Linking Social, Open, and Enterprise Data

Tope Omitola
Electronics and Computer Science
University of Southampton, UK
tobo@ecs.soton.ac.uk

John Davies
British Telecommunicatons, plc. UK
john.nj.davies@bt.com

Alistair Duke
British Telecommunicatons, plc. UK
alistair.duke@bt.com

Hugh Glaser
Seme4 Ltd., UK
hugh.glaser@seme4.com

Nigel Shadbolt
Electronics and Computer Science
University of Southampton, UK
nrs@ecs.soton.ac.uk

ABSTRACT

The new world of big data, of the LOD cloud, of the app economy, and of social media means that organisations no longer own, much less control, all the data they need to make the best informed business decisions. In this paper, we describe how we built a system using Linked Data principles to bring in data from Web 2.0 sites (LinkedIn, Salesforce), and other external business sites such as OpenCorporates, linking these together with pertinent internal British Telecommunications enterprise data into that enterprise data space. We describe the challenges faced during the implementation, which include sourcing the datasets, finding the appropriate "join points" from the individual datasets, as well as developing the client application used for data publication. We describe our solutions to these challenges and discuss the design decisions made. We conclude by drawing some general principles from this work.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Web-based services; H.4.3 [**Communications Applications**]: Information browsers; H.5.4 [**Hypertext/Hypermedia**]: Architectures, Navigation

General Terms

Hypertext/Hypermedia

Keywords

Hypertext/Hypermedia, Architectures, Navigation, User issues, Semantic networks

WIMS'14, June 2-4, 2014 - Thessaloniki, Greece

ACM 978-1-4503-2538-7/14/06...$15.00
http://dx.doi.org/10.1145/2611040.2611086 .

1. INTRODUCTION

Data is everywhere. Each phone call, email, chat request, or person-to-person interaction between a customer and a brand provides organisations with potentially invaluable information. This wealth of data can reveal precious insight into customers' needs and desires, allowing companies to personalise their communications and services. It has been observed that in the last few years, productivity in businesses has been driven by finding efficiencies and through pricing mechanisms, but essentially a business revolves around customers and their social connections, for example, churners influence others to churn while adopters encourage others to adopt. For a company to thrive, this wealth of data needs to be used to identify opportunities in new sectors, support employees, customers, and other external partners. While a company can use these data to create a more intelligent understanding of their customers, this is only the beginning of a whole series of positive outcomes - from reinforced retention to reduced churn, from greater adoption to new service introduction, from clear decision making to more accurate sales spend.

This surfeit of customers' data is being buttressed by other factors. The rise of social media, the rise of Big Data, and the opening up of public data by governments around the world are inputs that a company can make use of to capture value and create new services. In order for this to occur, an enterprise needs unified access to all its data and information structures, and these external data sources. However, traditional enterprise systems have grown through slow accretion and segregated into information silos making it very difficult to connect and share data even within an organisation's operating boundaries. This restricted ability to cut across datasets within an organisation represents an Achilles' heel for their growth, preventing them from taking advantage of this data revolution.

On the other hand, data silos are not necessarily bad in and of themselves. They are important for accountability, focus, and specialisation. Data silos allow departments within an enterprise to distinguish themselves and to achieve their unique goals. The key to success for enterprises in this data revolution is to retain the good elements of silos and to devise mechanisms to link related, but previously disconnected, data elements. For example effective data linking mechanisms will allow a telecommunications company to link sales and service data, giving call centre staff the needed information to sell new products to customers calling in with service requests. Such data linking mechanisms will allow data to find data.

1.1 Data Finding Data

When systems are constructed to make it easy for "data to find data", there exists an opportunity for new insights, insights on new relationships between an enterprise's historical data and the incoming data, insights into how these relationships can create value, and how they can be used to create new services. The ability to access data or information from multiple data sources is known as information inter-operability, and Linked Data has been found to be an enabling technology for system and data inter-operation.

2. RELATED WORK

System inter-operation has been a perennial problem in the history of information systems. Linked Data has been found to be an enabling technology for enterprise system integration. Mihindukulasooriya et. al.[1] described how a Linked Data platform (LDP) can be used as an infrastructure for enterprise application integration (EAI). They described the challenges and presented advantages for using an LDP for solving these challenges. Frischmuth et. al.[2] discussed how Linked Data can be used for EAI, and presented several examples of its successful applications in EAI systems.

