COMBINING INFORMATION EXTRACTION AND DATA INTEGRATION IN THE ESTEST SYSTEM

Dean Williams
School of Computer Science and Information Systems, Birkbeck College, University of London
Malet Street, London WC1E 7HX, U.K.

Alexandra Poulovassilis
School of Computer Science and Information Systems, Birkbeck College, University of London
Malet Street, London WC1E 7HX, U.K.

Keywords: Information Extraction, Data Integration, ESTEST.

Abstract: We describe an approach which builds on techniques from Data Integration and Information Extraction in order to make better use of the unstructured data found in application domains such as the Semantic Web which require the integration of information from structured data sources, ontologies and text. We describe the design and implementation of the ESTEST system which integrates available structured and semi-structured data sources into a virtual global schema which is used to partially configure an information extraction process. The information extracted from the text is merged with this virtual global database and is available for query processing over the entire integrated resource. As a result of this semantic integration, new queries can now be answered which would not be possible from the structured and semi-structured data alone. We give some experimental results from the ESTEST system in use.

1 INTRODUCTION

The Semantic Web requires us to be able to integrate information from a variety of sources, including unstructured text from web pages, semi-structured XML data, structured databases, and metadata sources such as ontologies. Other applications exist which also need to make used of heterogeneous data that is structured to varying degrees as well as related free text. For example, in UK Road Traffic Accident reports data in a standard structured format is combined with free text accounts in a formalised subset of English; in crime investigation operational intelligence gathering, textual observations are associated with structured data relating to people and places; and in Bioinfomatics structured databases such as SWISS-PROT (Bairoch et al., 2000) include comment fields containing related unstructured information.

Data integration systems provide a single virtual global schema over a collection of heterogeneous data sources that facilitates global queries across the sources (Lenzerini, 2002; Halevy, 2003). Such systems are able to integrate data occurring in a variety of structured and semi-structured formats but, to our knowledge, they have not so far attempted to include unstructured text. In Information Extraction (IE) systems, pre-defined entities are extracted from text and this data fills slots in a template using shallow NLP techniques (Appelt, 1999). Data integration and IE are therefore complementary technologies and we argue that a system that combines them can provide a basis for applications that need to integrate information from text as well as structured and semi-structured data sources. Our ESTEST system integrates the schemas of structured data sources with ontologies and other available semi-structured data sources, creating a virtual global schema which is then used as the template and a source of named entities for a subsequent IE phase against the text sources. Metadata from the data sources can be used to assist the IE process by semi-automatically creating the required input to the IE modules. The templates filled by the IE process will result in a new data source which can be integrated with the virtual global schema. The resulting extended virtual global database can subsequently used for answering global queries which could not have been answered from the structured and semi-structured data alone.

The rest of this paper is structured as follows: Section 2 describes the architecture of ESTEST and use of existing data integration and IE software. Section 3 give the design and implementation of our ESTEST system, alongside an example that illustrates its usage. Section 4 presents results from initial experiments in the Road Traffic Accident application domain. Section 5 compares and contrasts our approach with related work. Finally, in Section 6 we give our conclusions and plans for future work.

2 BACKGROUND

The ESTEST *Experimental Software to Extract Structure from Text* system is implemented as a layer over the AutoMed (AutoMed Project, 2006) data integration toolkit and the GATE (Cunningham et al., 2002) IE framework, making use of their facilities. We now briefly describe GATE and AutoMed.

AutoMed. In data integration systems, several data sources, each with an associated schema, are integrated to form a single virtual database with an associated global schema. Data sources may conform to different data models and therefore need to be transformed into a common data model as part of the integration process. AutoMed is able to support a variety of common data models by providing graph-based metamodel, the Hypergraph Data Model (HDM). AutoMed provides facilities for specifying higher-level modeling languages in terms of this HDM e.g. relational, entity-relational, XML. These specifications are stored in AutoMed's Model Definitions Repository (MDR). A generic wrapper for each data model is provided, with specialisations for interacting with specific databases or repositories e.g. a set of relational wrappers for interacting with the common relational DBMS. The schemas of such data sources can be extracted by the appropriate wrapper and are stored in AutoMed's Schemas & Transformations Repository (STR). AutoMed provides a set of primitive transformations that can be applied to schemas e.g. to add, delete and rename schema constructs. AutoMed schemas are therefore incrementally transformed and integrated by sequences of such transformations (termed *pathways*).

Add and delete transformations are accompanied by a query (expressed in a functional query language, IQL (A.Poulovassilis, 2004)) which specifies the extent of the added or deleted construct in terms of the rest of the constructs in the schema. These queries can be used to translate queries or data along a transformation pathway (McBrien and Poulovassilis, 2003). In particular, queries expressed in IQL can be posed on a virtual integrated schema, are reformulated by AutoMed's Query Processor into relevant sub-queries for each data source, and are sent to the data source wrappers for evaluation. The wrappers interact with the data sources for the evaluation of these sub-queries, and with the Query Processor for post-processing of sub-query results.

