Available online at www.sciencedirect.com
ScienceDirect
Int. J. Human-Computer Studies 65 (2007) 537–562
International Journal of Human-Computer Studies
www.elsevier.com/locate/ijhcs

Ontologies as facilitators for repurposing web documents

Mark J. Weal*, Harith Alani, Sanghee Kim, Paul H. Lewis, David E. Millard, Patrick A.S. Sinclair, David C. De Roure, Nigel R. Shadbolt

Intelligence, Agents, Multimedia Group, School of Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK

Received 30 August 2005; received in revised form 26 January 2007; accepted 1 February 2007
Communicated by F. Ciravegna
Available online 28 February 2007

Abstract

This paper investigates the role of ontologies as a central part of an architecture to repurpose existing material from the web. A prototype system called ArtEquAKT is presented, which combines information extraction, knowledge management and consolidation techniques and adaptive document generation.

All of these components are co-ordinated using one central ontology, providing a common vocabulary for describing the information fragments as they are processed. Each of the components of the architecture is described in detail and an evaluation of the system discussed. Conclusions are drawn as to the effectiveness of such an approach and further challenges are outlined.

Keywords: Knowledge extraction; Ontology population; Narrative generation

1. Introduction

There is a vast amount of information present on the web for a wide range of domains. Despite this challenging information space, web users are able to search for material of interest using commercial search engines such as Google and Yahoo. Such search methods present information to the user *in its original form, structured for its original purpose*. Often this is more than adequate for the reader, but sometimes they might want something more focused on a particular task, or collated from a broader set of resources.

Let us take a concrete example, one we will use as the basis for much of what follows in this paper. A person wishes to find out information about an artist. Perhaps they have a specific piece of information they wish to find; perhaps they want a summary of his/her work; maybe they just want a biography of his/her life. Typing the name of the artist into a search engine like Google they are returned a set of search results. If they are lucky and the artist is reasonably well known, the first search result may be a useful biography such as those produced at the WebMuseum[1] and the biography may answer all of their questions. However this might not occur for a variety of reasons:

- A useful biography of the artist exists, but it is buried amongst a mass of returned results for hotels, art shops, dentist practices that all use the artists name by coincidence or design.
- A good biography exists for the artist but it does not contain specific facts that the reader was interested in finding. These may well exist on less comprehensive biography pages.
- The artist is relatively unknown, and although a number of web pages contain fragments of information about them, no single page is satisfactory.
- The reader is unable to ask specific questions about an artist such as 'What was Holbein's date of birth?' or 'Who were Holbein's influences?'

To solve this problem, what the user needs is a system with a behaviour depicted in Fig. 1. A system that scours the web looking for any pages or fragments of pages that

*Corresponding author. Tel.: +44 23 80594059; fax: +44 23 80592865.
E-mail address: mjw@ecs.soton.ac.uk (M.J. Weal).

[1] http://www.ibiblio.org/wm/

1071-5819/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijhcs.2007.02.001

contain information about the artist and combines them into one final document that satisfies the reader.

To produce such a system requires a solution to a number of key problems:

- We have to find documents on the web that might contain useful content.
- We have to identify and extract the relevant bits of information from the documents.
- We need to be able to understand and structure that which we have extracted if we are to be able to reconfigure it for specific purposes and avoid duplication, or inaccurate information.
- We have to establish what inaccurate means in this context.
- We need to design suitable document structures which we might wish to produce.
- We have to generate the documents from the information we have extracted.

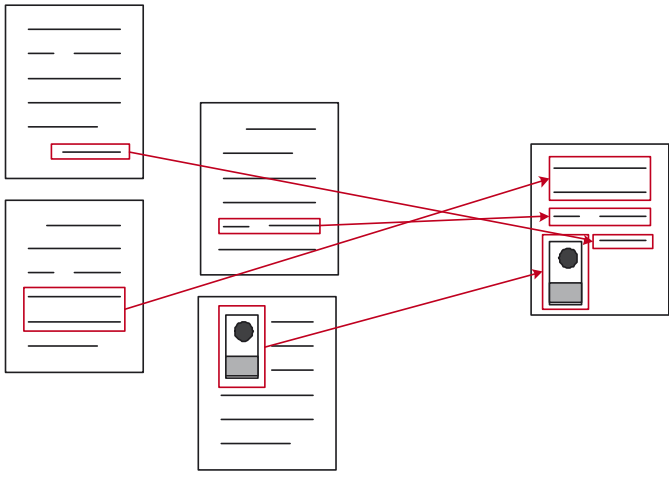

Fig. 1. The compositing of documents.

These are not new problems. Some of them are at the core of whole research fields. What we are aiming for in our work is an understanding of how these problems fit together; how they can be assembled into a chain of processes that achieves the goal, and how the success/failure of any process affects the others. Our approach is to use an ontological model of the domain as a facilitator throughout all the processes. This provides a common vocabulary and specifies the semantics of key relationships within the domain. It can be used as a structure for a knowledge base (KB) of information accumulated from the web, which can then be used as the basis for reasoning and document generation.

