KMi

Knowledge Media Institute

Integration of Information Extraction with an Ontology

*M. Vargas-Vera, J. Domingue, Y. Kalfoglou,
E. Motta and S. Buckingham Shum*

KMI-TR-109

May, 2001

www.kmi.open.ac.uk/tr/papers/kmi-tr-109.pdf

The Open
University

Integration of Information Extraction with an Ontology

Maria Vargas-Vera, John Domingue, Yannis Kalfoglou,
Enrico Motta and Simon Buckingham Shum
Knowledge Media Institute (KMi),
The Open University,
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
{m.vargas-vera, j.b.domingue, y.kalfoglou, e.motta, s.buckingham.shum}@open.ac.uk

ABSTRACT

This paper describes the integration of an ontology with an information extraction (IE) tool. Our main goal is extract knowledge from text to populate the ontology, and so alleviate the problem of ontology maintenance. The IE tool extracts information using partial parsing and machine learning techniques. Our domain of study is “KMi Planet”, a Web-based news server that helps to communicate relevant information between members in our institute. Currently our system finds instances of classes or subclasses. Although in the future we expect to create new classes and subclasses from new concepts appearing in text.

Keywords

Ontology population, Information extraction

1. INTRODUCTION

The goal of a Information Extraction system (IE) is to extract specific types of information from text. For example, an IE system in the domain of KMi (Knowledge Media Institute) organisation, the system should be able to extract the name of KMi projects, KMi funding organisations, awards, dates, etc. The main advantage of this task is that portions of a text that are not relevant to the domain can be ignored. Therefore IE is less computationally expensive than a Natural Language Processing system. Essentially, IE can be seen as the task of pulling predefined relations from texts. Efforts have been made to apply IE to several domains, for instance, scientific articles such as MEDLINE (it contains abstracts of biomedical journals) [2], bibliographic notices [10], and medical records [14]. Ontologies can be used in IE systems to help them extract relations from semi or unstructured documents, statements or terms [13]. Also, recent work on semi-automatic ontology acquisition by means of IE, supported by machine-learning methods, is described in [7, 6]. In similar lines there is the CMU’s approach for extracting information from hypertext using machine learning techniques (Bayes classifier) and making use of an ontology [1].

Most IE systems use some form of partial parsing to recognise syntactic constructs without generating a complete parse tree for each sentence. Such partial parsing has the advantages of greater speed and robustness. High speed is necessary to apply the IE to a large set of documents. The robustness achieved by allowing useful work to be done from a partial parsing, is essential to deal with unstructured and informal texts (such as the e-mail messages we consider).

In order to build an IE tool we had integrated several components from the University of Massachusetts Amherst (UMass) which are fully described in Rillof[12]. Riloff classifies text using extraction patterns and semantic features associated to slots in a predefined frames [11]. For example, in the MUC’s terrorist domain that she considers, the event “murder” is represented by a following slots victim, perpetrator and weapon. Each of these slots could have assigned a semantic class. This means that the value for each slot is restricted to have a value in that class.

Each of the templates are triggered by the main verb, in this case “murder”, in any tense. Trigger words can be reliably identified using linguistic rules like the ones described in [12]. For example, if the targeted information is the subject or the direct object of a verb then the best trigger word should be the main verb. The sentence is also matched using prepositions.

In particular, the main aim of this paper is to describe the use of template-driven IE to populate an ontology. Our system could be used to update an ontology by finding new instances of the classes defined on the ontology. The template matching itself is supported semantically by referring to the ontology, but also contains some lightweight NLP techniques in order to syntactically identify some *fragments* of the sentences. We believe it is important to mix the syntactic and semantic. The semantic checking is often necessary to resolve ambiguities, for example, ontologies can provide us with axioms of common sense knowledge such “if someone is visiting a place then this someone should be a person.” Conversely, some grammar constructions (such as dates) can be recognized robustly. Overall, our primary contribution is to integrate a template-driven IE engine with an ontology engine to supply the necessary semantic content.

The paper is organised as follows: Section 2 briefly describes our suite of tools in order to give some background. In Section 3 we present a typology of two events as are defined in KMi ontology. Section 4 shows the integration of several components from UMass in order to build our IE tool. Section 5 describes the use of ontology to cope with the ambiguity in the identification of objects in the story. Section 6 shows the OCML [1] code generated after badger obtains all instantiations. Section 7 discusses the process of populating an ontology. Finally, Section 8 gives conclusions and directions for future work.

[1] OCML is a language designed for knowledge modeling

2. THE KMI PLANET SYSTEM AND PLANETONTO

KMi Planet is a Web-based news server that facilities communication between members of KMi. To quote from [5]:

> The authors integrated a suite of tools, called **PlanetOnto** that supports a speedy but high-quality publishing process, allows ontology-driven document formalization and augments standard browsing and search facilities with deductive knowledge retrieval.

Two primary components are the story library and the ontology library. The Story database contains the text of the stories that have been provided to Planet by the journalists. In the case of KMi Planet it contains stories which are relevant to our institute. The Ontology Library contains several existing ontologies, in particular the KMi ontology. OCML is the modeling language which allows the creation of classes and instances in the ontology.

The main architecture of PlanetOnto is shown in Figure 1. In PlanetOnto we identify three types of users: journalists who send stories to KMi planet, knowledge engineers who maintain the Planet ontology, and readers of the Planet stories.

PlanetOnto augmented the basic publish/find scenario supported by KMi planet, and supports the following activities [2].

