have two basic advantages over the past:

- They permit context-aware application development. For instance, they include metadata such as geospatial and chronological usage data.
- In the case of open operating systems, data extraction is feasible. Therefore, such devices can be programmed to gather and extract data at will. The gathered data have quantitative and qualitative effects, and because of the “on-the-move” nature of mobile devices, they make context easily identifiable and tractable.

As far as internal business contexts are concerned, we are particularly interested in mobile data that falls within the user's communications (texts, calls etc.). To the best of our knowledge, little to no work has been published that assess the use of such data. According to our hypothesis this aggregation leads to a level of information, called the *Communication Level*, which makes possible the derivation of more complex information regarding the user and the developing communicational contexts that he/she participates in. With this approach we build *communication profiles* of users and, when applied to an interconnected network of communication initiators and recipients, we retrieve and assess the patterns that arise.

C. External Sources

External data can generally be considered as dynamic or static. Dynamic data originates from social media and otherwise dynamic sources, such as Google+, Twitter and Facebook, blogs, web 2.0 platforms in general and so on, whereas static data includes unchanging (or slowly changing) information, such as business registries and other repositories. Both are powerful sources of information, but must be handled differently.

1) Social Media

We consider social data to be of high business value, because, if interpreted correctly, they hold key information and knowledge about public sentiment. For instance, this is of particular interest in the case of e-commerce, where products and services are openly offered, marketed, promoted and discussed within social environments.

Social media provide added value in many ways. Two main value drivers associated with them are *social media marketing* and *sentiment mining*. Social media marketing refers to the practice of gaining popularity by means of social media. This can be achieved through advertising via social networks, creating special interest groups about products and services and so on, making it easier for these marketing methods to become partially autonomous and crowdsourced, in the sense that the people joining such groups are usually providers of positive feedback. In this sense, social media marketing strategies are partly self-sustaining. Sentiment mining refers to extracting and analyzing public thoughts and opinions. Recent approaches in web-based sentiment mining have shown that public opinion can be successfully extracted, having useful results in e-commerce [5]. Sentiment analysis can provide valuable insight on the public opinion over a resource (or set of resources), as well as open the way to automated reputation handling methods, both important issues for a business to survive in a web-based environment.

2) (Linked) Open Data

Open Data refers to the notion of openness (i.e. free public access) in certain kinds of data, which can range from crime reports to government decisions and from bus schedules to registry information about corporate entities. Open datasets are often designed to be machine-understandable and for this reason it is common for them to be shared through document formats such as JSON, RDF and XML. Usually, the existence or absence of a conceptual schema for the underlying structure lies in the discretion of the data designers, but it is reasonable that at least some form of schematic structure is present. In order to be more descriptive about the represented resources, ontologies can be used. Open datasets are being published in increasing volumes and often ontologies are being designed and published along with them. This way, not only is it easier to query the data in more intelligent ways, but it allows room for remote searching and querying, as well as reasoning and any kind of inferencing in general.

A particular case of data openness is that of *Linked Open Data*. The main difference lies in the usage of a semantic web standard called *Uniform Resource Identifier* (URI) to refer to entities, and the interlinking of different URIs via ontological properties. The general idea of the semantic web is to be able to describe entities without being committed by their inherent structure. Proper usage of URIs ensues that resources (instead of data entries) are uniquely identified and referenced. Therefore, linking different resources by their URIs creates a network of interconnected entities that are described by sets of metadata. This is, in fact, the essence of the semantic web vision, as resources can be described and referenced unambiguously throughout different and heterogeneous platforms.

Within business contexts, such as e-commerce or B2B collaboration, the shift to semantic web technologies and (linked) open data paradigms during the last few years is guided by the provision of unique descriptions of corporate entities as reference (e.g. OpenCorporates.com), publishing of information relevant to company-related affairs, such as accounting facts and figures, operating decisions and board changes and so on, as well as publishing offers for business transactions that include products and services (e.g. GoodRelations), to name a few.

OpenCorporates.com is an open database of companies and corporations around the world, having more than 45 million company entries at the time of writing. It is open, in the sense that it fulfils the *Open Database License*, allowing both crowdsourcing (for adding and modifying entries) and sharing stored data in a free fashion. Entries include data coming from different sources such as local or centralized registries and they are often categorized based on corporate groupings. Board decisions are also available for many of the entries, depending on the state of their existence in national data repositories. Therefore:

- Data about corporate entities do not need to be stored locally. Hence, they require no special curating or maintenance cost at all, as they are queried on demand from the external repository.
- Data entries in the OpenCorporates database are aggregations of data that stem from certified data repositories. Therefore, they are always up-to-date and remotely accessible.
- As data is queried on demand, we can ask for any sub-part of the whole entry we need, without having to deal with data redundancy.