Fig. 2 illustrates how such a chain of processes might operate. Each process takes its cues from the ontology, which provides a common reference model for all parts of the chain. At the front are the information extraction (IE) technologies. These include the search technologies as well as those technologies carrying out natural language understanding and extraction. The search technologies might use existing information from the KBs and structural information from the ontology to help build their queries. In the case of the current prototype the creation of queries from the ontology is not automatic. The vocabulary and relationships held within the ontology provide an underpining for the IE tools.

Once the raw source information has been harvested, processed and structured, it is passed to the KB technologies for structuring and consolidation. As well as using the ontology to organise the storage of the gathered information, the relationships can be used to help the heuristic based consolidation processes and to help verify the information.

Once a structured KB of information has been collated, stories (constructed sequences of information fragments) can be generated from this using narrative generation tools and techniques. Again, the ontology feeds into the process,

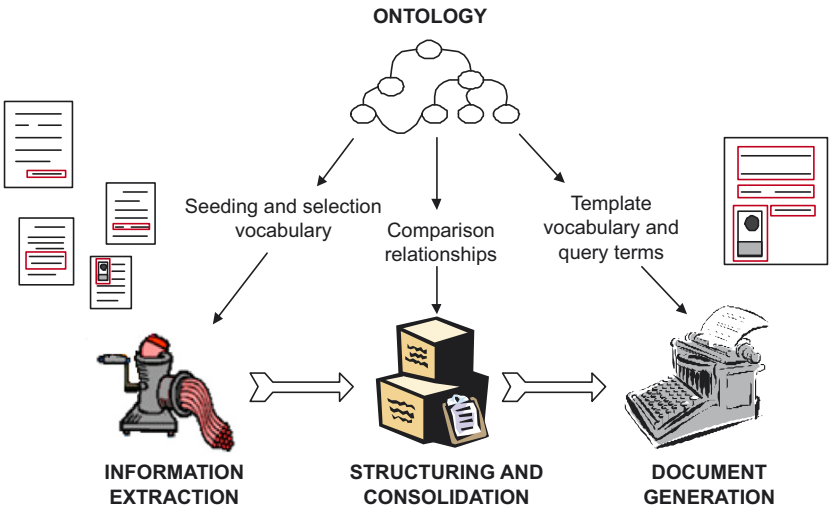

Fig. 2. The chain of processes organised by an ontology.

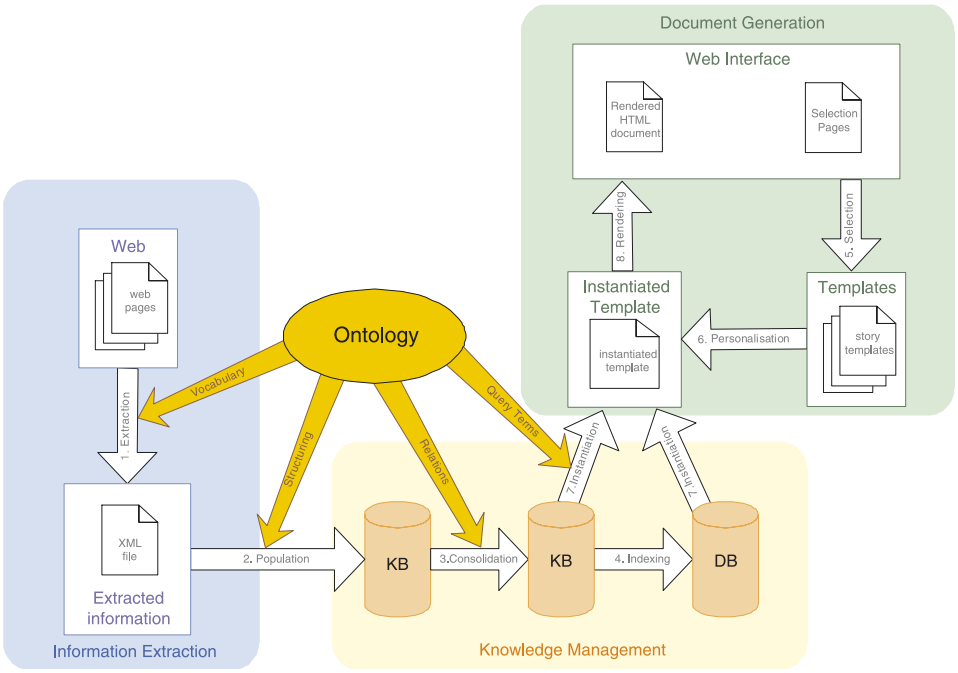

Fig. 3. The ArtEquAKT conceptual architecture.

with the story template queries being structured around the vocabulary and relationships of the ontology and the subsequent stories adapted so as not to duplicate information.

The prototype system which we have constructed to support such a chain we have called ArtEquAKT.