1. **Story submission**. A journalist submits a story to KMi planet using e-mail text. Then the story is formatted and stored.

2. **Story reading**. A Planet reader browses through the latest stories using a standard Web browser,

3. **Story annotation**. Either a journalist or a knowledge engineer manually annotates the story using Knote (the Planet knowledge editor),

4. **Provision of customised alerts**. An agent called Newsboy builds user profiles from patterns of access to PlanetOnto and then uses these profiles to alert readers about relevant new stories.

5. **Ontology editing**. A tool called **WebOnto** [4] provides Web-based visualisation, browsing and editing support for the ontology. It allows easier development and maintenance of the knowledge models, themselves specified in OCML (Conceptual Modeling Language) [9].

6. **Story soliciting**. An agent called Newshound, periodically solicits stories from the journalists.

7. **Story retrieval and query answering**. The Lois interface supports integrated access to the story archive

Recently, two tools have been integrated in the architecture: **myPlanet** and an **IE tool**.

- **myPlanet** is an extension to Newsboy and helps story readers to read only the stories that are of interest instead of reading all stories in the archive. It uses a manually predefined set of cue-phrases for each of 'research areas' defined in the ontology. For example for genetic algorithms one cue-phrase is "evolutionary algorithms". Consider the following example: if someone is interested in *research* area *Genetic Algorithms*. A search engine will return all the stories that talk about that *research area*. myPlanet (by using the ontological relations) will also find all Projects that have research area Genetic Algorithms and then search for stories that talk about these projects, thus returning them to the reader even if the story text itself does not contain the phrase "genetic algorithms".

- **Information extraction** is a tool which extracts information from e-mail text and it connects with webonto to prove theorems using the deductive capabilities of the KMi ontology. Our IE tool was constructed by integrating and customising three components from UMass (Marmot, Crystal and Badger).

3. EVENT TYPOLOGY

KMi domain consists of events or activities happening in our Institute. This activities (events) are defined formally in our ontology as classes. Currently, in our KMi ontology we have defined 40 different types of events. As the event typology is already defined in the KMi ontology. Then, for each event we already had defined the kind of objects (entities) that could extracted by using the IE tool. Figure 2 shows a portion of the hierarchy of events as defined in KMi ontology.

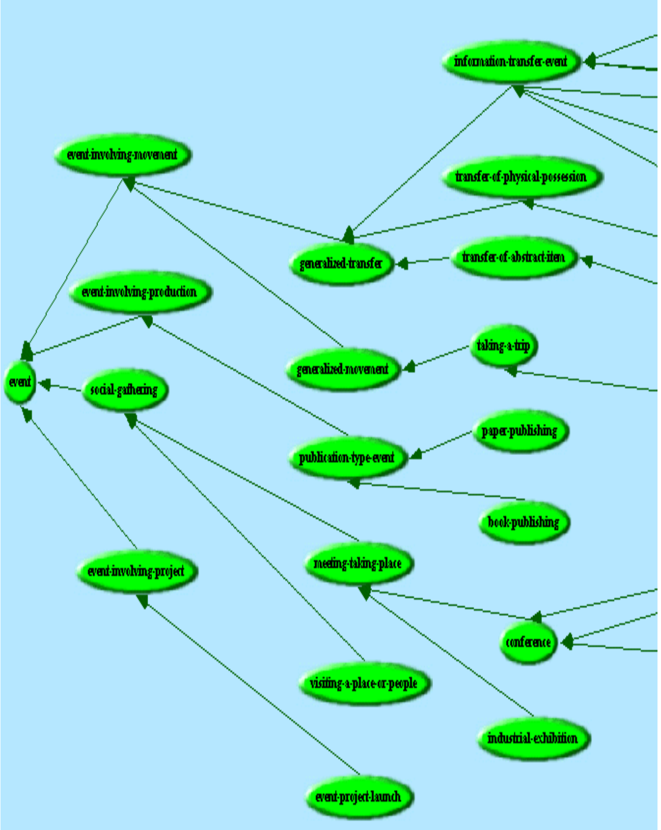


Figure 2: Event hierarchy

For the sake of space we only present the structure of three type of events from the event hierarchy: visiting-a-place-or-people, conferring-a-monetary-award and demonstration-of-technology.

```
Event 1: visiting-a-place-or-people
```