D. Ontologies as Data Structures

Knowledge-based structuring of data in modern approaches implies the use of ontologies, instead of traditional database schemas. As Gruber [6] states, ontologies are practically defined as *context-dependent shared agreements of varied representational levels* that are used to externalize (i.e. describe) the knowledge of a particular domain in order to make it shareable and reusable among remote entities. The use of ontologies in our implementation provides two main benefits. First, the data model is given formal semantics. This opens the way for more intelligent searching, as well as the application of reasoning. Second, a published ontology along with an API makes the data linkable, shareable and interchangeable with the outside world.

III. Conceptual Design

A. Defining the Test Scenario

The domain of *sales* was considered as a test domain for the implemented system. The graph-like nature of sales is based on people (actors in general) and their communications, thus making it a suitable test bed for our implementation. It is expandable, ranging from the presales to the after-sales service levels and many different tasks can be identified within it. In a realistic use case, such a system can be used to assess the business efficiency of salespeople based on their existence and activity within the communications network, improve their efforts based on this and provide a centred method of monitoring and handling business networks in general.

The advantage of using semantic technologies in this approach is that a common substrate of meaning is employed in a context and platform independent way. Communication instances are caused by (and exist within) different and remote sources, such as mobile phones, computers, landlines and so on. Furthermore, their type can vary as well, ranging from phone calls and text messages to shared events. By outlining their inherent meaning they can all be brought together and assessed as objects of comparable nature. Given that the proposed system is user-centric, the emerging communications graph will comprise of the user's node behaving both as a sink and source.

Especially in large enterprise networks, the result will be a highly interconnected graph of dynamic nature, where the nodes share the same knowledge and information principles between each other, without being restricted by the differences in technology the users might be constrained to use. Furthermore, it is interesting that the created network is not isolated from the external world and sometimes the boundaries are not distinct. There is not only high connectivity between internal nodes, but also a constant interaction between the internal and the external world in general.

Given the above, the scenario is defined as follows:

The implemented system is used by a sales company. The combination of external and internal information triggers sales threats. These threats regard prices and offers of particular products/services that the company provides. The system notifies with alerts when threats are triggered. These might be concerning the same products/services offered by competitors, or similar products/services with matching functionality (e.g. a competitor company issues a better offer about a particular product, thus threatening our company's existing clientele). The implemented system handles the alerts in two ways:

a. *Automatic creation of suggestions for handling a particular threat. These include identification of a subset of employees that is suitable to handle the threat, assignment of particular clients (or sets of clients) to these employees, suggestion of dominant communication strategies (by pointing out communication styles) and so on.*
b. *Provision of a browsing web environment with related information and statistics for the particular threat, in order to assist managers in handling it manually.*

Past information drawn from the system is shared among salespeople within the company, and the overall context of the scenario is identified through the company's ERP system. The system automatically uses a set of classifications on the market and clients in order to outline the conditions, the results of which are made available to the salesperson. Finally, the salesperson adapts his/her operational plan according to these, in order to respond to the threat. The results of his/her attempts are quantified and measured before they are made part of the shared knowledge, thus rebooting the cycle.

As can be seen, the purpose of the system is to assist decision-making, as well as automate part of it. The system's input is made up of the sources that were described in section II. The company's salespeople handle both new and existing clients. The system builds communication profiles for both clients and employees, based on past transactions and historical data. Therefore, when an existing client needs to be approached, the salesperson can consult the client's profile in order to select a method of approach. In the case of new clients, the market profile, as well as the product/service profile are identified and adopted. This is powerful because even when we don't have information on a new client's communication needs, we can approximate the approach by studying historical data about the product/service of interest and the market, identifying and quantifying the patterns that arise.

B. Ontology Engineering

An integral part of any system with semantic capabilities is its underlying ontology. Ontologies are being used as a means of formally representing knowledge in ways that allow for certain kinds of reasoning to be performed. This makes knowledge-based processing of information possible and allows for the application of rules and the implantation of rule-based scenarios, which in turn lead to the derivation of facts, suggestions or concrete statements. In our case, it will be shown how such rules can provide support on the decision-making processes of sales people.

Reusing existing ontologies is generally preferred over creating new ones, which can be a rather intensive task in terms of cost and work effort. In our approach, parts of existing and communally accepted ontologies have been employed in order to describe high-level concepts and relationships such as people, organizations and companies, products and services. The reuse of existing ontologies is a default recommendation. It is quicker and less expensive, but perhaps most importantly, ontology reuse is what makes knowledge sharing between heterogeneous sources possible.

Part of the system's overall ontology has been created from scratch as per the requirements of the scenario, always having in mind the principle of *minimal ontological commitment* [6]. This simply states that in ontology design, the representational claims should be exactly as many as needed for the description of the domain of interest, in order to avoid knowledge redundancy. Nevertheless, there are cases where excess commitments are needed, having in mind scalability and extensibility issues.