Clarke and Greig[3] used Linked Data to unify the descriptions of teaching resources and to support the semi-automatic harvesting of resources from the web. The system allowed teachers and students to annotate the resources as metadata to be kept private via partitioning of data stores, while the resulting data were published using XHTML documents embedded with RDFa making them consumable by both humans and machines, as appropriate.

Borst and Neubert[4] focussed on publishing the "Thesaurus of the German National Library of Economics" as Linked Data to increase access and reuse by other people and institutions. The authors chose to map their thesaurus to the Simple Knowledge Organisation System (SKOS) which they extended using other vocabularies such as Dublin Core. The data was published via generated XHTML pages with embedded RDFa allowing it to be used by humans and machines via content negotiation, while links to other resources such as library catalogues or dbpedia entries could be easily made.

Le Hors and Speicher[5] used Linked Data as an integration mechanism to overcome the problems of existing application integration architecture (e.g. APIs between all applications or a common database for all applications). Points of contact between applications were modelled as resources communicated via HTTP requests. Thus, in the context of software lifecycle management, a change request became an RDF resource linked to the defect being addressed. This removed the need for specific interfaces between discrete applications, such as change and test management applications, which instead accessed the resources directly, following Linked Data principles.

Similar to the studies described above, we use Linked Data as the integration layer for the different systems that are part of our architecture. However, our architecture is different. As we have had to work with existing databases and also because of performance, we have needed to bring these datasources into our storage, as RDF resources.

Ontologies, a central component of Linked Data, has been successfully applied in Ontology-based integration of information systems. Buccella et. al.[6] discussed the various ways of how ontologies have been applied for data integration. They described how ontologies have been used to solve the problem of syntactic and semantic heterogeneity prevalent in EAI systems. Gagnon[7] proposed an ontology-based integration methodology with a local to global ontology mapping to solve the issue of semantic heterogeneity of enterprise integration. Ontology-based data access and management (OBDM) is a methodology that is being effectively used to access, integrate, and manage data in big enterprises. It consists of a three-level architecture constituting an ontology, the data sources, and the mapping between the two. Antonioli et. al.[8] described their experience of successfully using OBDM to integrate systems at the Italian Department of Treasury.

In this study, we apply the tools and ideas of OBDM for data integration, especially to improve customer relationship systems within the enterprise.

3. USE CASE STUDY

As described above, enterprise systems have traditionally grown through slow accretion and separated into information silos, e.g., each new operating need has generated an ad-hoc application: ERP, CRM, ECM, directories, messaging, etc. But, with the new data revolution taking place, enterprises, such as BT, face many challenges, two of which are:

- How to manage and extract value from these disparate, isolated data, and
- How to take advantage of the information from external, non-enterprise, and other social media data to provide new, exciting, and useful services that create value for customers, which ultimately will affect the balance sheets of these businesses.

At BT, we have been investigating how Semantic technologies can be used to leverage the effort of our sales and commercial staff to help them be more responsive to customers, and to identify new sales opportunities.

3.1 The Application: Linking Enterprise, Social, and Open Data

The motivation behind the application is that BT wishes to have its people, especially the sales staff, to forge stronger links between one another, and to be able to make connecting with customers easier. In this regard, we are making use of an online social network, LinkedIn[1], tapping BT employees' social connections and using that to form stronger links and help with introductions to new customers. There are three components to the application: (a) A separate component that enables BT employees to log in, get their first and second order connections, and link these up with other internal BT systems, (b) a second component that is used to join together employees' human resources data with data of companies that are clients of the BT's sales department and joins these up with open external data of these companies,

[1] http://www.linkedin.com/

allowing sales leads and managers to get more information that can lead to new customers and new sales. None of these data is published in Linked Data format by the sources, and it is the middleware that transforms them into Linked Data, choosing the appropriate join points, and (c) a third, Linked Data consumption component that leads and managers can use to interact with the system.

3.1.1 LinkedIn Meshing.

In this application, a user (who had earlier volunteered for us to use their LinkedIn connections) signs into the application using their name, company email address, and employee identification number (EIN). The key challenge, which is widely studied, in disambiguating a person's identification is to decide whether similar names refer to the same person. Various solutions have been devised in the past, including Cosine similarity, Porter stemming, and other various Natural Language Processing techniques. Since EINs and company email addresses are unique to every employee (in this company), we have decided to use this more practical solution for name disambiguation of engaging the employees in giving the system their identifiers. After logging in, we make use of LinkedIn's APIs, and using LinkedIn's handle of the employee, we obtain their first- and second-order connections, especially their locations, job details, and industries they work in. This connection's information is transformed into Linked Data and inserted into our RDF store.