The queries supplied with primitive transformations also provide the necessary information for these transformations to be automatically *reversible*, e.g. an add transformation is reversed by a delete transformation with the same arguments. AutoMed is therefore defined as a *both-as-view (BAV)* data integration system in (McBrien and Poulovassilis, 2003) which gives an in-depth comparison of BAV with the other major data integration approaches, Global-As-View (GAV) and Local-As-View (LAV) (Lenzerini, 2002).

One of the main advantages of using AutoMed for ESTEST rather than a GAV or LAV-based data integration system is that, unlike GAV and LAV systems, AutoMed readily supports the evolution of both source and integrated schemas. This is because it allows transformation pathways to be extended, so that if a new data source is added, or if a data source or integrated schema evolves, then the entire integration process does not have to be repeated and instead the schemas and transformation pathways can be 're-paired'.

As a pre-requisite for the development of ESTEST we have made two extensions to the AutoMed toolkit which were required for ESTEST but which are also more generally applicable. Firstly, we have extended AutoMed to include support for data models used to represent ontologies. We have modeled RDF (Lassila and Swick, 1999) graphs and associated RDFS (Brickley and Guha, 2004) schemas in the HDM. A corresponding AutoMed wrapper for such data sources has been implemented using the JENA API (McBride, 2002). Although only RDF/S data sources are currently supported, the use of JENA means that additional ontology models can easily be added as specialisations of the current wrapper, similarly to the specialised AutoMed relational wrappers for specific RDMBS. Secondly, we have a developed the AutoMed HDM Store, a native HDM repository that is used for storing the data that is extracted by ESTEST from text sources.

GATE. The GATE system (Cunningham et al., 2002) provides a framework for building IE applications. It includes a wide range of standard components and also allows the integration of bespoke components. Applications are assembled as pipelines of components which are used to process collections of documents. Applications can be built and run either as standalone Java programs or through the GATE GUI. A pattern matching language called *JAPE* (Cunningham et al., 2000) is provided for constructing application specific grammars. Some standard JAPE grammars are provided with GATE e.g. for finding names of people in text. The result of running an application is a collection of annotations over a text. For example in the string "RAN IN FRONT OF BUS" an annotation might state that from the seventeenth to nineteenth character there is a reference to a public service vehicle.

3 THE ESTEST SYSTEM

The ESTEST system supports an incremental approach to integrating text with structured and semi-structured data sources, whereby the user iterates through a series of steps as new information sources need to be integrated and new query requirements arise. The ESTEST approach is described in (Williams and Poulovassilis, 2004; Williams, 2005). This paper describes an implementation of that approach, as well as giving some experimental results.

We use as our running example, a simple example application based on Road Traffic Accident reports. In the UK, accidents are reported using a format known as STATS-20 (UK Department for Transport, 1999). In STATS-20, a record exists for each accident, following which there are one or more records for the people and the vehicles involved in the accident. The majority of the schema consists of coded entries, and detailed guidance as to what circumstances each of the codes should be used accompanies these. A textual description of the accident is also reported, expressed in a stylised form of English. An example of the textual description collected for a specific accident might be “FOX RAN INTO ROAD CAUSING V1 TO SWERVE VIOLENTLY AND LEAVE ROAD”, where “V1” is short for “vehicle 1” and is understood to be the vehicle which caused the accident. The schema of the structured part of the STATS-20 data is well designed and there have been a number of revisions during its several decades of use. However, there are still queries that cannot be answered via this schema alone.

In our running example we suppose that an analysis of the road traffic accidents caused by animals is required, including the kind of animal causing the obstruction.

Figure 1 illustrates the main components of the ESTEST system and each of these is described in turn in sections 3.1 to 3.5 below.

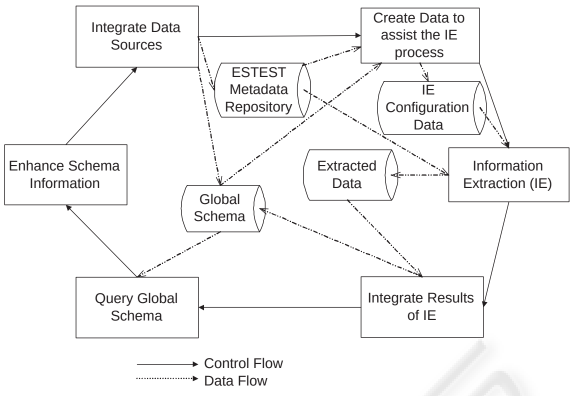

Figure 1: Overview of the ESTEST system.