1) Existing Ontologies

In our implementation, several external ontologies have been reused, such as the *GoodRelations Ontology*, the *Web Service Modelling Ontology* (WSMO) and the *Friend-of-a-Friend* (FOAF) vocabulary. To briefly describe the above:

- *GoodRelations* [7] provides a conceptual framework for representing e-commerce offers of goods (products and services) in a way that can be retrieved and processed by semantic web services. This way, product sellers and service providers can publish their offers in a machine-understandable format, whereas their counterparts (buyers, renters, leasers etc.) can automatically find offers relevant to their interests. Its credibility is supported by the fact that it is adopted by large corporations such as Google, Yahoo!, BestBuy and Sears. We use the GoodRelations ontology to represent offers of products and services, along with their metadata (e.g. price, availability etc.)
- *WSMO* provides a reference language for describing web services in order for autonomous agents to be able to search for and select the appropriate ones, depending on the context. We use WSMO to represent web services that are offered by companies.
- *FOAF* is an ontology for the description of individuals and organizations, and the relationships between them. It is widely accepted as a reference model for

the description of people and groups, and has the potential of highlighting networks of social interactions without having to rely on a centralized database. We use the FOAF vocabulary to represent people and organizations.

to adapt easily to external changes, even though in practice we end up with more classes.

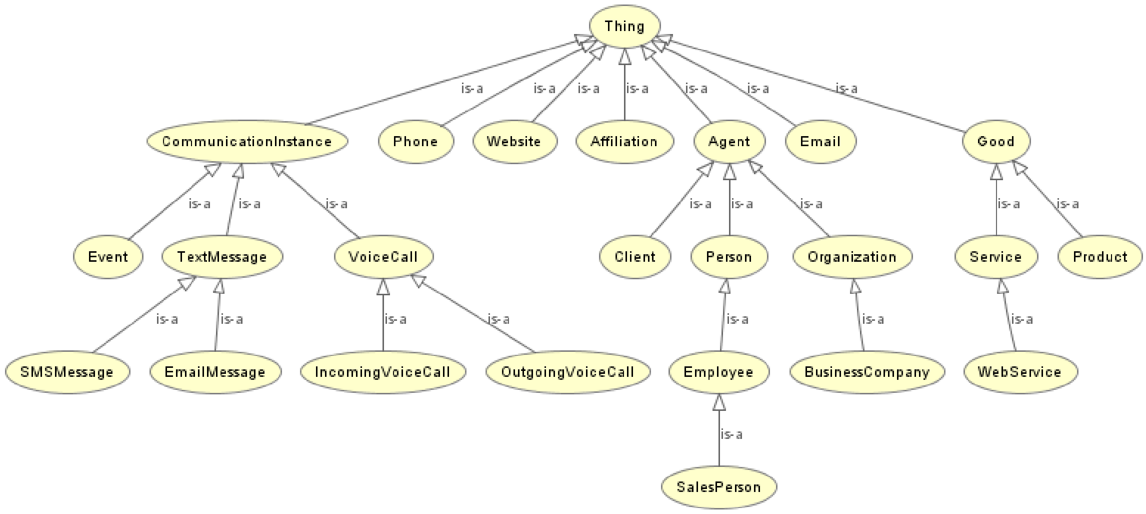

Figure 1: Local Ontology class breakdown

2) Local Ontology

Furthermore, a novel ontology, that will be called *Local Ontology* herein, has been created and implemented for our purposes. Our local ontology helps define the world that is relevant to the implemented scenario, as will be described.

The local ontology classes can be seen in figure 1, with simplified names to assist readability. As can be seen, all types of communications belong to the class *lo:CommunicationInstance*. Furthermore, products and services belong to the class *lo:Good*. Breakdowns of these classes can be seen in the figure as well. The classes *lo:Phone*, *lo:Email* and *lo:Website* are used to represent such resources and their metadata (e.g. type of phone, phone number etc.).

The external ontologies that are used in the implementation are declared to be equivalent classes, where applicable. For instance, an individual of the class *lo:Company* also belongs by definition to the classes *foaf:Organization* and *gr:BusinessEntity*. The borders and interlinking between the ontologies can be seen in figure 3.

As can be seen in figure 3, the external ontologies interact with the local ontology, as well as with each other. Given that we do not use the external ontologies at their entirety, the parts that are actually used are the coloured overlaps between the borders of the external ontologies and the local ontology. The fact that they overlap shows that we are not depending exclusively on these external ontologies, but instead we link existing concepts and relations in our local ontology with the corresponding ones in the external ones, performing a form of local alignment. This is done to reduce coupling, support extensibility and scalability and make the handling of data able

IV. System Implementation

A. Technological Choices

1) Programming Languages

The implemented system and all associated methods and classes are programmed in Java. Semantic web and ontology integration are done with the use of the Jena Semantic Web Framework [8], which provides ways of creating and processing RDF models.

Demonstration is implemented in the form of JSP (JavaServer Pages) on top of simple HTML. The end result is a web environment capable of providing basic demonstrative functionality on the implemented system's capabilities.