[2] URL:http://kmi.open.ac.uk/projects/planetonto/

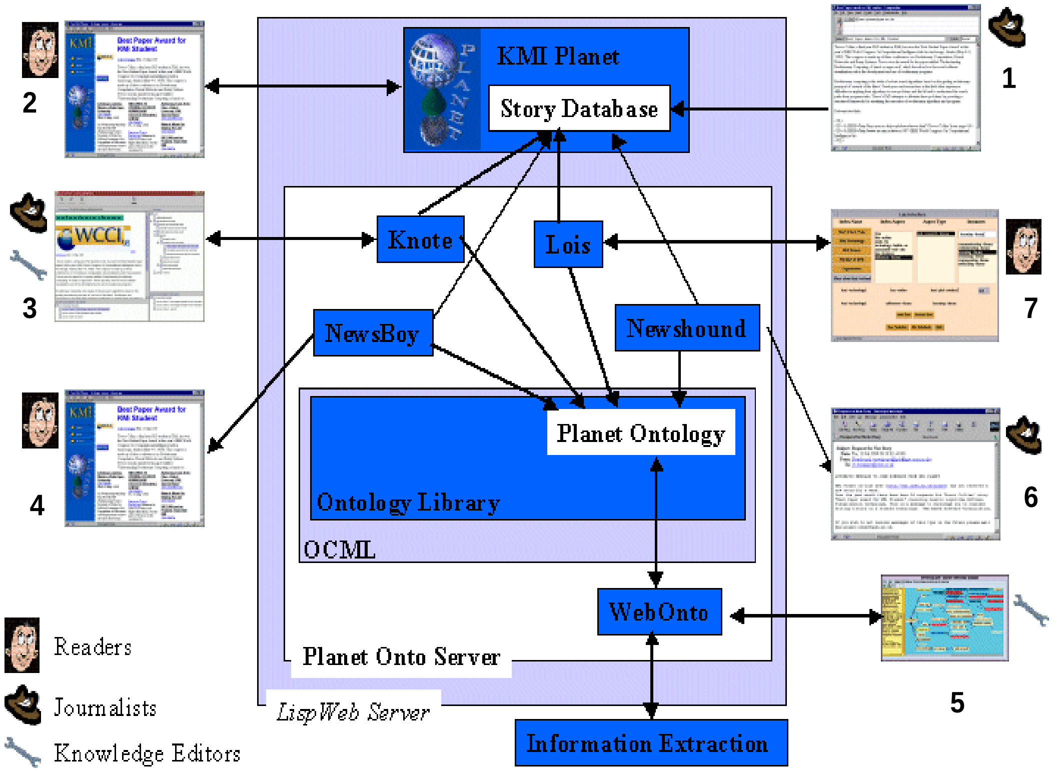


Figure 1: PlanetOnto architecture

```
visitor (list of person(s))
people-or-organisation-being-visited
    (list of person(s) or organization)
has-duration (duration)
start-time (time-point)
end-time (time-point)
has-location (a place)
other agents-involved (list of person(s))
main-agent (list of person(s))
```

The structure of Event 1 (visiting-a-place-or-people) describes a set of object's types which might be encountered in story describing an event visit, such as, visitor, people-or-organisation-being-visited, other agents-involved, etc.

```
Event 2: conferring-a-monetary-award

monetary award (sum of money)
has-duration (duration)
start-time (time-point)
end-time (time-point)
has-location (a place)
main-agent (list of person(s))
other agents-involved (list of person(s))
location-at-start (a place)
location-at-end (a place)
awarding-body (an organization)
has-award-rationale (project goals)
```

In the event 2 the IE system finds all the objects except values of slots which are computed using other values such like **has-duration**. This value is computed by subtracting start-time from end-time. Also, the value for the slot **has-award-rationale** is extracted from text by using heuristics such as if the word **goal** appears in the story then the system will extract as rationale all sentence until it finds full stop. The reason for this is because is to general to be learned by an IE tool. It does not follow any grammar rule about how the rationale could be expressed by a journalist who writes an story describing a project award.

```
Event 3: demonstration-of-technology

technology-being-demostrated (technology)
has-duration (duration)
start-time (time-point)
end-time (time-point)
has-location (a place)
other agents-involved (list of person(s))
main-agent (list of person(s))
location-at-start (a place)
location-at-end (a place)
medium-used (equipment)
subject-of-the-demo (title)
```

Event 5 contains the structure for the event "demonstration-of-technology". Entities that need to be recognised are technology, place, etc.

4. INFORMATION EXTRACTION TOOL

We had built an IE tool by integrating and customising several components such as Marmot, Badger, Crystal, our KMi ontology and our OCML preprocessor which translates the extracted information into OCML code (defined in next section). Currently, our domain of study is "KMi Planet", a Web-based news server that helps to communicate relevant information between members in our institute [3]. The raw input consists of e-mailed stories written by members of the laboratory.

The background for Marmot, Badger and Crystal and customisation of these components is presented in the following subsections.

4.1 Marmot

Marmot (from UMass) is a natural language preprocessing tool that accepts ascii files and produces an intermediate level of text analysis that is useful for IE applications. Sentences are separated and segmented into noun phrases, verb phrases prepositional phrases.

Marmot splits at clausal boundaries (given a generous notion of sentence). In short, Marmot provides an idiosyncratic syntactic analysis which is sufficient for badger to use in applying its extraction rules. Marmot has several functionalities: preprocesses abbreviations to guide sentence segmentation, resolves sentences boundaries, identifies parenthetical expressions, recognises entries from a phrasal lexicon and replace them, recognises dates and duration phrases, performs phrasal bracketing of noun, preposition and adverbial phrases, finally scopes conjunctions and disjunctions.

We had defined our own verbs, nouns, abbreviations and tags in order to apply Marmot to our KMi domain. For the sake of space we would analyse only the first 3 sentences in the story given in Figure 3.