3.1 Integrate Data Sources

The *ESTEST Integrator* component is configured with the collection of data sources available to the user. Each of these has an associated *ESTEST Wrapper* instance (there is an ESTEST Wrapper for each data model supported by AutoMed). The ESTEST Wrapper makes use of the corresponding AutoMed wrapper in order to construct a data source schema within the AutoMed STR. However, the ESTEST Wrapper is also able to extract additional metadata from the data source, which is stored in the *ESTEST Metadata Repository* — see Figure 1. The ESTEST wrapper also transforms the AutoMed representation of a data source schema into the ESTEST data model described below.

The first step for the Integrator is to iterate through the data sources and use the associated wrapper to construct an initial schema within the AutoMed STR.

In our example, two data sources are assumed to be available: AccDB is a relational database holding the relevant STATS-20 data and AccOnt is a user-developed RDF/S ontology concerning the type of obstructions which cause accidents. Figure 3 shows the AccOnt RDFS schema and some associated RDF triples. The AccDB database consists of three tables:

```
accident(acc_ref,road,road_type,
  hazard_id, acc_desc)
vehicle(acc_ref,veh_no,veh_type)
carriageway_hazards(hazard_id,
  hazard_desc)
```

In the accident table, each accident is uniquely identified by an acc_ref, the road attribute identifies the road the accident occurred on, and road_type indicates the type of road. The hazard_id contains the carriageway hazards code and this is a foreign key to the carriageway_hazards table. We assume that the multiple lines of the text description of the accident have been concatenated into the acc_desc column. There may be zero, one or more vehicles associated with an accident and information about each them is held in a row of the vehicle table. Here veh_reg uniquely identifies each vehicle involved in an accident and thus acc_ref,veh_no is the key of this table. The Integrator calls on the wrappers to create a relational schema for AccDB and RDF/S schema for AccOnt.

The data sources are each now converted from their model specific representation into the *ESTEST data model*. The table below shows the constructs of the ESTEST data model and their representation in the HDM. We see that the ESTEST model provides *concepts* which are used to represent anything that has an extent i.e. instance data. Concepts are represented by HDM nodes e.g. ⟨⟨fox⟩⟩, ⟨⟨animal⟩⟩, and are structured into an isA hierarchy e.g. ⟨⟨isA, fox, animal⟫. Concepts can have *attributes* which are represented by a node and an unnamed edge in the HDM e.g. the attribute ⟪animal, number_of_legs⟫. We note that in AutoMed's IQL query language instances of modeling constructs within a schema are uniquely identified by their *scheme*, enclosed within double chevrons, ⟪ ... ⟫.

ESTEST Data Model

Construct	**HDM Representation**
Concept:⟪c⟫	Node: ⟪c⟫
Attribute:⟪c,a⟫	Node: ⟪a⟫ Edge: ⟪_,c,a⟫
isA: ⟪isA,c1,c2⟫	Constraint ⟪c1⟫ ⊆ ⟪c2⟫

The two data source schemas in our example are automatically converted into their equivalent ESTEST representation when their ESTEST wrapper is called by the Integrator. The representation of AccDB is shown in Figure 2 and that of AccOnt in Figure 3.

Similarly, the Integrator now calls each ESTEST wrapper to collect metadata from its data source. This metadata is used to assist in finding correspondences and for suggesting to the user which schema constucts could be of use in the later IE step (see section 3.3 below). As well as type information, Word Forms and Abbreviations are collected as described below. This metadata is stored in the ESTEST Metadata Repository.

Word Forms are words or phrases which represent a concept. Word forms associated with concepts are important in ESTEST because of their use in the IE process. The ambiguity inherent in natural language means that word forms can be associated with many concepts. The, often limited, textual clues available in the data source schemas are used by ESTEST to find word forms. These clues include schema object identifiers and comment features such as the 'remarks' supported by the JDBC database API. In database schemas, a number of informal naming conventions are often seen, for example naming database columns "acc_identifier" or "accIdentifier". The Integrator recognises a number of these conventions and breaks identifiers returned by the wrappers into their component parts.

Abbreviations are often used in database schema object identification, and the Integrator has an extendable set of heuristics for parsing common naming conventions e.g. a rule exists for identifying when an abbreviation of the table name is prefixed onto the column names in a relational table e.g. the abbreviation "acc" in the column "acc_ref" of the "accident" table used in the example. The user can also enter abbreviations commonly used in the application domain. Abbreviations are combined with full word forms to generate possible combinations.

There are a number of alternative sources for expanding the word forms for a concept when required: manually entered, from database description metadata, from lower down the concept isA hierarchy, or from the WordNet (Fellbaum, 1998) natural language ontology. The Integrator is able to use these to expand the number of word forms for a concept as required. A confidence measure is assigned to the word form depending on its source.