2) ERP System

The implemented system is context-driven. The main assumption is the existence of an ERP system that stores information regarding the operations of the company in its internal database. We have chosen to expand on OpenERP [9], which has rounded capabilities and small size. Thomas Herzog [10] shows that OpenERP (cited as TinyERP at the time of writing) covers all necessary functionalities with a smaller set of database tables than the competing systems. Furthermore, OpenERP has a large development community and extensive documentation. Nevertheless, the choice of ERP is meant to be non-restrictive, in the sense that we build upon a neutral system so that correlation with the specific implementation is kept to a minimum, and at the same time effectively taking advantage of the generic ERP functionalities that every system has.

3) Mobile Platform

The implementation uses Android as the scenario's mobile environment. Android provides a programming framework, called the Android SDK, which is practically built on top of

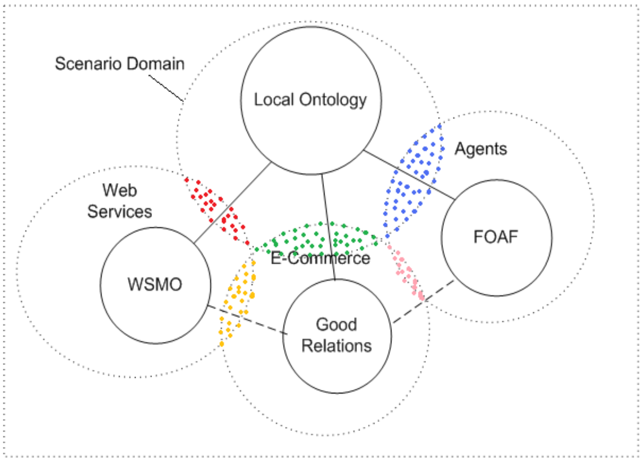

Figure 2: Ontology interaction. The dotted lines around the circles show the impact of each ontology. Conjunctions between different ontologies are shown in coloured dots.

Java classes, making the incorporation of external Java libraries natural. This provides convenient mechanisms for incorporating other implemented components into android devices. A future extension of the proposed system will look into integration of iPhone environments.

4) RDF Store

Because of the use of data from diverse sources (web, mobile phones, ERP database etc.), there needs to be a common denomination, as has been discussed. This is possible with the use of ontologies. Storing data under semantic structure is particularly easy with the use of RDF.

The data that is converted to RDF needs to be stored in specially designed database systems, called triplestores. These are optimized for storing data in the form of triples (subject-predicate-object). We have selected *OpenLink Virtuoso* for use as a non-commercial open source solution that comes with community support. The choice was based on the fact that it has rounded capabilities and good querying performance, as suggested by standardized benchmarking tests [11]. Furthermore, Virtuoso contains java libraries for connecting with the Jena Semantic Web Framework, called the *VirtJena Provider,* therefore making it less complicated for the implemented system to perform RDF data interactions between the Virtuoso server and the various platforms that are used.

B. Information Retrieval and Semantification

Various data sources are used for information retrieval in this scenario, as has been presented. Each of them is queried differently, according to the technologies it is subject to. They are all converted to RDF using the Jena framework, the

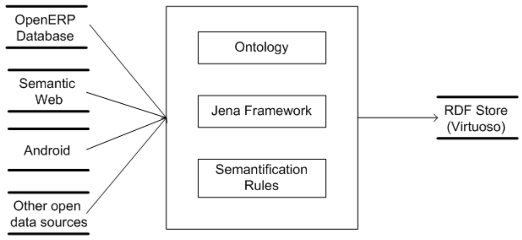

Figure 3: Abstract RDF conversion and storage

system's ontologies and a set of conversion rules. After the conversion, they are stored in Virtuoso. RDF data is then offered depending on the context, and the use of negotiation rules ensures that the availability of the data is limited and controlled accordingly. In practice, this would mean that *different subsets of the data will be made available to different parties at different contexts*.

C. Information Aggregation and Utilization

1) Content Aggregation

As has already been established, ERP systems form the centre of the scenario, in the sense that the ERP is the main context driver. For this reason our implementation stands as an extension to its functionality, providing several layers, both structural and functional, around the domain of interest. The structural layer is a consequence of the provision of semantics, while the functional layers are the sum of all extra functionalities that are implemented on top of the system's current ones. This can be seen in figure 2.

The interconnections between employee and existing client nodes denote established salesperson-client relationships as well as communication relationships between them, which are extracted from the OpenERP database and the communication profiles that are drawn from the employees' corporate Android phones. The potential client space is linked with dotted lines, in order to represent the uncertain communication links between them.

Furthermore, it can be seen that the ERP, which is the central node, is surrounded by a dashed circle, which forms the semantic layer of the system. Information is shared within this layer throughout the extended system. Practically, the data that exists within the information layer is in RDF form, stored in the dedicated Virtuoso store. The mobile data is exchanged at the boundary of the semantic layer. Finally, external information that stems from web sources such as social media and (linked) open data is exchanged through the boundaries of the semantic layer as well. Within these, data is stored in aggregation, under common semantic structure.