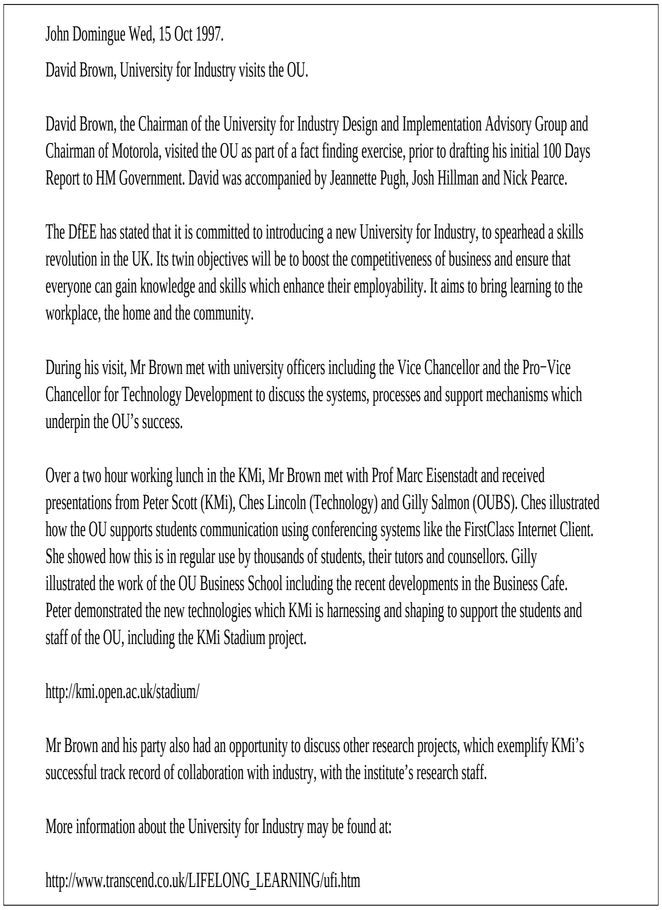
The output from Marmot is shown as follows:

```
<ex> 1 1
SUBJ(1): JOHN DOMINGUE
ADVP(2): @WED_%COMMA%_15_OCT_1997@
PUNC(3): %PERIOD%
</ex>
```

In the first sentence, Marmot recognised two entities firstly a subject (SUBJ) which is JOHN DOMINGUE and secondly a date. The latest is recognised and marked between the symbol "@". Dates are recognised robustly as regular expressions.

```
<ex> 2 1
SUBJ(1): DAVID BROWN %COMMA% UNIVERSITY
PP  (2): FOR INDUSTRY
VB  (3): VISITS
OBJ1(4): THE OU
PUNC(5): %PERIOD%
</ex>
```

In sentence number 2, DAVID BROWN is recognised as subject (SUBJ), a prepositional phrase (PP) "FOR INDUSTRY" is encounter, the verb (VB) VISITS is also found, OBJ1 takes the value of THE OU and finally a punctuation symbol (PUNC) is the full stop is encountered at the end of the sentence.



John Domingue Wed, 15 Oct 1997.

David Brown, University for Industry visits the OU.

David Brown, the Chairman of the University for Industry Design and Implementation Advisory Group and Chairman of Motorola, visited the OU as part of a fact finding exercise, prior to drafting his initial 100 Days Report to HM Government. David was accompanied by Jeannette Pugh, Josh Hillman and Nick Pearce.

The DfEE has stated that it is committed to introducing a new University for Industry, to spearhead a skills revolution in the UK. Its twin objectives will be to boost the competitiveness of business and ensure that everyone can gain knowledge and skills which enhance their employability. It aims to bring learning to the workplace, the home and the community.

During his visit, Mr Brown met with university officers including the Vice Chancellor and the Pro−Vice Chancellor for Technology Development to discuss the systems, processes and support mechanisms which underpin the OU's success.

Over a two hour working lunch in the KMi, Mr Brown met with Prof Marc Eisenstadt and received presentations from Peter Scott (KMi), Ches Lincoln (Technology) and Gilly Salmon (OUBS). Ches illustrated how the OU supports students communication using conferencing systems like the FirstClass Internet Client. She showed how this is in regular use by thousands of students, their tutors and counsellors. Gilly illustrated the work of the OU Business School including the recent developments in the Business Cafe. Peter demonstrated the new technologies which KMi is harnessing and shaping to support the students and staff of the OU, including the KMi Stadium project.

http://kmi.open.ac.uk/stadium/

Mr Brown and his party also had an opportunity to discuss other research projects, which exemplify KMi's successful track record of collaboration with industry, with the institute's research staff.

More information about the University for Industry may be found at:

http://www.transcend.co.uk/LIFELONG_LEARNING/ufi.htm

Figure 3: Email Story

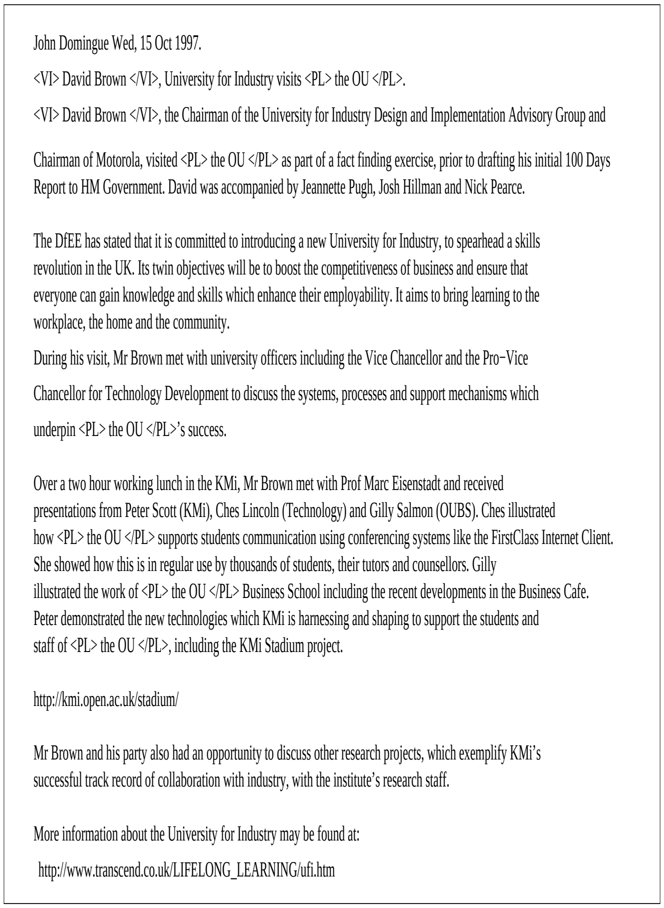
```
<ex> 3 1
SUBJ(1): DAVID BROWN %COMMA% THE CHAIRMAN OF
THE UNIVERSITY
PP  (2): FOR INDUSTRY DESIGN AND IMPLEMENTATION
ADVISORY GROUP AND CHAIRMAN OF MOTOROLA
PUNC(3): %COMMA%
VB  (4): VISITED
OBJ1(5): THE OU
</ex>
```