Type metadata is also extracted from the data sources and is used to suggest which schema constructs might be text to be processed by the IE component or used as sources of named entities. Named entity recognition is one of the core tasks in information recognition and involves looking up flat-file lists containing the known instances of some type. ESTEST extends this by identifying concepts in the database schema that are likely to be sources of named entity types and uses either their extent or the word forms associated with the concept for the ESTEST named entity recognition component.

In our running example, abbreviations are suggested e.g. "acc" for "accident", "veh" for "vehicle". The Integrator generates alternative word forms for the schema names and presents them to the user e.g. for the schema construct ⟪accident, acc_ref⟫ the associated word forms are "accident reference", "accident" and "reference". The user is now asked if they wish to expand the current word forms of each concept. Suppose that for now only the concept ⟪animal⟫ is selected to be expanded and solely from the schema. In this case the word forms "cat" and "fox" are added to ⟪animal⟫ from the isA hierarchy.

In order to complete the integration and develop the global schema, the Integrator next attempts to find correspondences between concepts in different source schemas using the gathered metadata. The user can accept or amend the list of suggested equivalent schema objects. The corresponding schema constucts are now renamed to have the same name. Using the facilities provided by AutoMed, each of the ESTEST model schemas is incrementally transformed until they each contain all the schema constucts of all the source schemas — these are the union schemas shown in Figure 5. An arbitrary one of these schemas is designated finally as the *global schema*. In our simple example just "accident" from the domain ontology and "accident" from the schema is suggested as a correspondence by the Integrator and is accepted by the user. No additional manual correspondences are supplied and the initial global schema shown in Figure 4 is created. Finally, as the later IE process will require a repository to store the extracted information, an additional data source is created for this purpose by the Integrator. The schema of this new data source is the HDM representation of the ESTEST global schema just derived. This new data source is integrated into the global schema in the same way as the other data sources.

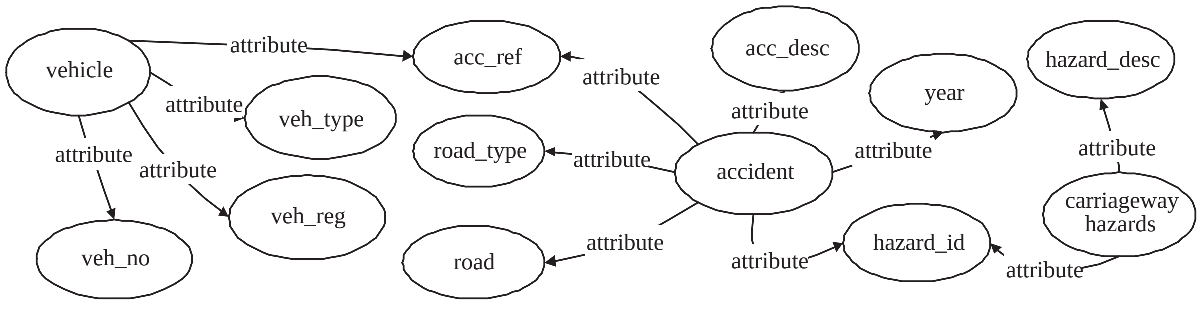

Figure 2: The AccDB schema represented in the ESTEST model.