3.1.2 Linking in into other Internal and External Systems.

Here, the (data) middleware connects to the employee resourcing system, and also with data hosted on the local SalesForce system that had been collected by the sales team, and which contain the names of companies (known as "accounts") that the sales team sells products to. In order to help sales leads know more about these accounts (i.e. companies), we source public information of these accounts from Opencorporates[2], which is an Open Data company that aspires to "have a URL (and associated information) for every company in the world". The data middleware transforms all these into Linked Data format setting the appropriate linkages between these datasets, and inserts them into our RDF store.

3.1.3 Linked Data Consumption.

The application also contains a Consumption element that is used by sales leads and managers to interact with the system.

3.1.4 Prototype System

A prototype system is being trialled. The system currently hosts over seven thousand BT employees and five thousand external company accounts.

4. CHALLENGES OF LINKING ENTERPRISE DATA WITH EXTERNAL DATA

Enterprise data lend themselves as both an opportunity and a challenge. They have well-defined boundaries and protocols regulating the transition and the data flow across boundaries. Some of these protocols are neither formalised nor written down to any extent, and many employees in these enterprises resort to "copy-and-paste" methods for the data to be used across inter-company boundaries. Also, some of the datasets are stored in different databases, some in spreadsheets, and some in free-form texts. These data may be well understood by the people working in the relevant departments, but their contexts and semantics may have been lost when they cross departmental boundaries. Therefore, when it comes to inter-departmental data exchange, it is difficult to know what to link with what.

These problems are magnified when it comes to linking enterprise data with external data. The external data sources engendered by the current open and social data revolution are of a different tenor. They change continuously, their shape are evanescent, and their curation, inter-linking, and analysis needs are different. Some are unstructured, while some are semi-structured.

Some of the challenges when attempting to link enterprise and external data, which we observed when building the application, include:

1. Establishing the Business Case. Before attempting to interlink with external data, the enterprise needs to set out its goals and establish what data needs to be linked.

2. Data Discovery and Provenance. The appropriate internal and external data need to be searched for and discovered. For internal data, this job is made easier if the goals and business case have been set out, as these goals would direct attention to the internal data useful to achieve the interlinking. However, this task is more challenging for discovering external data. There are many external datasets that cover "general knowledge", such as Wikipedia (and its Linked Data equivalent, DBpedia). Since enterprises live on their internal knowledge which they may like to combine with domain-specific knowledge (and data), "domain-specific knowledge bases", such as Bio2RDF[3] (popular in genomic work), are more useful for enterprise inter-linking. When it comes to quality, many external datasets are found wanting. This is exacerbated in the Linked Open Data cloud, where many of the datasets are maintained by one person or not at all. Therefore the case of data quality, licensing, provenance, and trustworthiness need to be carefully judged, for external data sources.

3. Information Extraction and Data Harvesting. Once the appropriate internal and external datasets have been identified, in many cases, not all the data will be useful. So, the pertinent data items will need to be extracted. These datasets may also be stored in different databases requiring different data access protocols.

   In the enterprise context, some datasets are stored in relational databases which may require data access protocols such as JDBC and ODBC, while some datasets are stored in LDAP-directories such as Novell's eDirectory, and some are stored in spreadsheets. In addition, the different data models also need to be taken into consideration before the right items can be extracted.

[2] http://www.opencorporates.com

[3] http://bio2rdf.org/

When it comes to external datasets, many Web2.0 sites provide APIs to access their data, while some need to be screen-scraped. For Linked Data sites, their datasets can be downloaded in archive form, while some need to be extracted using SPARQL[4] queries.

4. Data Cleaning. Most of these datasets, internal and external, will not be in the format that can be made use of. Mechanisms for data cleaning need to be developed to make them useful for the inter-linking process.

5. Data Interlinking (or the Choice of Join Points). Data interlinking involves the setting of links between different data sources, and is very much application-specific. For enterprise data, not constructed with linking in mind, it is difficult to know which entities within them to use for interlinking. This job is made easier if the earlier stage of setting up the Business Case has been done successfully. These internal enterprise entities are the ones that guide the entities we choose from the external datasets.