1.1. The ArtEquAKT project

The ArtEquAKT project seeks to create dynamic biographies of artists by harvesting biographical information from the web (Alani et al., 2003). The system uses the information harvested from the web to automatically instantiate ontologies. It then reconstructs the required biographies using the populated ontology and annotated fragments based on user preferences and story schema.

ArtEquAKT draws on the expertise of two EPSRC Interdisciplinary Research Collaborations (IRCs), AKT and Equator, in addition to the domain expertise of the European project Artiste which, amongst other things, provided input into aspects of the IE systems.

1.2. Overview

In Section 2 we will describe the ontology used in more detail, including discussion on its design and the appropriation of existing ontologies. More detailed information is included on how the IE techniques are applied using the ontology as a grounding vocabulary (Section 2.3) and the consolidation and verification processes carried out during storage of the extracted information (Sections 2.4 and 2.5). Section 2.6 focuses on the narrative generation aspects of ArtEquAKT. In Section 3, a discussion and an evaluation are provided. Section 4 gives exemplars of the related work in the areas drawn together in the project and finally conclusions are drawn along with suggestions for where this work might develop in the future in Section 5.

2. The ArtEquAKT system

In this part we examine the ArtEquAKT architecture and its component parts. First we will discuss the conceptual representation of the architecture shown in Fig. 3, illustrating processes and information flows. These have been broken down into the three broad areas: IE, knowledge management and document generation. The following sections provide a more detailed system overview of the constructed prototype (see Fig. 4) showing the interaction of the various system components used in the document construction process. An additional section will then briefly cover the user interface to the system.

2.1. Conceptual architecture

Fig. 3 shows a conceptual architecture for the system depicting the different processes that take place during the construction of a document (labelled 1–8.) The information is also shown, illustrating how it is modified and manipulated as it passes through the system. The diagram illustrates the role of the central ontology in these processes, each of which is described in more detail below.

(1) *Extraction*: The extraction process involves the use of search engines to identify documents containing

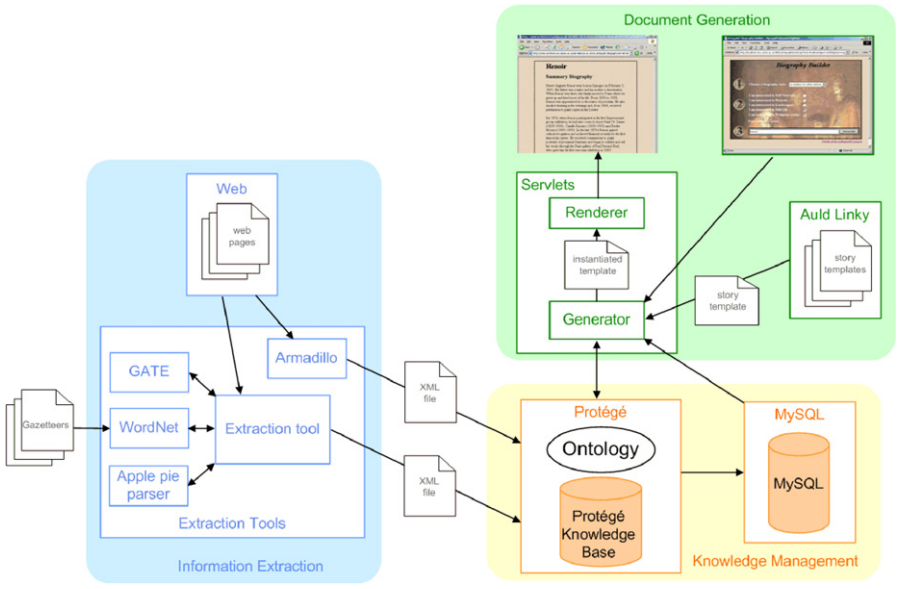

Fig. 4. The ArtEquAKT prototype architecture.

information about the artist requested and the subsequent extraction of facts and relations and images from within the documents.

(2) *Population*: The extracted information is fed into the KB creating new entities and relationships.
(3) *Consolidation*: The KB is consolidated to combine entities that are concluded to be identical based on similar information and relationships.
(4) *Indexing*: The database (DB) index is created from the KB allowing fast access for simple paragraph and sentence queries.
(5) *Selection*: The reader selects a biography from a list of possible types, enters a list of preferences and the name of an artist.
(6) *Personalisation*: The selected template is personalised using the readers preferences to selectively remove parts of the template that are not of interest to the reader.
(7) *Instantiation*: Using queries to the KB and DB, the template is instantiated with information. The instantiated template contains hypermedia structures.
(8) *Rendering*: The hypermedia structures are rendered as a HTML page with adaptive content. Some information considered supplementary is initially hidden from the user but is available on request.

The specific implementation details of the ArtEquAKT prototype developed to illustrate the conceptual architecture described above can be seen in Fig. 4. The three areas of IE, knowledge management and document generation have again been highlighted to help indicate the mapping between the concepts and the final implementation.