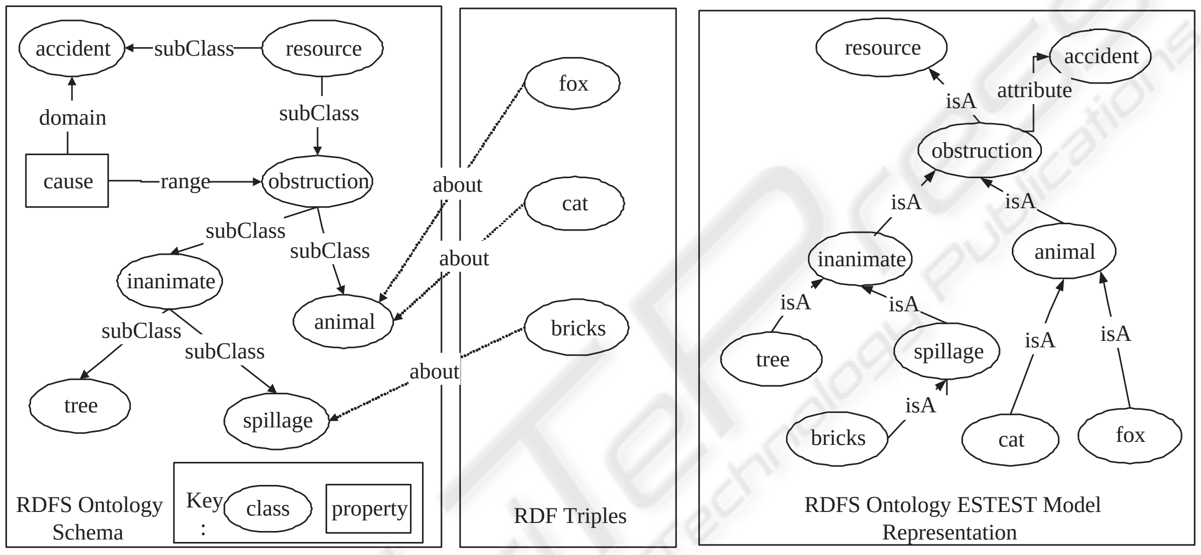

Figure 3: AccOnt Ontology and its representation in the ESTEST model.

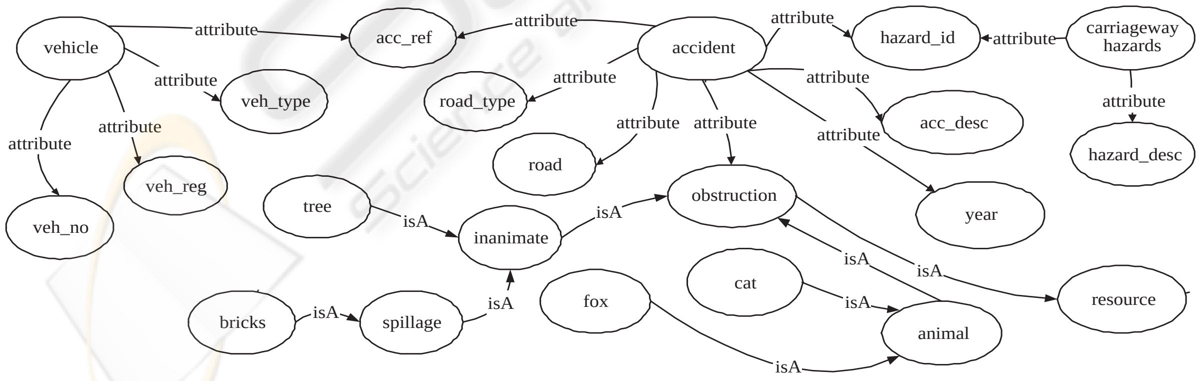

Figure 4: Initial ESTEST global schema.

For our example, the resulting network of transformed and integrated schemas is shown in Figure 5 — for each data source there is now an AutoMed schema and correspondences between schema constructs in the data sources have been identified. There is a pathway to an equivalent ESTEST model schema and then on to a union schema. Any one of these union schema can be used as the global schema for

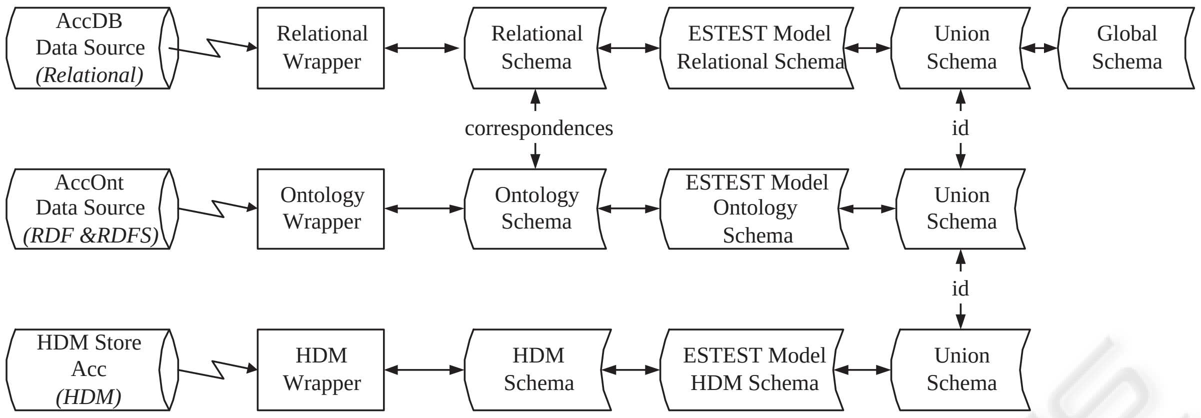

Figure 5: AutoMed Schema Network for the Road Traffic Accident Example.

the entire network.