6. Data Modelling. A goal of interoperation is to have a unified access to these disparate data sources. This unified access will enable a holistic view of the enterprise and its external connections and lead to ease of querying. It is useful to make use of a unified ontology that describes the domain of interest. This ontology should include the entities and their attributes that are germane to the domain. One advantage of this is the provision of a declarative approach to information integration. By making the representation of the domain explicit in the ontology, we achieve reusability of the agreement protocols between the various players, and also of the acquired knowledge. Another advantage of this unified ontology to drive the integration process is that we do not need to merge all the data sources in advance. This process of integration can be done incrementally as these sources are discovered, thereby driving down integration costs.

7. Data Management. Some of the data may need to be brought in at run-time, or cached and made available when needed. Therefore, policies and appropriate mechanisms to manage the live and cached data sets need to be devised. These policies may be driven by rate of use of the application within the company, speed of update and the quality of service provided by the external data server.

The aforementioned challenges motivated our system design and architecture.

5. SYSTEM DESIGN AND ARCHITECTURE

The system (figure 1) consists of the following modules.

5.1 Application Layer

Users interact with the application and with the system via the Application Layer. There also exists a command line interface (CLI) that can be used to interact with the system.

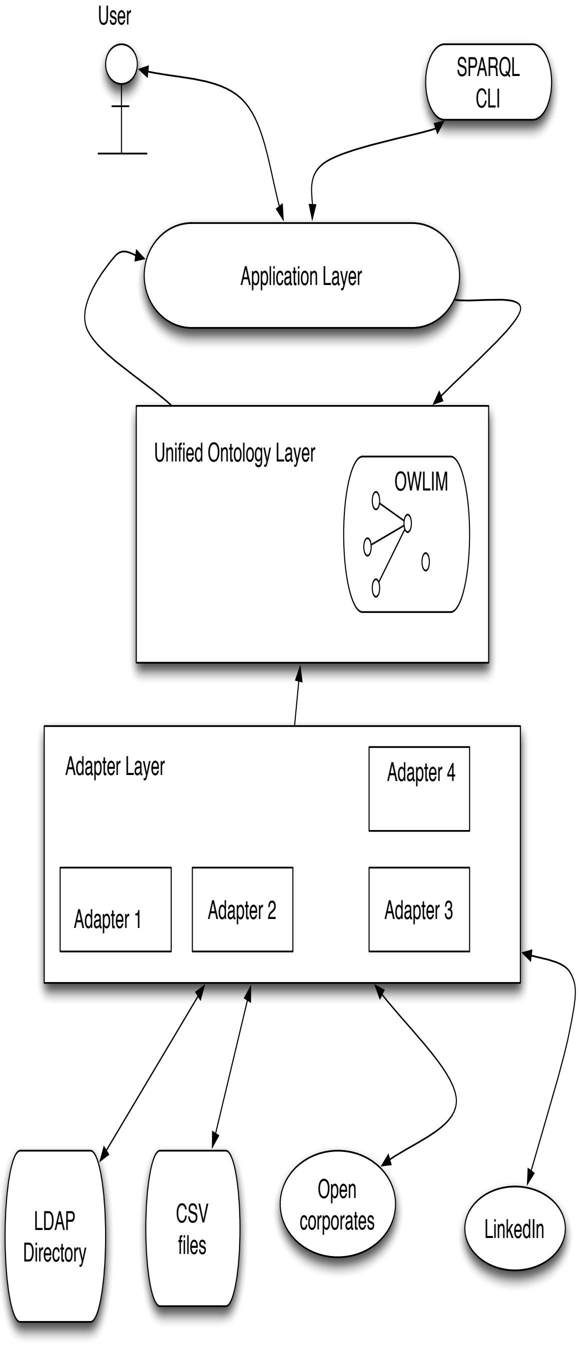

Figure 1: Architecture for linking social, open and enterprise data, showing the interactions of the Ontology layer with the Adapter Layer. The Ontology layer also manages the RDF store (Owlim).