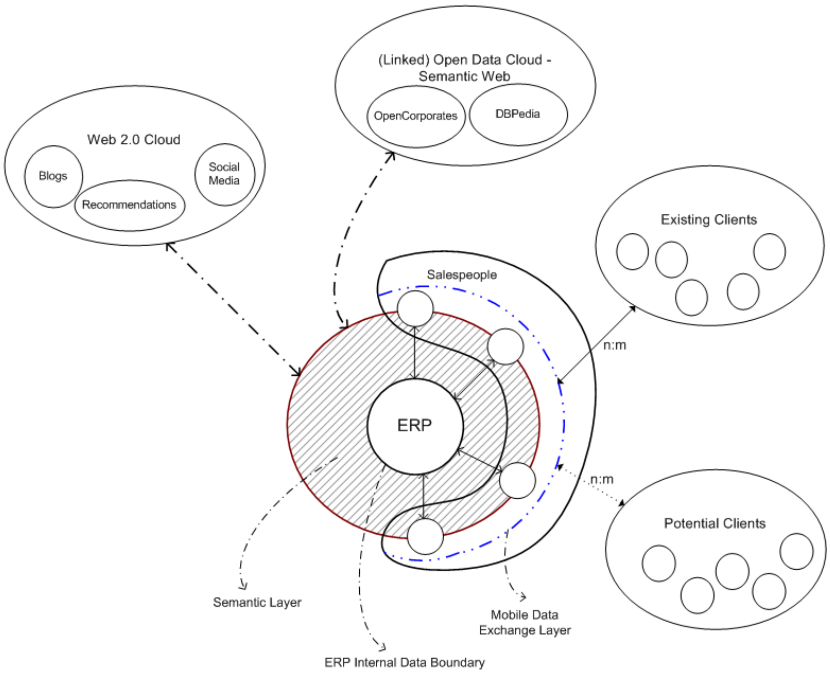

Figure 4: Schematic abstraction of the implemented system

2) Web Interface

In order to demonstrate the capacities of the implementation, an HTML-based website has been created and deployed. This is used as a web environment where employees with managerial status are notified when price alerts are triggered. The implemented version includes a browsing environment of employees, clients and products/services, with respect to various sets of restrictions. Furthermore, the user can read descriptions and metadata of incoming alerts, get automatically-generated handling strategies, assign employees to clients and products, see communication profiles of employees and clients and read several data-centric analyses of cross-patterns between the variables. The web environment is built upon layers of JavaServer Pages (JSP) and Java servlets that use a mixture of Jena classes and HTML design components in order to remotely query the Virtuoso RDF store, analyze the incoming data and provide views accordingly.

a) Alert Notifications

Upon request of this webpage, the server is queried in order to return a list of the incoming, yet unhandled, alerts, along with a subset of their non-functional properties. It is assumed that the alerts are created from external sources, mainly social media. For instance, when a competitor publishes in a social media website a product/service offer that is better, this is extracted automatically and causes an alert to trigger.

This gives the user the first view of the web environment, which is a listing of incoming price alerts, along with a brief overview of their associated metadata. For this reason, it is displayed as a table, where each row is an incoming alert notification which has not been handled yet. The user can browse these notifications and sort them by each of their metadata fields, which are

- Product/Service
- Price
- Business Function (as prescribed by the GoodRelations ontology)
- Severity
- Urgency
- Certainty

By selecting an incoming alert notification, the web environment shows the user a custom profile page of the particular alert. As each price alert is created in reference to a particular product with a particular business function, the *alert profile* page will go on to provide an initial analysis of the alert based on these constants. Based on this design, an alert profile page offers the following structures of information:

1. *Department Rankings*: these are ordered listings of the departments that have been involved with transactions that refer to the particular set of product and business function. There are two available rankings, (i) based on total sales of the particular offering and (ii) based on success rates of similar alerts in the past. These metrics are computed for the employees that are associated with the offering and aggregation takes place over their departments.
2. *Employee Rankings*: similarly, the same procedure takes place for individual employees, without aggregating the results to their respective departments. Only employees that are associated with the particular offering are shown.
3. *Client Rankings*: similarly, clients are queried based on the particular offering, the difference being that there are no success rate measures of clients for past

alerts, as these scores refer to the handling procedure of alerts by employees.

In each case, links are provided for departments, employees, clients and the specified product. Selecting an employee or a client will show his/her communication usage, limited to the particular offering. Product/service links will navigate the user to the full product/service profile, as will be described.

b) Employee, Client and Product/Service profiles

Employee and Client views provide insights to the associated agent, including their static as well as dynamic information profiles. Employee and Client profiles are similar for the most part, with the differences being in the types of associations that are produced. Their conjunction is made up of the following information:

- Static information that defines the employee/client as an ERP entity, including contact details drawn from the Contacts Level. In the case of employees, their department associations are further provided. In the case of clients, a link to data drawn from OpenCorporates is also present.
- An ordered listing of the top products/services associated with the particular agent, along with processed information. Specifically, the following information is retrieved and shown:
    1. Total transactions by number
    2. Total income from past transactions
    3. Rankings of business functions associated with the particular product/service, along with their percentage in income and transactions
    4. Rankings of communication types, with relative percentage rates
    5. Top 3 client rankings associated with the particular product and employee/client
- An ordered listing of the top clients (in the case of employee profiles) and employees (in the case of client profiles), including the following:
    1. Total transactions by number for the particular agent (employee or client)
    2. Total income from past transactions
    3. Top 3 product/service rankings (for the pair employee-client or vice-versa)
- An ordered listing of all communication types associated with the agent, including metadata such as :
    1. Total duration of voice calls that the agent participated in, average duration for the agent and his/her department, average score for the agent and his/her department
    2. Total length (in characters) of all types of messages, average length for agent and department, average score for agent and department
    3. Total duration of events the agent participated in, average duration for agent and department, average score for agent and department
- An ordered listing of all business functions independently of the product/service they refer to, including total number of transactions and total income.

In the case of Product/Service profiles, the static information includes a breakdown of each available business function associated with the product/service, along with pricing information. Furthermore, the four bullets listed above for the case of agent profiles are included here, with the appropriate differences in the subset of constants for each query.

c) Threat handling

Incoming threat alerts are ranked and classified based on their description. The system offers its users the possibility to assess the threat automatically. This means that after an incoming threat is qualified, the system will suggest the best means to handle it, based on particular rule sets. In our implementation, incoming threats are always in reference to a particular product/service. The threatened clients are identified and their profiles assessed. The information that can be derived from these profiles gives valuable intelligence on how these clients were handled in the past, which employees have participated in transactions with them, how successful historical transactions were, what methods of communication they prefer, or what methods of communication are the most successful with the particular product/service. Moreover, the system builds product/service profiles for the threatened good, in order for threat handlers to assess situations where past historical data are absent. The overall alert resolution process can be seen in figure 5.

Evaluation of incoming alerts is one step before assigning handlers to the individual alerts themselves, which is the adoption of sales strategies and effective handling from salespeople. The different metadata associated with each alert can help classify and categorize them, in order to assign their assessment to distinct evaluators. For instance, manual investigation of alerts can be assigned to evaluators on the basis of their severity, urgency, certainty, or combination of these. Technically this can be achieved as follows: the web environment is user-dependent (a state that can be activated with simple username/password usage) and identification of the user leads to the incorporation of accessibility rules that will restrict the query results accordingly. Inexperienced evaluators will not be shown high-severity alerts, while experienced ones will have access to all.

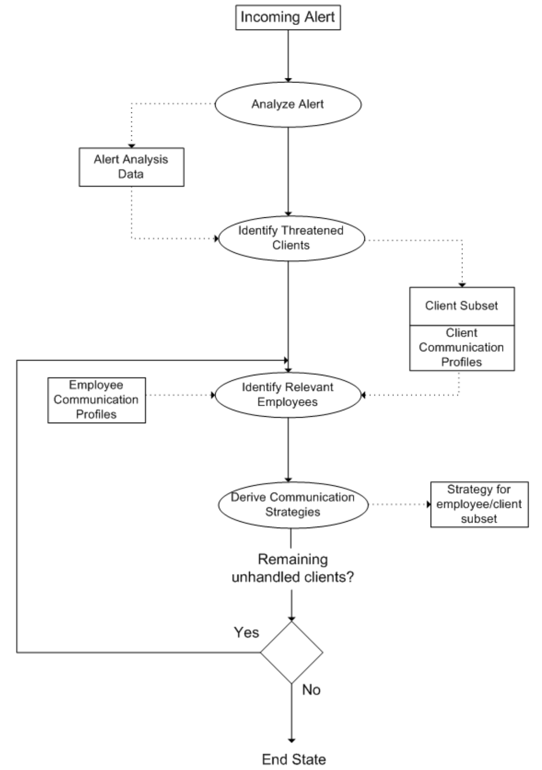

Figure 5:Activity diagram of the alert resolution process

D. Test Data

The creation of test data is done programmatically, with the use of Java classes. These are concerned with three types of data, depending on their (assumed) origin, namely OpenERP, Android test data and data from external sources.

The OpenERP database is populated with a number of distinct employees, clients, departments and products/services, the breakdown of which can be seen in the table that follows. These data are considered to be at the first level of the data creation process, as they are needed as a basis for what follows, namely transactions, communications and so on. The above four dimensions (employees, clients, products/services and departments) interact with each other in accordance with the given ontological schemata in order to create information that resides in higher levels, as has already been discussed. More particularly:

- Employee and Client data are converted to Android contacts in a test Android environment.
- Using the Android SDK, the test Android environment is populated with communication instances of various types. It should be noted that the Android environment is meant to be an aggregation of multiple Android phones. This leads to the creation of communication data between the employee and client sets, with a n:m type of association.
- In order to simulate a realistic environment of employee-client interactions, a semi-random mechanism has been implemented, that uses existing handler relationships drawn from OpenERP, to identify subsets of clients for each employee that are most likely to have interacted communicatively under the context of shared transactions.