3.2 Create Metadata to Assist the IE Process

The *ESTEST Configuration* component now makes use of the global schema and collected metadata to suggest basic information extraction rules, such as macros for named entity recognition, and to create templates to be filled from concepts in the schema which have attributes of unknown value. In IE systems, templates are filled by annotations over text (the template slots). ESTEST extends this notion of templates in a number of ways. Firstly, some slots in a template are pre-filled from the structured part of the source data and these values are used to attempt to disambiguate when multiple annotations are suggested over the same text. Secondly, ESTEST makes use of type information contained in the metadata to validate suggested annotations. Finally, ESTEST extends the idea of the relationship that exists between annotations. Normally this only goes as far as linking slots to templates. There is Annotation Schema idea in GATE but this is restricted to defining the annotation features which are allowed and is used to drive the manual entry of annotations. In our system, we link annotation types to concepts in the global schema. In this way the structural relationships between annotation types can be used and extracted annotations can be linked to related instances of other concepts in the source data.

The user confirms or amends the automatically produced configuration. In our example, the user selects just ⟨⟨animal⟩⟩ from the list of suggested named entity sources. the Configuration component identifies the ⟨⟨accident⟩⟩ concept as a template. As the ⟨⟨obstruction⟩⟩ concept has no extent, the IE step will attempt to find values for this. A macro for ⟨⟨animal⟩⟩ is created:

```
Macro: ANIMAL
  ({Lookup.minorType == animal })
```

and a stub JAPE rule for ⟨⟨obstruction⟩⟩. The user enhances this stub as follows:

```
Rule: OBSTRUCTION1
(({Token.string == "RUNS"} |
  {Token.string == "WALKS"}|
  {Token.string  == "JUMPS"})
  (SPACE)?
  ({Token.string == "INTO"} |
  {Token.string == "ONTO"} |
  {Token.string  == "IN FRONT OF"} )
  (ANIMAL) ) :obstruction -->
        :obstruction.obstruction =
        {kind = "Obstruction",
                  rule = "OBSTRUCTION1"}
```

3.3 Information Extraction Component

Now the GATE-based ESTEST IE component is run. The following standard components GATE are configured and assembled into a pipeline: *Document Reset* which ensures the document is reset to its original state for reruns; the *English Tokeniser* splits text into tokens such as strings and punctuation; *Sentence Splitter* divides text into sentences; the *JAPE Processor* takes the grammar rules developed above and applies them to the text. Also used is our bespoke *Schema Gazetteer* component which extends the gazetteers used in the Named Entity recognition core IE task. The annotation type is linked to a concept in the global schema and instances are found either by presenting a query to the global schema for the extent concept across the data sources, or the word forms previously associated with the concept are obtained.

To illustrate, suppose in our example the AccDB database contains just three accidents with the following descriptions:

AccDB Accident Descriptions	
Acc Ref	**Description**
A001234	FOX RUNS INTO ROAD CAUSING V1 TO SWERVE VIOLENTLY AND LEAVE ROAD OFFSIDE
A005678	UPPERTON ROAD LEICESTER JUNCTION SYKEFIELD AVENUE V1 TRAV SYKEFIELD AVE FAILS TO STOP AT XRDS AND HITS V2 TRAV UPPERTON RD V2 THEN HITS V3 PKD ON OS OF UPPERTON RD
A009012	ESCAPED KANGAROO JUMPS IN FRONT OF V1

When the IE component has run the following annotations are found:

Annotations			
Annotation	**Start**	**End**	**Literal**
ANIMAL	1	3	FOX
OBSTRUCTION	1	13	FOX RUNS INTO ROAD

3.4 Integrate Results of IE

The data extracted from the text is now stored in the HDM repository. In our example an instance id is generated "#1" and new instances of HDM nodes ⟨⟨fox⟩⟩, ⟨⟨animal⟩⟩ and ⟨⟨obstruction⟩⟩, are added with value [#1]. Edges ⟨⟨isA,fox,animal⟩⟩ and ⟨⟨isA,animal,obstruction⟩⟩ have the value of the pair {#1,#1}. Edge ⟨⟨attribute,accident,obstruction⟩⟩ has value {A001234,#1}. Queries on the global schema now include the new fact that a fox caused accident A001234.

3.5 Remaining ESTEST Phases

The user can now pose queries to the global schema the results of which will include the new data extracted from the text. The global schema may subsequently be extended by new data sources being added, or new schema constucts identified and added to it. The user may also choose to expand the number of word forms associated with schema concepts. Following any such changes, the process is then repeated and new data extracted from the text.

In our example, suppose the user suspects that the results may not be complete and expands the word forms from WordNet for ⟨⟨animal⟩⟩. The Integrator suggests a mapping between this schema object and a WordNet *synset* (a set of word forms with the same meaning) which the user confirms. Word forms are then obtained by descending the WordNet hypernym relations. The list obtained includes "kangaroo" and as a result re-running the Configuration and IE components now produces similar additional annotations for the third accident report, and an additional fact in the HDM store of KANGAROO as an obstruction for accident A009012.