[4] http://www.w3.org/TR/2013/REC-sparql11-query-20130321/

5.2 The Unified Ontology Layer

The system includes an ontology that drives the data integration process. This ontology contains the entities of interest, and they are:

1. BTEmployee: This class represents an employee of BT. This is modelled as a subclass of a foaf Person and a cube Observation. It has the following properties:
   - hasJobTitle: This property's input (i.e. its domain) is a BTEmployee and its output (i.e. its range) is a string
   - hasOUC: this takes a BTEmployee and tells us which OUC (organisation unit code) they have,
   - worksinLineOfBusiness: this gives the line of business (LOB) the BTEmployee works in,
   - hasManager: gives the manager of this BTEmployee,
   - hasEmployeeUIN: gives the User Identification Number (UIN) of this employee,
   - hasManagerUIN: gives the manager's UIN of this employee; and
   - hasBusinessUnit: gives the business unit this BTEmployee belongs.
2. SalesForceUser: This class models an individual who is part of the Sales team. They may, or not, be an employee of BT. This class has the following properties: (a) hasOUC: this property outputs the OUC of this individual, and (b) has Account: this outputs the Account, i.e., the company this individual looks after.
3. EmployeeSocialMediaUser: This class represents an employee that uses their BT identity on social media.
4. Account: This class represents a specific account, i.e. a company, that a BT employee looks after in a particular role. The class has the following property: hasSICCode: this property gives the Standard Industrial Classification (SIC) code of this company account.
5. AccountRole: This class models a BT employee working on an Account in a particular role. It has the following properties: (a) hasRoleName: this represents the name of this role, (b) hasAssociatedAccount: this gives the Account of this role, (c) hasEmployee: this gives the employee taking on this role.
6. BusinessUnit: This represents a Business Unit within BT.
7. SICCode: This represents the Standard Industrial Classification of the company of which an account is held.

A snippet of this ontology is shown below:

```
def:BTEmployee rdf:type owl:Class ;
 rdfs:label "BTEmployee: A BT Employee"@en ;
 rdfs:subClassOf scovo:Item, qb:Observation,
 foaf:Person .
def:EmployeeSocialMediaUser rdf:type owl:Class ;
 rdfs:label "EmployeeSocialMediaUser: A BT
   Employee that uses their BT identity on
   social media"@en ;
 rdfs:subClassOf scovo:Item, qb:Observation,
 foaf:Person, foaf:OnlineAccount .
def:SalesForceUser rdf:type owl:Class ;
 rdfs:label "SalesForceUser: A Salesforce user"@en;
 rdfs:subClassOf scovo:Item, qb:Observation,
 foaf:Person .
def:Account rdf:type owl:Class ;
 rdfs:label "Account: A BT account"@en ;
 rdfs:subClassOf org:Organization .
def:AccountRole rdf:type owl:Class ;
 rdfs:label "AccountRole: a specific role/account
 pair"@en .
def:BusinessUnit rdf:type owl:Class ;
 rdfs:label "BusinessUnit: A BusinessUnit within
 British Telecom"@en ;
 rdfs:subClassOf org:OrganizationalUnit,
 scovo:Dimension,
 qb:DimensionProperty  .
def:SICCode rdf:type owl:Class ;
 rdfs:label "SICCode: The Standard
  Industrial Classification code of the company
  of which the
  account is held"@en ;
 rdfs:subClassOf scovo:Dimension,
 qb:DimensionProperty  .
```

The RDF data store we use is OWLIM[5]. We choose OWLIM for its automatic handling of "sameAs" inferencing. As we are integrating different datasets, some of the entities in these datasets refer to the same concept. We would like to assert co-references amongst these entities. Co-reference resolution is a significant hurdle to overcome in the realisation of large Semantic Web applications. By setting these co-references ourselves, using "owl:sameAs", and having OWLIM resolves these for us automatically has been useful. There have been controversies regarding the use of "sameAs" in Linked Data[9]. Our intensional meaning of "sameAs" is the same as the one given in 4.2 of [9].

5.3 Adapter Layer for Information Extraction, Data Harvesting, and Interlinking

As our data sources require different data access protocols, we built adapters to help us retrieve pertinent data and to turn these into RDF (turtle format[6]).