In the same fashion, in sentence number 3, DAVID BROWN is recognised as subject, the word VISITED is recognised as verb and OBJ1 as THE OU.

4.2 Crystal

A second component that we integrate in our information extraction tool is called Crystal (from UMass). Crystal is a dictionary induction tool. It derives a dictionary of concept node (CN) from a training corpus. The first step in dictionary creation is the annotation of a set of training texts by a domain expert. Each phrase that contains information to be extracted is tagged (with SGML style tags). In order to perform the tagging process, Crystal contains a TextMarker interface called TMI. An example of annotated story is shown in Figure 4.

In our example we had manually annotated our story using TMI. In order to annotate the story we use two tags <VI> and <PL>. Crystal generates case frames that can be used to extract general class of information such as visitor, place, etc. Consequently the case frames should be useful for extracting similar information from future texts.



John Domingue Wed, 15 Oct 1997.

<VI> David Brown </VI>, University for Industry visits <PL> the OU </PL>.

<VI> David Brown </VI>, the Chairman of the University for Industry Design and Implementation Advisory Group and

Chairman of Motorola, visited <PL> the OU </PL> as part of a fact finding exercise, prior to drafting his initial 100 Days Report to HM Government. David was accompanied by Jeannette Pugh, Josh Hillman and Nick Pearce.

The DfEE has stated that it is committed to introducing a new University for Industry, to spearhead a skills revolution in the UK. Its twin objectives will be to boost the competitiveness of business and ensure that everyone can gain knowledge and skills which enhance their employability. It aims to bring learning to the workplace, the home and the community.

During his visit, Mr Brown met with university officers including the Vice Chancellor and the Pro−Vice Chancellor for Technology Development to discuss the systems, processes and support mechanisms which underpin <PL> the OU </PL>’s success.

Over a two hour working lunch in the KMi, Mr Brown met with Prof Marc Eisenstadt and received presentations from Peter Scott (KMi), Ches Lincoln (Technology) and Gilly Salmon (OUBS). Ches illustrated how <PL> the OU </PL> supports students communication using conferencing systems like the FirstClass Internet Client. She showed how this is in regular use by thousands of students, their tutors and counsellors. Gilly illustrated the work of <PL> the OU </PL> Business School including the recent developments in the Business Cafe. Peter demonstrated the new technologies which KMi is harnessing and shaping to support the students and staff of <PL> the OU </PL>, including the KMi Stadium project.

http://kmi.open.ac.uk/stadium/

Mr Brown and his party also had an opportunity to discuss other research projects, which exemplify KMi’s successful track record of collaboration with industry, with the institute’s research staff.

More information about the University for Industry may be found at:

http://www.transcend.co.uk/LIFELONG_LEARNING/ufi.htm

Figure 4: Annotated story

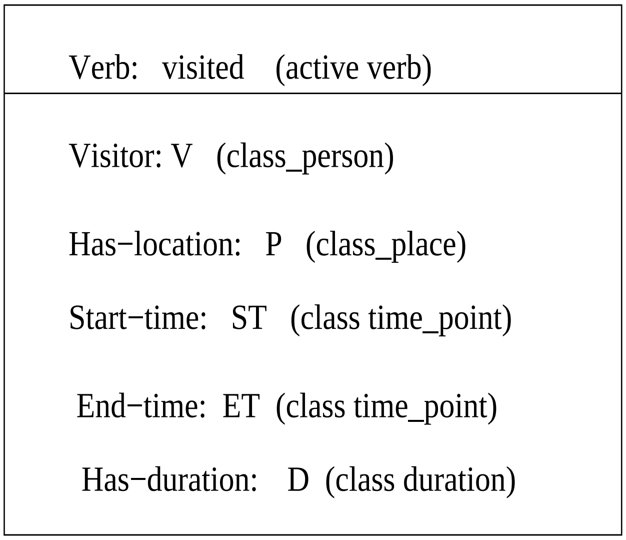
In the example **David Brown** was annotated as visitor and **The OU** is annotated as place.

Crystal initialises a CN dictionary for each positive instance of each type of event. The initial CN definitions are designed to extract the relevant phrases in the training instance that creates them but are too specific to apply to a unseen sentences. The main task of Crystal is to gradually relax the constraints on the initial definitions and also to merges similar definitions.