4 EXPERIMENTS WITH ROAD TRAFFIC ACCIDENT DATA

There are a number of variables which affect the performance of the ESTEST system including: the availability of structured data sources relating to the text, the degree of similarity between the text instances, the amount of effort spent by the user configuring the system to the specific application domain, and the domain expertise of the user. In order to provide some initial confirmation of the potential of our approach we have experimented with a data set of Road Traffic Accident reports consisting of 1658 accident reports. These were six-months worth of reports from one of Britain's 50 police forces. We identified five queries of varying complexity which cannot be answered by the STATS-20 structured data alone. These queries are shown in the table below.

RTA Queries	
Q1	Which accidents involved red traffic lights?
Q2	How many accidents took place 30-50m of a junction?
Q3	How many accidents involve drunk pedestrians?
Q4	Which accidents were caused by animals?
Q5	How many resulted in a collision with a lamppost?

The ESTEST system was configured using a randomly chosen set of 300 reports.

We then ran the system over the remaining 1358 unseen reports and compared the results to a subsequent manual examination of each report. The table below shows these results in terms of the actual number of relevant reports, the recall (the number of correctly identified reports as a percentage of all the correct reports) and the precision (the number of correctly identified reports as a percentage of all the identified reports).

Results of RTA Query Experiments			
Query	**Relevant Reports**	**Recall**	**Precision**
Q1	25	84%	95%
Q2	89	99%	99%
Q3	9	78%	100%
Q4	14	86%	86%
Q5	15	88%	93%

Where appropriate the queries combined structured and text results, for example STATS-20 structured data includes a flag used to indicate whether the accident took place within 20 meters of a junction but does not reveal the distance otherwise. In the query ”How many accidents took place 30-50m of a junction?” accidents with the 20 meters flag set were discarded and only the remaining reports with relevant IE results considered. Configuring the system took 5 hours to develop a domain ontology and enhancing the generated stub rules for the IE process. These results are promising even with the short time spent configuring the system by a non-domain expert user.

5 RELATED WORK

An alternative to rule-based IE is text mining, in which some NLP process creates a structured data set from the text and then this is mined in order to discover patterns in the structured data (A.H.Tan, 1999). Our ESTEST system, in contrast, is driven by specific new querying requirements and the potential of making use of new data sources. None the less, a text mining extension might be a useful addition to ESTEST for some application domains e.g. the SWISS-PROT database, while (U.Y. Nahm, 2000) has shown the potential for combining IE and Text Mining in general.

Recent developments in IE have included moves to support the challenges of knowledge management (Cunningham et al., 2005) and applications for the semantic web e.g. support for ontologies has recently been added to GATE (Bontcheva et al., 2004). A common starting point for recent research is the limitations of the traditional core Named Entity Recognition task in IE where annotations are assigned a type from a list of types names, and recent work is moving towards the idea of *Semantic Annotation* (Kiryakov et al., 2003) where the annotation is linked to a concept in an ontology. This is similar to the ESTEST approach of linking annotations to a concept in a virtual global schema.

The developers of the Knowledge and Information Management (KIM) platform (Popov et al., 2004) believe that a lightweight ontology that provides structure but few axioms is sufficient for the IE task. Our ESTEST data model can similarly be thought of as a lightweight ontology providing a taxonomy and the facility for concepts to have attributes. KIM is based on an ontology of ‘everything’ (KIMO) pre-defined to include concepts and entities from the common IE tasks; annotations that are found by the IE system are treated separately from the knowledge in the ontology. In contrast, ESTEST develops the global schema from available structured data sources specific to the application, and seeks to expand the instance data and schema incrementally by adding to the previously known data and schema. Also, ESTEST makes use of already known instances of structured data which relate to the specific text being processed in order to assist in finding slot values, and it uses other schema constucts and word forms obtained from metadata as sources for named entities.

Finally, previous work on making use of the text in UK Road Traffic Accident reports used expert knowledge of the sub-language in the reports to code description logic-based grammar rules (Wu and Heydecker, 1998). ESTEST makes use of the structured data as well as the text to answer queries and the approach is more generally applicable to other application domains as it does not depend on a specific restricted sub-language.

6 CONCLUSIONS AND FUTURE WORK

We have described the ESTEST system, which combines techniques from Data Integration and Information Extraction in order to make integrated use of heterogeneous data that is structured to varying degrees as well as related free text. ESTEST does this by extracting metadata from data sources to support their integration into a virtual global schema expressed in the ESTEST data model. This schema and its associated metadata are used to semi-automatically configure an IE process. The newly extracted information from the text is merged into the virtual global database which can then be used to answer new queries that could not have been answered before. The user can extend the global schema, add new data sources, enhance the IE configuration, and rerun as required.