Our internal datasets were primarily sourced from (a) data stores holding details of the sales staff and the accounts (i.e. the companies) of which they are in charge (The SalesForceStaff datastore), and (b) the enterprise General Employee data store. The General Employee datastore is an LDAP-backed datastore with the following data model:

```
o=bt
 ou=btplc
  ou=people
    <live people records>
    ou=deleted
      <people deleted in last 3 months>
  ou=functional
     <live functional accounts>
     ou=deleted
```

[5] http://owlim.ontotext.com/display/OWLIMv54/OWLIM-Lite

[6] We chose turtle for its compactness and clarity.

```
        <functional accounts deleted in last 3 months>
  ou=buildings
     <all active locations>
  ou=reference
    ou=companyindicator
     <all company/LoB codes and descriptions>
     ...
```

We built an adapter for this LDAP repository to access the data in the aforementioned nodes, and to convert these data into RDF. A snippet is shown below:

```
<http://data.bt.com/id/staff/directory/ds1/
   9f8ed380-124d-48fa-8dfd-4ac6273356f8>
   rdf:type def:BTEmployee ;
   def:hasEmployeeUIN "123456789"^^xsd:integer ;
   foaf:familyName "Duke"^^xsd:string ;
   foaf:name "Ali Duke"^^xsd:string ;
   foaf:givenName "Ali"^^xsd:string ;
   def:hasJobTitle "Principal Researcher"^^xsd:string ;
   def:lineOfBusiness lob:TSO ;
   def:hasBusinessUnit id:TUB ;
   foaf:mbox "ali.duke@bt.com"^^xsd:string ;
   def:hasManager
      <http://data.bt.com/id/staff/directory/ds1/
         65f0c469-9b9f-4e58-9a21-510bfd5ad9e1>
      ... .
```

The SalesForceStaff datastore is a set of spreadsheets. We have made use of two main spreadsheets, and they are: (a)The SalesForceTeamMember spreadsheet. This is a Comma Separated Value (CSV) delimited file with several columns. For this file, the entities of interest are: the first and last names, the EIN, and the email. The second set of spreadsheet, called the ExternalAccountInfo file, contains several columns, of which we are interested in AccountName, SIC (Standard Industrial Classification code), Account Manager's EIN, Account Manager's Name, AccountID, Account Team Member's Name, and Account Team Member's Role. The AccountName is the name of an external (non-BT) company which is a client of the Sales team, Account Manager's Name is the name of a BT employee who manages this account, and the person performing a particular role on the account is the Account Team Member.

We built an adapter used to access the CSV files, and to correlate the Account Team Member's EIN with the their EIN from the General Employee datastore. We also set "sameAs" links between this team member and their entry from the General Employee datastore. After triplification, the above data structure is converted into:

```
<http://data.bt.com/id/staff/salesforce/directory/ds1
  /13/07038ad3-eadb-4ab7-a500-d064566a4791/>
  rdf:type def:SalesForceUser ;
  foaf:familyName   "Duke"^^xsd:string ;
  foaf:firstName   "Ali"^^xsd:string ;
  def:hasEmployeeUIN   "123456789"^^xsd:integer ;
  owl:sameAs
    <http://data.bt.com/id/staff/directory/ds1/
        9f8ed380-124d-48fa-8dfd-4ac6273356f8>;
  foaf:mbox "ali.duke@bt.com"^^xsd:string;
  org:organization   "BT"^^xsd:string ;.
```

In order to generate persistent URIs for these entities, we have eschewed the usual practice of appending an incremented counter to the URI as this may lead to errors if URI re-generation is needed[7]. We have used the universally unique identifier (UUID) class provided by Java to provide us with an immutable unique identifier.

5.3.1 The Choice of Join-points

In order to mesh data sources together, join points need to be decided. The join points will depend on the business case and the goals of the data integration. For our data integration process, we decide on the following join points:

1. For LinkedIn, the join points are the name and employer of the (BT) employee. These are used to retrieve the data from LinkedIn, which are transformed into RDF. The retrieved data from LinkedIn are the first-order and second-order connections of the employee. An example snippet of a LinkedIn output converted to RDF is shown below:

   ```
   <http://data.bt.com/id/external/linkedin/21379>
      a <http://rdfs.org/sioc/ns#User> ;
      <http://rdfs.org/sioc/ns#member_of>
          <http://www.linked.com> ;
      <http://rdfs.org/sioc/ns#link>
      <http://www.linkedin.com/profile/view
        ?id=21379>;
      <http://xmlns.com/foaf/0.1/familyName>
          "Coran"^^xsd:string ;
      <http://xmlns.com/foaf/0.1/givenName>
         "Ral"^^xsd:string ;
      <http://www.w3.org/ns/org#memberOf>
         "Founder & CEO"^^xsd:string ;
      <http://www.w3.org/ns/org#location>
         "London, United Kingdom"^^xsd:string ;
      <http://data.bt.com/def/industryWorkingIn>
         "Online Media"^^xsd:string; .
   ```