Crystal finds generalisations of its initial CN definitions by comparing definitions that are similar. This similarity is deduced by counting the number of relaxations required to unify two CN definitions. Then a new definition is created with constraints relaxed. Finally the new definition is tested against the training corpus to insure that it does not extract phrases that were not marked with the original two definitions. This means that Crystal takes similar instances and generalises into a more general rule by preserving the properties from each of the CN definitions which are generalised.

The inductive concept learning in Crystal is similar to the inductive learning algorithm described in [8] a specific-to-general data-driven search to find the most specific generalisation that covers all positive instances. Crystal finds the most specific generalisation that covers all positive instances but uses a greedy unification of similar instances rather than breadth-first search.

Coming back to our example given in Figure 3. We have that Crystal learns a conceptual node such as the one shown.



Verb: visited (active verb)

Visitor: V (class_person)

Has−location: P (class_place)

Start−time: ST (class time_point)

End−time: ET (class time_point)

Has−duration: D (class duration)

Figure 5: Concept node for the visiting event

These conceptual node states that “X visited”. So that in the future whenever the pattern “X visited ” appears in the text the case frame will extract “X” as the visitor.

For the pattern X visited Y, we basically we are extracting relations r(X,Y) from texts which could be interpreted as “X visited Y” and the Lexicon for relation r is the union of the lexicon(X) and lexicon(Y). If we find this relation in our texts then we find a instance for the event “visiting-a-place-or-people”.

In this example we do not have the case that two different templates might apply to the same sentence. But it is possible to encounter these cases. Let us consider the following example from the MUC domain:

“A visitor from England was hurt when two terrorists attempted to kill the major”.

if **visitor from England** is marked as victim **two terrorist** are marked as perpetrators and **major** as victim.

Crystal generates 3 frame cases that represents the following patterns:

If a text contains the expression “X was hurt” then the system extracts “X” as the victim.

If a text contains the expression “X attempted to kill” then the system extracts “X” as perpetrator.

If the text contains the expression “attempted to kill Y” then the system extracts “Y” as the victim.

In recent years had been great interest in annotated-based techniques for producing automatically dictionaries. The reason for this is that automatic creation of conceptual dictionaries is important factor for portability and scalability of an IE system.

Crystal has been tested on corpus of 300 stories. Crystal was able to induce a dictionary of CN definitions. In our domain we had achieved 100% precision [3] and 100% recall.

We remark, that in some cases the text does not provide enough

[3] recall is the percentage of possible phrases that the dictionary extracts and precision is the percentage correct of the extracted phrases.

context. Then Crystal would not learn any useful CN definitions. In this cases we would apply apply heuristics to identify proper nouns. NLP approaches typically use heuristics, for example, if a word is capitalised and not starting a sentence then it is a proper name. If a string contains “& Co” or “Ltd” then it can be tagged as a proper noun of type company.

4.3 Badger

A third component called Badger (from UMass) which was also integrated into our IE tool.

The main task of badger is to take each sentence in the text (in our case a story written in a e-mail message) and see if it matches any of our CN definitions. If no extraction CN definition applies to a sentence, then no information will be extracted; this means that irrelevant text can be processed very quickly.

It might occurs that Badger obtains more than one type of event for an story. Then our IE system decides to classify the story according with the following criteria: how many feature for each type were encountered in the story.

Badger obtained a case frame instantiations for Place and Visitor using conceptual nodes defined in the dictionary constructed by Crystal. In the following output from Badger the following conventions were used: the name of the slot appears in the left hand side of the arrow and the value for the slot on the right hand side of the arrow. In David Brown story, Badger instantiated Place to The OU and visitor to David Brown. The type of event is obtained from the value of Type and the document ID from docid.

```
<cn>ID: 169 164 158 145
          Type: visiting-a-place-or-people
docid =  david-brown-story
sentence_num =    3
segment_num =    1
Place ==> OBJ1: THE OU
</cn>
<cn>ID: 89 39
          Type: visiting-a-place-or-people
docid =  david-brown-story
sentence_num =    3
segment_num =    1
Visitor ==> SUBJ: DAVID BROWN
   %COMMA% THE CHAIRMAN OF THE UNIVERSITY
</cn>
```

The above output means that Badger had instantiated (using the CN definitions and domain lexicon) to a frame of the form:

```
Concept Node:
        CN-type: visiting-a-place-or-person
        Slots:
        Visitor   tag: VI
        Start-time  tag: ST
        End-Time    tag: ET
        Place     tag: PL
        Research-group tag: GR
```

Date is not stated in the story. So Start-time and End-time are instantiated to the date in which the story was written.

5. INFERENCE CAPABILITIES BY USING AN ONTOLOGY

An example of an story belonging to the type of event conferring-a-monetary-award is defined as follows. This example is described in this paper because shows the inference capabilities which could be obtained from using an IE tool plus an ontology.

> IBROW has been awarded 1 million Ecu from the European Commission to carry out research in the area of knowledge-based systems.