This approach is novel in a number of ways. First, to our knowledge this is the first time that a Data Integration system has been extended to include support for free text. Second, ESTEST uses schema integration techniques to integrate available structured data into a global schema which is used as a light-weight ontology for semantic annotation — this is a realistic application-specific alternative to building ontologies from scratch. Third, ESTEST uses the global schema to semi-automatically configure the IE process, thereby reducing the configuration overhead of the IE process.

We have given a simple example illustrating the operation of ESTEST on Road Traffic Accident reports. Initial experimental results in the same domain indicate the approach is able to increase the utility of text stored alongside structured data and that the system’s performance is at least comparable with standalone IE systems.

Based on this preliminary evaluation we have identified three areas for further enhancement of ESTEST: developing its schema matching component by using IE on the textual schema metadata; exploring identifier disambiguation techniques to assist with the co-referencing task in IE; and improving the support for enhancing the global schema with new structure found in the text.

To investigate how generally applicable our approach is, we will then evaluate the enhanced ESTEST system in the crime investigation and semantic web application domains. In this evaluation, we will consider the results obtained by ESTEST users who are experts in the given application domain. These results will be compared to both any existing approaches (such as manual inspection or keyword search) employed by these expert users and to results obtained by them using a stand-alone IE system.

Finally, in order to support the end user, we will analyse the requirements of a workbench for accessing the ESTEST components and functionality via a graphical user interface.

REFERENCES

A.H.Tan (1999). Text mining: The state of the art and the challanges. *Proc. of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases*, pages 65–70.

A.Poulovassilis (2004). A tutorial on the IQL query language. Technical report, AutoMed Project.

Appelt, D. (1999). An introduction to Information Extraction. *Artificial Intelligence Communications*, 12(3):161–172.

AutoMed Project (2006). http://www.doc.ic.ac.uk/automed/.

Bairoch, A., Boeckmann, B., Ferro, S., and Gasteiger, E. (2000). Swiss-Prot: Juggling between evolution and stability. *Brief. Bioinform.*, 5:39–55.

Bontcheva, K., Tablan, V., Maynard, D., and Cunningham, H. (2004). Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering*, 10:349—373.

Brickley, D. and Guha, R. (2004). RDF vocabulary description language 1.0: RDF schema. *W3C Recommendation*. http://www.w3.org/TR/rdf-schema/.

Cunningham, H., Bontcheva, K., and Li, Y. (2005). Knowledge Management and Human Language: Crossing the Chasm. *Journal of Knowledge Management*, 9(5):108–131.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Cunningham, H., Maynard, D., and Tablan, V. (2000). JAPE: a Java Annotation Patterns Engine (Second Edition). Research memorandum, University of Sheffield.

Fellbaum, C. (1998). WordNet an electronic lexical database.

Halevy, A. (2003). Data Integration: A Status Report. In Weikum, G., Schöning, H., and Rahm, E., editors, *BTW*, volume 26 of *LNI*, pages 24–29. GI.

Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., and Goranov, M. (2003). Semantic Annotation, Indexing, and Retrieval. In *2nd International Semantic Web Conference (ISWC2003)*, pages 484–499.

Lassila, O. and Swick, R. (1999). Resource description framework (RDF) model and syntax specification. *W3C Recommendation*. http://www.w3.org/TR/REC-rdf-syntax/.

Lenzerini, M. (2002). Data Integration: A Theorectical Perspective. In *Proc. PODS02*, pages 247–258.

McBride, B. (2002). Jena: A semantic web toolkit. *IEEE Internet Computing*, 6(6):55–59.

McBrien, P. and A.Poulovassilis (2003). Defining peer-to-peer data integration using both as view rules. In *Proc. Workshop on Databases, Information Systems and Peer-to-Peer Computing (at VLDB'03), Berlin*.

McBrien, P. and Poulovassilis, A. (2003). Data integration by bi-directional schema transformation rules. In *Proc. ICDE'03*, pages 227–238.

Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., and Kirilov, A. (2004). KIM - a semantic platform for information extraction and retrieval. *Nat. Lang. Eng.*, 10(3-4):375–392.

UK Department for Transport (1999). Stats20: Instructions for the completion of road accident report form. http://www.dft.gov.uk.

U.Y. Nahm, R. M. (2000). Using Information Extraction to aid the discovery of prediction rules from text. *Proc. of the KDD-2000 Workshop on text Mining*, pages 51–58.

Williams, D. (2005). Combining data integration and information extraction techniques. In *Proc. Workshop on Data Mining and Knowledge Discovery, at BNCOD'05*, pages 96–101.

Williams, D. and Poulovassilis, A. (2004). An example of the ESTEST approach to combining unstructured text and structured data. In *Proc. of the Database and Expert Systems Applications (DEXA'04)*, pages 191–195. IEEE Computer Society.

Wu, J. and Heydecker, B. (1998). Natural language understanding in road accident data analysis. *Advances in Engineering Software*, 29:599–610.