External source and usage data are concerned with alerts (present and past) and their metadata, web sources that act as creators of alerts, such as data mined from social media as well as scores for communications and success rates for past alerts. Moreover, clients have been linked to real companies from Opencorporates.com, whose details are updated on demand from the Opencorporates.com database.

Apart from past transactions and communication instances, a set of present and past alerts is created. Past alerts are needed in order to populate the data store with information about how employees handled price alerts in the past. The employees that are tagged as handlers of past alerts are chosen by taking into consideration their correlation with the offering that is associated with the particular alert, making sure that there is a certain degree of realistic consistency throughout the test data. Furthermore, alerts are assigned severity, urgency and certainty measures.

V. Results and conclusions

This work provides a conceptual framework and a demonstrative implementation for taking advantage of the business communication layer of information, which has grown to be a diverse environment of heterogeneous sources and causality relationships. Business communications cover a large range of information flows between business and public agents. The heterogeneity of the term is apparent when considering that business communications stem from both internal and external contexts. External contexts are created from data-generating sources that range from social media to semantic web related interoperability architectures, while internal contexts are created and controlled within the company's resources (and their directly linked nodes). Several screenshots from the web interface can be seen in figure 6.

The implemented system's functionalities are examined under several different perspectives, in reference to how business capabilities can be improved. These are:

1. Knowledge Management Perspective
2. Semantic Web Perspective
3. Platform Unification Perspective

A. Knowledge Management Perspective

The implementation serves as a demonstration of how modern ERP systems can be provided with access to this type of information, under a common knowledge substrate. The underlying ontology provides support for internal knowledge management, making the querying and serving of data more intelligent and multidimensional, thereby complementing on the functionality of traditional databases. In the case of the

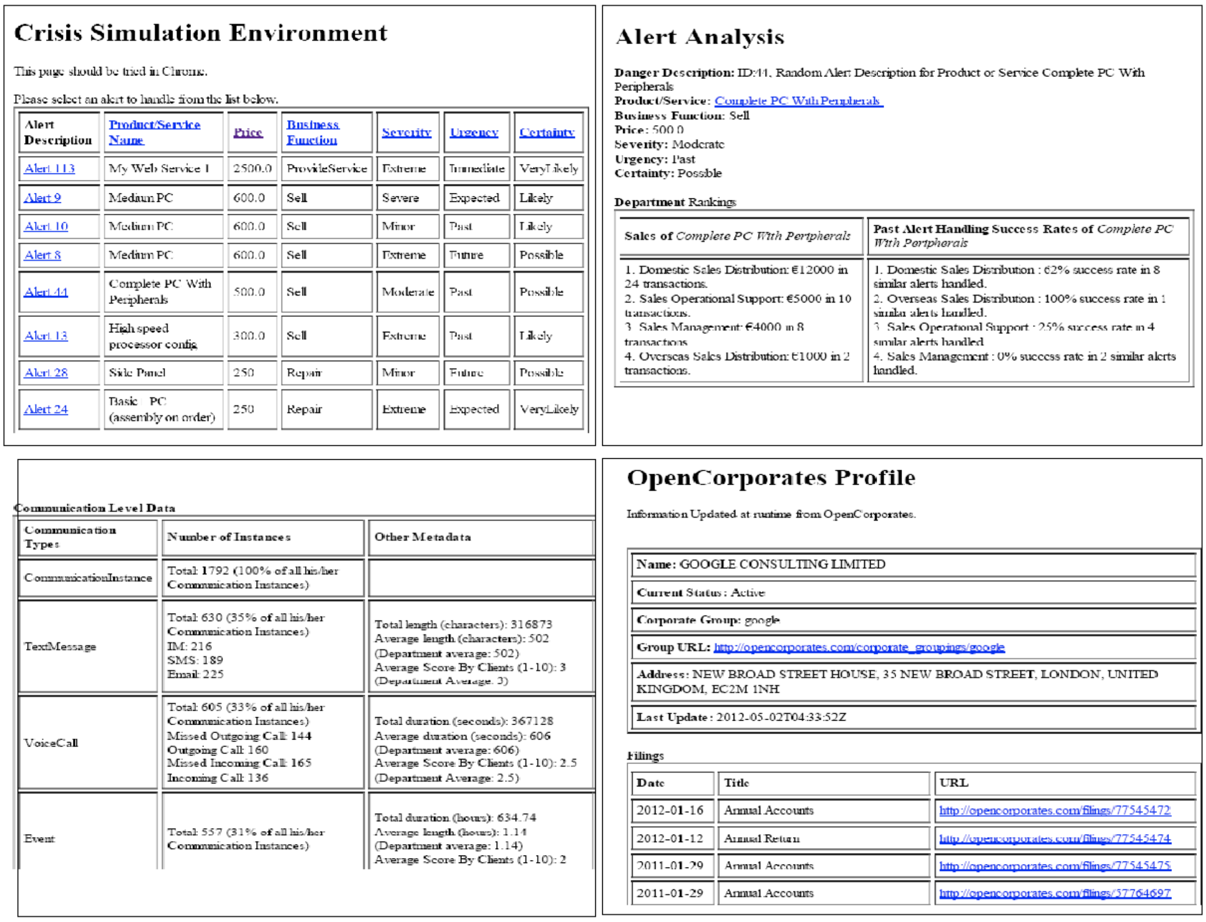

Figure 6: Screenshots from the web interface

implemented scenario, rules are used in several occasions, such as:

- Controlled data access: using ontology and inference-based accessibility rules, permission negotiation is performed at run-time and varies with respect to the inferred context.
- Ontology-based classification: along with ontological inferences, which lie in the first layer of reasoning, rule sets are used on ontology schemata in order to perform custom classifications, as is done in the case of alert handling.

Providing knowledge management to existing ERP systems cannot, on its own, improve and evolve business management, because in closed worlds an ERP system is exclusively internal. The power of this approach comes when knowledge-based internal support is accompanied with extensions to accommodate and manage third-party data and information, as is the case under the semantic web scope.

B. Semantic Web Perspective

With the use of semantic web standards and technologies, the system is ready for integration with external semantic web and web 3.0 resources. In the business environment, web services are emerging as ways of incorporating knowledge exchange in the markets, making the sharing of information a task that can be handled by automated software agents. As these agents are usually components of larger systems, the retrieved knowledge can be processed and consumed internally without limitations.

For instance, it was mentioned that large corporations such as Google and BuyMore use GoodRelations descriptions for offered goods. It has been demonstrated how such descriptions can be retrieved and processed internally within the ERP system, without the need to design and deploy costly data integration components. The existing list of products and services of the scenario's company is easily mapped to classes and properties taken from GoodRelations.

Using external sources of information, such as the OpenCorporates database, has the advantage of liberating local data storage and its management, while providing up-to-date information that is externally curated. In the implementation, the client list has been taken directly from OpenCorporates, the database of which is queried on demand and on the fly. The queried data are connected with the semantic layers of the implementation, thereby providing uniform views of heterogeneous data.

C. Platform Unification Perspective

The previous two points show that both perspectives share a common goal, which is the unification of the various data- and value-creating platforms under one scheme, in order not only to combine and manage diverse information, but to make it machine-understandable as well. The proposed extensions form a unification layer around the company's core (as seen from the ERP point of view), that can take input and give output on demand and under rule-based control. The implications as far as business management improvements are concerned, have to do with the following points.

1. Minimization of response time: it has been briefly discussed how business threats can be identified and quantified, automatically or manually. What adds value through this approach is that the response time is minimized, and scaled down to “web-time”.
2. Enrichment of internal context: the internal context, as defined by ERP systems, is enriched and expanded in a way that converts the traditional closed-world paradigm to a more open business environment, that takes advantage of abundant information and promotes interoperability.
3. Ontologies remove restrictions: any technological limitations that are imposed by data management are removed with the use of shared, structured meaning.
4. Exploitation of mobile information: mobile devices provide an abundance of usage data and metadata, thus defining contexts (and sub-contexts) on their own. The incorporation of these to a company's business communication management, and the semantification of the outcome provides powerful views to what was originally perceived as tacit knowledge.

References

[1] R. Herschel and I. Yermish, “Knowledge Management in Business Intelligence,” vol. 4, W. R. King, Ed. Springer US, 2009, pp. 131–143.

[2] H. R. Choi, H. S. Kim, B. J. Park, N.-K. Park, and S. W. Lee, “An ERP approach for container terminal operating systems,” *Maritime Policy & Management*, vol. 30, no. 3, pp. 197–210, Jul. 2003.

[3] A. Smaizys and O. Vasilecas, “Business Rules Based Agile ERP Systems Development,” vol. 20, no. 3, pp. 439–460, 2009.

[4] H. Graf and N. Tellian, “Smartphones as enabler of Supply Chain Event Management Upper Austria University of Applied Sciences.”

[5] A. Davies, “Language-independent Bayesian sentiment mining of Twitter TWITTER DATA , GEO MODELLING AND,” vol. 11, 2011.

[6] T. R. Gruber, “Toward Principles for the Design of Ontologies Used for Knowledge Sharing,” 1993.

[7] M. Hepp, “GoodRelations : An Ontology for Describing Products and Services Offers on the Web,” pp. 329–346, 2008.

[8] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, “Jena: implementing the semantic web recommendations,” in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, 2004, pp. 74–83.

[9] F. Pinckaers and G. Gardiner, *A Modern Approach to Integrated Business Management*. 2009.

[10] T. Herzog, “A Comparison of Open Source ERP Systems,” no. June, 2006.

[11] V. Mironov, N. Seethappan, and W. Blonde, “Benchmarking triple stores with biological data,” *Web Applications and*, 2010.