The output is shown as below.

```
<cn>ID: 80 Type: conferring-a-monetary-award
docid =  ibrow-story
sentence_num =    1
segment_num =    1
Funder ==> PP: FROM THE EUROPEAN COMMISSION
</cn>

<cn>ID: 106 Type:  conferring-a-monetary-award
docid =  ibrow-story
sentence_num =    1
segment_num =    1
Money ==> OBJ1: 1 MILLION ECU
</cn>

<cn>ID: 24 Type: conferring-a-monetary-award
docid =  ibrow-story
sentence_num =    1
segment_num =    1
Project-Institution ==> SUBJ: IBROW
</cn>
```

In this last example, we need to use the KMi planet ontology to find if Project-Institution is a **institution name** or **a project name**, and this is done by a simple traversal of the inheritance links in the ontology. Specifically, to remove ambiguity we sent a query to Web-onto asking for the set of all educational-organizations using the following query code.

```
 web-onto display akt-kmi-planet-kb
 ocml-eval(setofall ?x
         (educational-organization ?x))
```

This gives a list containing all educational-organizations:

```
 to give @(the-open-university
         ...
         org-knowledge-media-institute)
```

IBROW does not match any of these, however, we also send a query to Web-onto asking for the set of all kmi-projects:

```
web-onto display akt-kmi-planet-kb
  ocml-eval(setofall ?x
            (kmi-project ?x))
```

yielding

```
to give @(project-d3e
          ...
          project-kmi-planet
          ...
          project-ibrow
          ...
          project-heronsgate-mars-buggy)
```

and hence a match of “IBROW” to project-ibrow

In a similar fashion a query is sent to webonto in order to find if Funder is a valid funder body.

```
web-onto display akt-kmi-planet-kb
   ocml-eval(setofall ?x
             (awarding-body ?x))
```

```
    to give @( ...
               org-european-commission
               org-british-council)
```

At same time some **semantic relations** could be obtained by using the KMi planet ontology. For our example about IBROW (example 3) we can derive the following semantic relations:

“ibrow is KMi project” and “KMi is part-of the Open-University”

The OCML query to derive that KMi is part of the open university is as follows:

```
web-onto display akt-kmi-planet-kb

   ocml-eval(setofall ?x
             (organization-unit-part-of ?x
               the-open-university))


 to give @(knowledge-media-institute
   acad-unit-department-of-earth-science
   acad-unit-department-of-statistics-ou
   acad-unit-faculty-of-maths-and-computing-ou
   ...
   org-office-for-technology-development)
```

therefore we could conclude that:

“the Open-University has been awarded 1 million Ecu from the European Commission”

In a future implementation we will be interested in finding more complex relations by using our KMi Planet ontology.

Finally, we remark that OCML (the query language used by webonto) has adopted the closed world assumption (CWA), in the same fashion as Prolog, and so facts that are not provable are regarded as “false” as opposed to “unknown”.

6. OCML CODE GENERATED FROM OUR SYSTEM

Our goal is to use the information obtained by Badger and KMi ontology in order to be able to populate our KMi ontology with new instances of classes. In order to accomplish this task we had plugged another component which is a translator from Badger’s output to OCML code. The main function of this translator is to tokenise the Badger output and then find the CN definitions (cn markers) and extract all the objects encountered in the story. The name of each slot in the frame case corresponds to the name of the field in the class definition and the value for the field is the extracted information or it is a computed value using other extracted values.

For the example story shown in Figure 3 we end up with a visiting-a-place-or-people event and produce the intermediate output:

```
(def-instance  visit-of-david-brown-
     the-chairman-of-the-university
     visiting-a-place-people
  ((has-duration 1-day)
   (start-time  wed-15-oct-1997)
   (end-time  wed-15-oct-1997)
   (has-location  the-ou )
   (visitor  david-brown-the-
     chairman-of-the-university)
  )
)
```

where an instance of the type event visiting-a-place-or-people has been defined with the name “visit-of-david-brown-the-chairman-of-the-university”.

7. POPULATING THE ONTOLOGY

Building domain-specific ontologies often requires time-consuming expensive manual construction. Therefore we envisage IE as a technology that might help us during ontology maintenance process. During the population step our IE system has to fill predefined slots associated with each event, as already defined the ontology. Our goal is to automatically fill as many slots as possible. However, some of the slots will probably still require manual intervention. There are several reasons for this problem:

- there is information that is not stated in the story,
- none of our templates match with the sentence that might provide the information (incomplete set of templates)

We note that there are some cases when the instances are not defined in the ontology and then determining the type of an object is not straightforward. This has to be derived from a proof. Currently, we still looking to this aspect of our research.

Figure 6 shows the extracted information from David Brown story.

Once the system had extracted the information the user will presented with all extracted information even the one that cannot be categorized as belonging to a type of object defined in our domain. Therefore, before updating the ontology we will require that a person check/complete the extracted information. In other words the user have editing permissions before updating KMi ontology.

Currently, we had explored the creation of new instances of a class but we remark that we would like to create automatically new classes or subclasses in the future.

Badger and Crystal work using a predefined set of frames which are defined in the specification of the domain. Therefore, for each event’s class defined in KMi ontology we had defined a frame. Then, in order to create new classes we have to be able to create new temporal frames, i.e. if a new entity appears several times in the text. This is still under current investigation.