2.2. The ArtEquAKT ontology

For ArtEquAKT the requirement was to build an ontology to represent the domain of artists and artefacts. The main part of this ontology was constructed from selected sections in the CIDOC Conceptual Reference Model (CRM) ontology[2]. The CRM ontology is written in RDF and is designed to represent museum artefacts, their production, ownership, location, etc. This ontology was modified for ArtEquAKT and enriched with additional classes and relationships to represent a variety of information related to artists, their personal information, family relations, relations with other artists, details of their work, etc. The ArtEquAKT ontology and KB are accessible via an ontology server.

The ArtEquAKT ontology was implemented in Protégé (Musen et al., 2000), an ontology–engineering tool developed by Stanford Medical Informatics. The ArtEquAKT ontology contains 43 classes and over 230 relations, and is formalised in RDF. Fig. 5 shows a subset of the ontology held in Protégé.

2.3. IE in ArtEquAKT

ArtEquAKT's knowledge extraction tool aims to identify and extract knowledge triples from text documents and to provide them as RDF triples for entry into the KB (Kim et al., 2002). ArtEquAKT uses its ontology coupled with a general-purpose lexicon (WordNet, Miller et al., 1993), a syntactic parser (Apple Pie Parser, Sekine and Grishman, 1995), an entity-recogniser (GATE, Cunningham et al., 2002b), and a wrapper (Armadillo, Ciravegna et al., 2004) as supporting tools for identifying knowledge fragments.

Documents relevant to a given artist are identified using online search engines and content similarity analysis. The similarity of a candidate document is measured against an example biography using a term vector similarity measure. HTML tags are removed to extract only texts from the

[2]http://cidoc.ics.forth.gr/

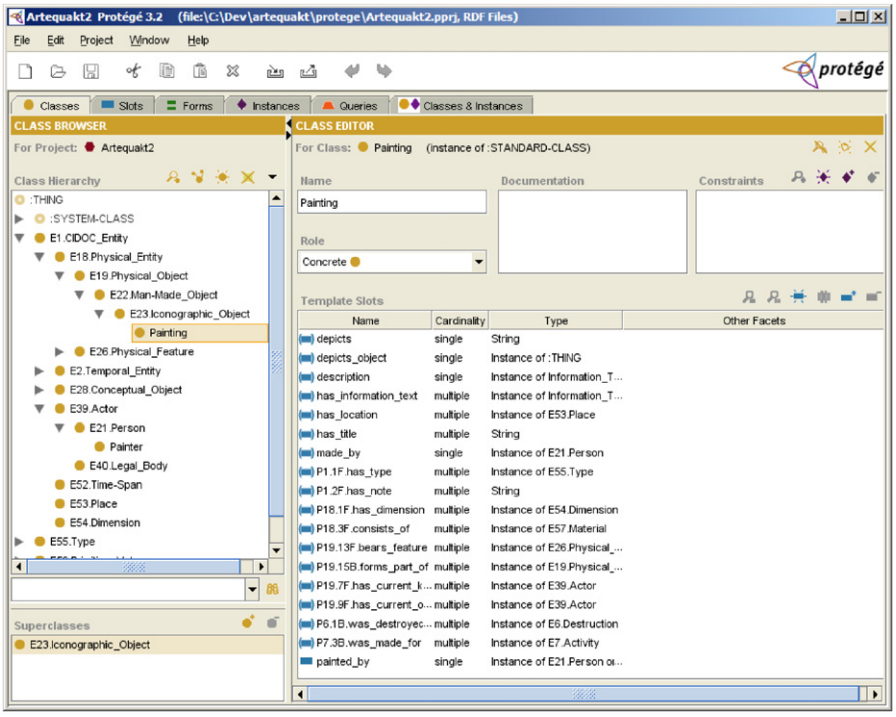

Fig. 5. A portion of the ArtEquAKT ontology.

biography. A deletion of very common words based on a stoplist file is performed in advance. This is followed by a stemming algorithm which removes the common morphological and inflexional endings from words and converts them into a normalised form. A simple term-frequency processing is then applied to the biography in order to convert the biography into a term-vector model. We use the cosine function in order to measure a content similarity between the biography and a candidate document retrieved by using the search engine. Documents with similarity above a given threshold are selected for analysis. These documents are then analysed syntactically and semantically to identify any relevant knowledge to extract. ArtEquAKT attempts to identify not just entities, but also their relationships.

There are some trusted and rich online sources and DBs for information about artists (e.g. ULAN[3]), which could be used to bootstrap systems like ArtEquAKT. However, the greater challenge is to be able to automatically locate and extract information from the web, irrespective of source or structure. Therefore, ArtEquAKT was not bootstrapped with any data from existing DBs or other structured sources to experiment with relying entirely on automatic IE from arbitrary sources. Bootstrapping, however, is a good strategy for speeding up data gathering with reliable data, which can be used to verify any automatically collected information (e.g. Grishman and Sundheim, 1996).