2. the join points used for opencorporates.com are the companies (i.e. the accounts), which the SalesForce team members are in charge of. These companies' data are retrieved from Opencorporates (via their API), using the appropriate Adapter, and setting the appropriate sameAs links between the Account class from the internal dataset, and information of the company given by OpenCorporates. An RDF snippet is shown below:

   ```
   <https://opencorporates.com/companies/gb/
   00607154> rdf:type schemaorg:Organization ;
     owl:sameAs
       <http://data.bt.com/id/staff/
            salesforce/directory/ds1/13/
              Account/0012000000SiKgP> ;.
   ```

   These are stored in the Owlim RDF store.

   In order to get more information of the companies involved, we need to get the Linked Data format information of these companies. There are a variety of choices of storing these data. Two of them include to get the company information at the time when the data is running or to cache this information to be ready before run-time. We chose the latter for performance

[7] Thanks to [https://www.linkedin.com/in/barrynorton(Barry Norton)] for this advice.

reasons and in order to insulate us from any thing that may happen to the site while applications are running. There is a policy in place to refresh these data at recurrent intervals.

6. DATA CONSUMPTION AND QUERY FACILITIES

As described in subsection 3.1, we provide a component which allows sales leads and managers to interact with the application. One advantage of Linked Data consumption is the ability to view disparate datasets of differing shapes with the same lens. This brings user interface consistency and the ability for the user to quickly understand what they are being shown. This has been made possible because of the usage of Linked Data. Giving URIs to all resources and generating user-understandable labels for them helps the client interface to generate a consistent view for these resources.

Because the Unified Ontology layer (as described in section 5) links together disparate data sources, it provides a platform where more complicated queries can be asked of the system. Some of these queries are listed below.

Query Example 1.

```
select distinct ?btEmployee ?linkedInFriend  {
?btEmployee  a def:EmployeeSocialMediaUser .
?btEmployee  rel:knowsOf ?linkedInFriend .
?linkedInFriend a <http://rdfs.org/sioc/ns#User> .
?linkedInFriend <http://www.w3.org/ns/org#location>
  ?whereLocated .
FILTER regex(?whereLocated, "Germany") .
}
```

Query Example 1 asks for first- and second-order connections of BT employees on LinkedIn that live in Germany. The text “Germany” is passed to the query by the client interface.

Query Example 2.

```
select distinct ?btEmployee ?linkedInFriend  {
?btEmployee  a def:EmployeeSocialMediaUser .
?btEmployee  rel:knowsOf ?linkedInFriend .
?linkedInFriend a <http://rdfs.org/sioc/ns#User> .
?linkedInFriend <http://www.w3.org/ns/org#location>
  ?whereLocated .
?linkedInFriend def:industryWorkingIn
  ?industryWorkingIn .
FILTER regex(?whereLocated, "Germany") .
FILTER regex(?industryWorkingIn,
    "Information Technology").
}
```

Query Example 2 asks for the connections that are in the Information Technology industry and live in Germany.

Query Example 3.

```
select distinct ?btEmployee ?linkedInFriend  {
?btEmployee  a def:EmployeeSocialMediaUser .
?btEmployee  a def:SalesForceUser .
?btEmployee  def:hasRoleName ?hasRoleName .
FILTER ( str(?hasRoleName) = "Sales Manager" )
?btEmployee  rel:knowsOf ?linkedInFriend .
?linkedInFriend a <http://rdfs.org/sioc/ns#User> .
?linkedInFriend def:industryWorkingIn
  ?industryWorkingIn .
FILTER regex(?industryWorkingIn, "Oil & Energy") .
}
```

Query Example 3 asks for Sales Managers within the BT SalesForce team whose LinkedIn connections work in the oil industry. The texts “Sales Manager” and “Oil & Energy” are passed to the query by the client interface.