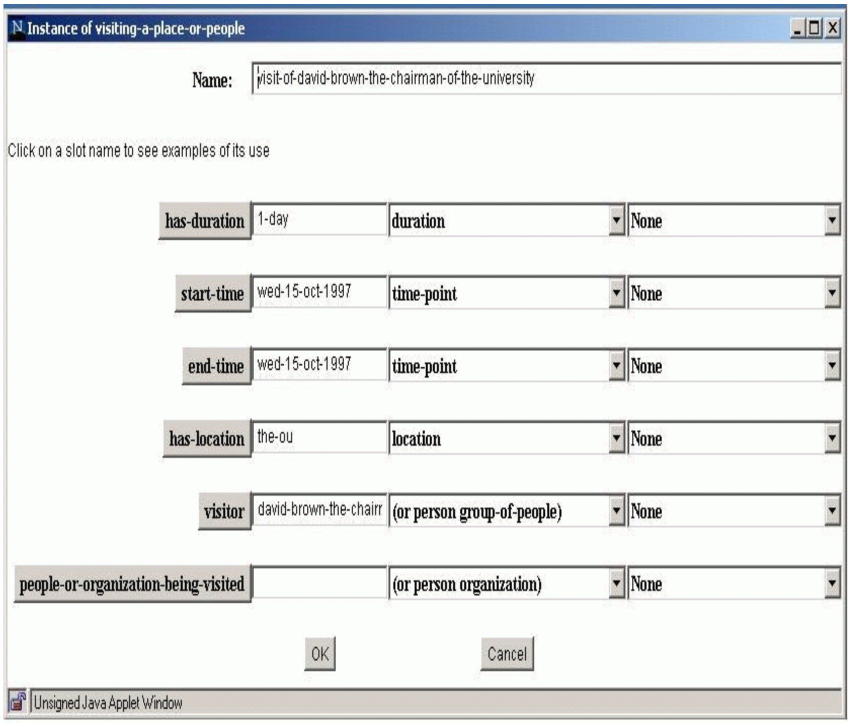
8. CONCLUSIONS AND FUTURE DIRECTIONS

We had built an IE system using Marmot, Crystal, Badger, a translator to OCML code and KMi ontology. We obtained good results using the IE tool in KMi stories, However we found that we have to integrate a regular expression recogniser in our system. The reason for this is that Crystal does not learn patterns in summaries of documented examples of lifelong learning (our second study case).

Currently, our system had been trained using the archive of 300 stories that we had collected in KMi [4] The training step was performed using typical examples of stories belonging to each of the different type events defined in the ontology. However, in the future we would like to use the IE tool in a different domain. We are interested in using the IE system in lifelong learning initiatives, companies project reports, Curriculum Vitae (CV’s), or application of jobs.

Another possible direction that we would like to explore is to incorporate in Crystal a different Machine Learning algorithm in order to compare performance between them.

[4] URL:http://kmi.open.ac.uk/planet/



Instance of visiting-a-place-or-people

Name: visit-of-david-brown-the-chairman-of-the-university

Click on a slot name to see examples of its use

has-duration	1-day	duration	None
start-time	wed-15-oct-1997	time-point	None
end-time	wed-15-oct-1997	time-point	None
has-location	the-ou	location	None
visitor	david-brown-the-chairr	(or person group-of-people)	None
people-or-organization-being-visited		(or person organization)	None

Figure 6: Extracted information

Currently, in our IE tool works in plain email text. But, in future we would consider the possibility of using our IE tool in hypertext.

Besides the above issues, Badger could be extended in order to save its output in XML (Extensible Markup Language). This will increase the portability of our IE system as XML is the universal format for structured documents and data on the Web.

Finally, we would like to provide our IE system with visualisation capabilities such as visualisation of entities, etc.

9. REFERENCES

[1] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, K. Nigam T. Mitchell, and S. Slattery. Learning to Construct Knowledge Bases from the World Wide Web. *Artificial Intelligence*, 1999.

[2] M. Craven and J. Kumlien. Constructing Biological Knowledge Bases by Extracting Information from Text Sources. In *Proceedings of The 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*, 1999.

[3] J. Domingue and P. Scott. Kmi planet: putting the knowledge back into media. *The Knowledge Web*, pages 173–184, 1999.

[4] J.B. Domingue. Tadzebao and WebOnto:Discussing, Browsing and Editing Ontologies on the Web. In *Proceedings of the Knowledge Acquisition Workshop*, 1998.

[5] J.B. Domingue and E. Motta. PlanetOnto: From News Publishing to Integrated Knowledge Management Support. *IEEE Intelligent Systems*, 15(3):26–32, 2000.

[6] J-U. Kietz, A. Maedche, and R. Volz. A method for semi-automatic ontology acquisition from a corporate intranet. In *Proceedings of the EKAW'00 Workshop on Ontologies and Text, Juan-Les-Pins, France*, oct 2000.

[7] A. Maedche and S. Staab. Semi-automatic engineering of ontologies from texts. In *Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, SEKE2000, Chicago, IL, USA*, pages 231–239, jul 2000.

[8] T. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[9] E. Motta. *Reusable Components for Knowledge Modelling*. IOS Press, Netherlands, 1999.

[10] D. Proux and Y. Chenevoy. Natural Language Processing for Book Storage: Automatic Extraction of Information from Bibliographic Notices. In *Proceedings of The Natural Language Processing Pacific Rim Symposium (NLPRS'97)*, pages 229–234, 1997.

[11] E. Riloff. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of The 11th National Conference on Artificial Intelligence (AAAI-93)*, pages 811–816. AAAI press, 1993.

[12] E. Riloff. An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *AI Journal*, 85:101–134, 1996.

[13] C. Roux, D. Proux, F. Rechenmann, and L. Julliard. An Ontology Enrichment Method for a Pragmatic Information Extraction System gathering Data on Genetic Interactions. In *Proceedings of The 14th European Conference on Artificial Intelligence (Workshop on Ontology Learning ECAI-2000)*, 2000.

[14] S. Soderland, D. Aronow, D. Fisher, J. Aseltine, and W. Lehnert. Machine Learning of Text Analysis Rules for Clinical Records. Tr 39, Center for Intelligent Information Retrieval, 1995.