The IE component in ArtEquAKT uses the ontology coupled with the general-purpose lexical DB WordNet (Miller et al., 1993) and a software architecture for language engineering used for entity recognition, GATE (General Architecture for Text Engineering, Cunningham et al., 2002a) as IE tools for identifying knowledge fragments consisting not just of entities, but also the relationships between them. Automatic term expansion based on WordNet is used to increase the scope of text analysis to cover syntactic patterns that imprecisely match our definitions.

When a user searches for an artist, if the given artist is new to the KB, the IE process is initiated. Firstly, a script submits the artist's name as a query to search engines (currently we use 'Google').

In order to select only art-related web pages (as opposed to pages which may match the search criteria but are concerned with other topics) we use a keyword template. The template keywords are extracted from trusted biography sites and used to identify the likelihood of a search result being an artist biography. The search engine results are compared to the template and those that surpass a certain threshold are taken to be biographies with the remainder discarded. The filtered URLs are perhaps those sites such as restaurants or hotel web pages, which may contain the artists name but do not meet the necessary similarity measure.

Table 1 shows the list of web pages that the system selected to extract information about Renoir.

Each selected document is then divided into paragraphs and sentences. Each sentence is analysed syntactically and semantically to identify any relevant knowledge to extract. The Apple Pie Parser (Sekine and Grishman, 1995) is used for grouping grammatically related phrases as the result of

[3]http://www.getty.edu/research/conducting_research/vocabularies/ulan/

syntactical analysis. Semantic examination then locates the main components of a given sentence (i.e. 'subject', 'verb', 'object'), and identifies named entities (e.g. 'Renoir' is a *Person*, 'Paris' is a *Place*) using GATE and WordNet. GATE is also used to resolve anaphoric references (personal pronouns). Fig. 6 illustrates this process. Below is an example of an extracted paragraph:

> Pierre-Auguste Renoir was born in Limoges on February 25, 1841. His father was a tailor and his mother a dressmaker.

Table 1
Web pages from which information about Renoir was extracted from

1	www.respree.com/cgi-bin/SoftCart.exe/biography/pierre-auguste-renoir.html?E+scstore
2	www.phillipscollection.org/html/lbp.html
3	www.abcgallery.com/R/renoir/renoirbio.html
4	www.theartgallery.com.au/ArtEducation/greatartists/Renoir/about/
5	www.art-and-artist.co.uk/impressionist/
6	www.biography.com/impressionists/artists_renoir.html
7	csmweb2.emcweb.com/durable/1997/09/04/feat/arts.1.html
8	www.guardian.co.uk/arts/portrait/story/0,11109,740299,00.html
9	www.island-of-freedom.com/renoir.htm
10	www.expo-renoir.com/2.cfm

The challenge is to extract relationships between any identified pair of entities. Knowledge about the domain specific semantic relations is required, which can be inferred from the ArtEquAKT ontology and used to decide which relations are expected between the entities in hand. In addition, three lexical relations (synonyms, hypernyms, and hyponyms) from WordNet are used to reduce the problem of linguistic variations given identified entities. Since the relation may have multiple entries in WordNet (polysemous words), the mapping between an ontology relation and an entry in WordNet takes into account syntactic and semantic clues present in a sentence. For example, the relation of 'Person *date_of_birth* Date' maps into the concept of 'birth' which, in WordNet, has four noun senses and one verb sense. The first noun sense is selected since one of its hypernyms is 'time period' which has 'Date' as a hyponym.

Annotations provided by GATE and WordNet highlight that 'Pierre-Auguste Renoir' is a *Person's name*, 'February 25, 1841' is a *Date*, and 'Limoges' is a *Place*. Relation extraction is determined by the categorisation result of the verb 'bear' which matches with two potential relations; *date_of_birth* and *place_of_birth*. Since both relations are associated with 'February 25, 1841' and 'Limoges'