Query Example 4.

```
select distinct ?btEmployee ?linkedInFriend
   ?company ?companyAddress {
?btEmployee  a def:EmployeeSocialMediaUser .
?btEmployee  a def:SalesForceUser .
?btEmployee  def:hasRoleName ?hasRoleName .
FILTER ( str(?hasRoleName) = "Sales Manager" )
?btEmployee  rel:knowsOf ?linkedInFriend .
?linkedInFriend a <http://rdfs.org/sioc/ns#User> .
?linkedInFriend def:industryWorkingIn
   ?industryWorkingIn .
FILTER regex(?industryWorkingIn, "Oil & Energy") .
?linkedInFriend  <http://www.w3.org/ns/org#memberOf>
  ?companyFriendWorksIn .
?company a rov:RegisteredOrganization .
?company skos:prefLabel ?companyName .
FILTER regex(?companyName, ?companyFriendWorksIn) .
?company org:siteAddress ?addr .
?addr
 <http://www.w3.org/2006/vcard/ns#extended-address>
        ?companyAddress .
}
```

This query projects out the companies and addresses where these LinkedIn connections work.

7. GENERAL PRINCIPLES AND CONCLUSION

This paper described how we have used Ontology Based Data Management (OBDM) and Linked Data to integrate enterprise data with open data and social media. When linking external data with enterprise data, we found that there is a need for a strong business case. This enables buy-in from the stake-holders, and focusses attention on the type and quality of external data needed. Starting with a simple ontology is important, this can be extended as requirements change. When integrating disparate systems, there may be concepts that are equivalent or similar in these systems. These are co-references. Clarity of the type of equivalences to set amongst these is important, and an effective solution for their management needs to be in place. When bringing in external data, we found that a policy needs to be in place whether these external data will be brought in at run-time and/or cached. If during run-time, mitigating policies against external data un-availability are needed, and if data are cached, cache replacement policies need to be in place. The ability to give universal resource identifiers to all the resources has been useful for data consumption. All related items for a particular resource can be aggregated in a place making it easy to query and to interact with. For future work, we will be extending the system to bring in additional resources both from the enterprise and externally.

8. ACKNOWLEDGMENTS

This work is supported under SOCIAM: The Theory and Practice of Social Machines. The SOCIAM Project is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/J017728/1 and comprises the Universities of Southampton, Oxford, and Edinburgh.

9. REFERENCES

[1] Nandana Mihindukulasooriya, Raul Garcia-Castro, Miguel Esteban Gutierrez: *Linked Data Platform as a novel approach for Enterprise Application Integration.* in "ISWC 2013 Workshop on Consuming Linked Data".

[2] Philipp Frischmuth, Jakub Klimek, Soren Auer, Sebastian Tramp, Jorg Unbehauen, Kai Holzweig, Carl-Martin Marquardt: *Linked Data in Enterprise Information Integration.* in "Semantic Web Journal, 2012".

[3] Chris Clarke and Fiona Greig: *A Linked Open Data Resource List Management Tool for Undergraduate Students, June 2009*, available at http://www.w3.org/2001/sw/sweo/public/UseCases/Talis/ (April 2014).

[4] Timo Borst and Joachim Neubert: *Publishing STW Thesaurus for Economics as Linked Open Data, June 2009*, available at http://www.w3.org/2001/sw/sweo/public/UseCases/ZBW/ (April 2014).

[5] Arnaud Le Hors and Steve Speicher: *Open Services Lifecycle Collaboration framework based on Linked Data, November 2012*, available at http://www.w3.org/2001/sw/sweo/public/UseCases/IBM/ (April 2014).

[6] Agustina Buccella, Alejandra Cechich, Nieves R. Brisaboa: *Ontology-Based Data Integration.* in "Encyclopedia of Database Technologies and Applications 2005: 450-456".

[7] Michel Gagnon: *Ontology-Based Integration of Data Sources.* in "10th International Conference on Information Fusion, 2007".

[8] Natalia Antonioli, Francesco Castano, Cristina Civili, Spartaco Coletta, Stefano Grossi, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Domenico Fabio Savo, and Emanuela Virardi: *Ontology-Based Data Access: The Experience at the Italian Department of Treasury.* in "CAiSE Industrial Track, volume 1017 of CEUR Workshop Proceedings, page 9-16. CEUR-WS.org, (2013)".

[9] Harry Halpin, Ivan Herman, and Pat Hayes: *When owl:sameAs isn't the Same: An Analysis of Identity Links on the Semantic Web.* Available at http://www.w3.org/2009/12/rdf-ws/papers/ws21.