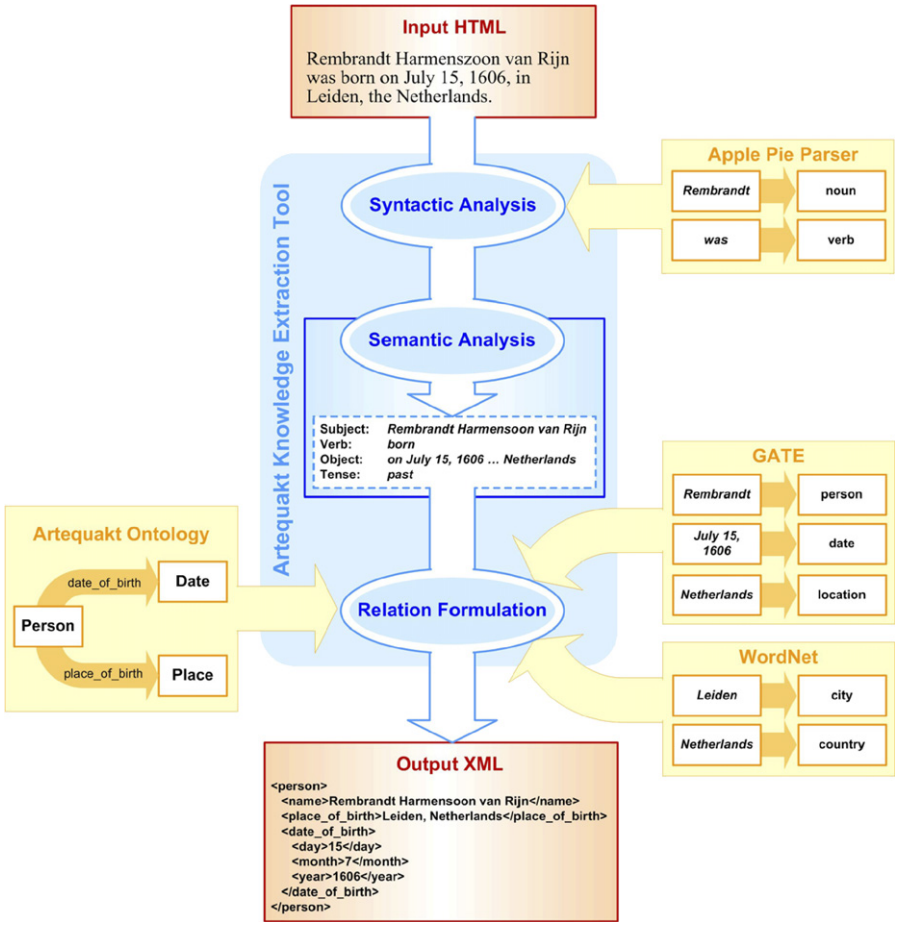

Fig. 6. The information extraction process.

respectively, this sentence generates the following knowledge triples about Renoir:

- Pierre-Auguste Renoir *date_of_birth* 25/2/1841
- Pierre-Auguste Renoir *place_of_birth* Limoges

The second sentence generates knowledge triples related to Renoir's family:

- Pierre-Auguste Renoir *has_father* Person_1
- Person_1 *job_title* Tailor
- Pierre-Auguste Renoir *has_mother* Person_2
- Person_2 *job_title* Dressmaker

The system does not do any 'sense' disambiguation here when extracting relations in texts. At this point it is focussed on extracting relations and since the relations are defined as triples, the correct extraction of the relations depends on the named-entity identification provided by GATE. A pseudo-code description of the extraction process is provided in Fig. 7.

Sense disambiguation is complex and error-prone and the set of relations defined in the prototype ontology is reasonably mutually independent from linguistic observation. This means that although *has* is ambiguous, the *has_father* relation can be correctly extracted if it links with two named-entities, i.e. Person and if somehow the sentence expresses the concept of *father*. Unique named identifies are created here *Person_1* and *Person_2* to represent the unnamed father and mother in the text. Furthermore, due to the web redundancy, although some errors are occurred in IE process, even a simple frequency-based filtering can remove the errors.

The extraction of relations is based on a sentence which consists of clauses. The correct extraction depends on the results of the named-entity identifications provided by

```
Variables:
    sentenceList: a list of identified sentences
    s1: one sentence in sentenceList
    textList: text fragments which are not annotated
    t1: one text fragment in textList
    wordnetList: entries of the WordNet for t1
    w1: one entry in wordnetList
    m1: a main concept(s)
    expandList: hypernyms and hyponyms of WordNet for t1
    e1: one entry in expandList
    relationList: relationships defined in the ontology
    r1: one relation in relationList

Repeat:
    foreach s1 <- sentenceList
    {
        get NE annotations using GATE
        highlight text fragments which are not annotated with any of pre-defined NEs
        SET textList
        foreach t1 <- textList
        {
            identify a main concept(s), i.e. a main verb or a head noun in t1
            SET m1
            refer to the WordNet definitions
            SET wordnetList
            IF (wordnetList is EMPTY) THEN { return NOTHING}
            ELSE
            {
                foreach w1 <- wordnetList
                {
                    IF (m1 or synonyms matches w1) THEN
                        retrieve all hypernyms and hyponyms
                        SET expandList
                        foreach e1 <- expandList
                        {
                            foreach r1 <- relationList
                            {
                                IF e1 matches r1 THEN {return r1}
                            }
                        }
                    }
                }
            }
        }
    }
```

Fig. 7. Pseudo-code for the relation extraction process.

GATE and a list of pre-compiled synonyms of the relations. Any ambiguities in relation extraction tend to result from the ambiguities of the sentence structure rather than due to the polysemy of words. For example, when two Person entities are identified, it is difficult to correctly recognise a main actor participating in the relation extracted. In our approach, we model each sentence based on clauses, which in fact are represented as *subject–verb–object* formats. If a given sentence could be easily converted into a *subject–verb–object* form, then the chances of extracting correct relations is increased. But for the sentences which have no explicit verbs or ill-formed sentences, the parsing errors can be high. Some relations are incorrectly extracted and the experimental results are shown in Table 2.

The output RDF representation is submitted to the ontology server to be inserted into the KB. It would be possible to use this RDF to annotate the existing pages for integration with a Semantic Web (Millard et al., 2003).

In addition to textual information, Armadillo is used to extract references to images of paintings on the web. These are associated with the artists in the KB and can be used in the rendered biographies. The RDF produced by Armadillo is fed into the ontology server in much the same way as the RDF concerning textual fragments.

Some of the extracted information needs to be represented in *n*-ary relations. For example 'Renoir began studying in the Ecole des Beaux-Arts in 1862'. The relation 'studying' here is associated with the year 1862, and thus cannot be treated as a simple binary relation. Some of the relations, such as "studying", are represented as reified relations in Protégé. More complex relations (e.g. 4-ary and higher) are not yet extracted properly by ArtEquAKT, but rather treated as binary ones, or cut to 3-ary relations. More work is required for a better extraction of such relations.

2.4. KB management in ArtEquAKT

In ArtEquAKT we investigate the possibility of moving towards a fully automatic approach of feeding the ontology with knowledge extracted from unstructured text. Information is extracted in ArtEquAKT with respect to a given ontology and provided as RDF files using tags mapped directly from names of classes and relationships in that ontology.

When ArtEquAKT's ontology server receives a new RDF file from the ArtEquAKT knowledge extractor, a feeder tool is activated to parse the file and add its knowledge triples to the KB automatically, thus populating the ontology with additional knowledge. Once the feeding process terminates, the consolidation tool searches for and merges any duplication in the KB.

The ArtEquAKT ontology is held within the Protégé ontology engine. This serves as the KB for holding the RDF extracted during the IE process. Paragraphs of text extracted from the documents are associated with the relevant instances in the KB. In addition, a MYSQL DB is used to index the paragraphs in order to speed up access to the original textual content. As new RDF is added to the KB it is consolidated with the existing information by a number of scripts running as part of the Protégé engine.

2.5. KB consolidation in ArtEquAKT

ArtEquAKT applies a set of heuristics and reasoning methods in an attempt to distinguish conflicting information and to identify and merge duplicate assertions in the KB automatically.

Consolidation of duplications in ArtEquAKT is carried out in three categories: thematic, geographic, and temporal. Thematic consolidation is mainly concerned with merging the entities that represent artists. This type of merging investigates the similarity of artists' names and the similarity and overlap of their information.

ArtEquAKT's geographical consolidation deals with any instantiations of place names using WordNet as a limited source of geographical information, such as place name synonyms and part-of relationships (e.g. London is part of England).

The temporal consolidation deals with identifying and merging duplicated dates, such as dates of birth, death,

Table 2
Precision/recall of extracted relations for 5 artists.

Artist (P/R) relation	Rembrandt (P/R)	Renoir (P/R)	Cassatt (P/R)	Goya (P/R)	Courbet (P/R)	Average per relation
Date of birth	75/43	100/50	100/67	80/40	100/100	91/60
Place of birth	100/63	100/14	100/50	100/40	100/63	100/46
Date of death	100/63	100/67	100/50	N/A/0	100/50	100/46
Place of death	100/100	100/43	N/A/0	100/20	100/33	100/39
Place of work	100/50	67/33	33/100	N/A/0	0/0	40/37
Place of study	100/20	100/14	100/75	100/20	100/29	100/32
Date of marriage	100/50	100/33	N/A	100/100	N/A/0	100/37
Name of spouse	100/38	N/A/0	N/A	N/A/0	N/A/0	100/10
Parent profession	100/57	50/67	0/0	67/100	100/100	63/65
Inspired by	100/43	50/60	0/0	100/17	100/33	70/31
Averages	98/53	85/38	61/43	92/34	88/41	85/42

marriage, etc. which could have been extracted in various formats and specificity levels.

The order in which these consolidation steps are applied may have an affect on the results. For example if we first consolidate based on dates and places of birth, then artists with the same values for these relations will be merged into one artist instance. Whether this is wrong or right is dependent on the types of duplication that exist in the KB.

ArtEquAKT first consolidates based on artists names, then consolidates geographical and temporal information, which then helps to consolidate the artists instances further. The order was chosen based on our understanding of the type of data and their duplications in our KB. The consolidation steps are described in the following sections. Note that ArtEquAKT was not bootstrapped with any information. Hence the consolidation is not forced by or based on any pre-existing information in the KB, but rather entirely dependent on the information that the system manages to extract from the web.

2.5.1. Duplicate information

There exist two main types of duplication in our KB; duplicate instances (e.g. multiple instances representing the same artist), and duplicate attribute values (e.g. multiple dates of birth extracted for the same artists).

ArtEquAKT's IE tool treats each recognised entity (e.g. Rembrandt, Paris) as a new instance prior to consolidation. This may result in creating instances with overlapping information (e.g. two Person instances with the same name and date of birth). The role of consolidation in ArtEquAKT includes analysing and comparing instances and attribute values of the instances of each type of concept in the KB (e.g. Person, Date) to identify inconsistencies and duplications.

The amount of overlap between the attribute values of any pair of instances could indicate their duplication potential. However, there are not always overlaps of information between instances. IE tools are sometimes only able to extract fragments of information about a given entity (e.g. an artist), especially if the source document or paragraph is small or difficult to analyse. This leads to the creation of new instances with only one or two facts associated with each. For example two artist instances with the name Rembrandt, where one instance has a location relationship to Holland, while the other has a date of birth of 1606. Comparing such shallow instances will not reveal their duplication potential other than that they share an artist's name. Furthermore, neither the source information nor the IE is always accurate. For example a Rembrandt instance can be extracted with the correct family attribute values, but with the wrong date of birth, in which case this instance will be mismatched with other Rembrandt instances in spite of referring to the same artist. The pseudo-code representing the algorithm for consolidation of duplicates can be seen in Fig. 8.

The following are the steps taken to consolidate thematic information, listed in the order in which they are applied.

Unique name assumption: One basic heuristic applied in ArtEquAKT is that artist names are unique, such that artist instances with identical names are merged. According to this heuristic, all instances with the name Rembrandt Harmenszoon van Rijn are combined into one instance. This heuristic is obviously not fool proof, but it works reasonably well in the limited domain of artists. An artist can have multiple names in the ontology. So if more than one name is shown to refer to the same artist, then all can be stored in the KB using the CRM's synonym relation for convenience.

Information overlap: There are cases where the full name of an artist is not given in the source document or its extraction fails, in which case they will not be captured by the unique-name heuristic. For example, when we extracted information about Rembrandt and merged same-name artists, two instances amongst those that remained for this artist are: Rembrandt and Rembrandt Harmenszoon van Rijn. In such a case we compare other attribute values, and merge the two instances if there is sufficient overlap. For the two Rembrandt instances, both had the same date and place of birth, and therefore were combined into one instance. The duplication would not have been caught if these attributes had different values.

```
FOR every two Atrist instances in KB
{
    IF Artist_1(name) = Artist_2(name) THEN merge
    ELSE IF Artist_1(name) overlap with Artst_2(name)
    {
       IF { Artist_1(date of birth) = Artist_2(date of birth) AND
            Artist_1(place of birth) = Artist_2(place of birth) } THEN merge
       ELSE IF { Artist_1(date of birth) = Artist_2(date of birth) AND
            Artist_1(date of death) = Artist_2(date of death) } THEN merge
       ELSE IF { Artist_1(place of birth) = Artist_2(place of birth) AND
            Artist_1(date of death) = Artist_2(date of death) } THEN merge
       ELSE IF { Artist_1(place of birth) = Artist_2(place of birth) AND
            Artist_1(place of death) = Artist_2(place of death) } THEN merge
       ELSE no merge
    }
    ELSE no merge
}
```

Fig. 8. Pseudo-code for consolidating duplicate information.

Attribute comparison: When the above heuristics are applied, merged instances might end up having multiple attribute values (e.g. multiple dates and places of birth), which in turn need to be analysed and consolidated. Note that some of these attributes might hold conflicting information that should be verified and held for future comparison and use.

Comparing the values of instance attributes is not always straightforward as these values are often extracted in different formats and specificity levels (e.g. synonymous place names, different date styles) making them harder to match. ArtEquAKT applies a set of heuristics and expansion methods in an attempt to match these values. Consider the following sentences:

(1) *Rembrandt was born in the 17th century in Leyden.*
(2) *Rembrandt was born in 1606 in Leiden, the Netherlands.*
(3) *Rembrandt was born on July 15 1606 in Holland.*

These sentences provide the same category of information about an artist, written in different formats and specificity levels and all three sentences are consistent. Storing this information in the KB in such different formats is confusing for the biography generator which can benefit from knowing which information is consistent and which is contradictory and also the level of specificity. Matching the above sentences required enriching the original ontology with some temporal and geographical reasoning.

2.5.2. Geographical consolidation

There has been much work on developing gazetteers of place names, such as the Thesaurus of Geographic Names (TGN) (Harpring, 1997) and Alexandria Digital Library (Hill et al., 1999). Ontologies can be integrated with such sources to provide the necessary knowledge about geographical hierarchies, place name variations, and other spatial information (Alani et al., 2000). ArtEquAKT derives its geographical knowledge from WordNet. WordNet (Miller et al., 1993) contains information about geopolitical place names and their hierarchies, providing three useful relations for the context of ArtEquAKT; synonym, holonym (part of), and meronym (sub part). The ArtEquAKT ontology is extended to add this information for each new instance of place added to the KB. Note that WordNet's geographical coverage is very limited in comparison to TGN and other similar geographic thesauri, but was sufficient for our immediate requirements and to demonstrate the principles of incorporation of a geographical thesaurus. The pseudo-code for consolidation can be seen in Fig. 9.

Place name synonyms: The synonym relationship is used to identify equivalent place names. For example the three sentences above mention several place names where Rembrandt was born. Using the synonym relationship in WordNet, Leyden can be identified as a variant spelling for Leiden, and that Holland and The Netherlands are often referred to synonymously (albeit incorrectly).

```
FOR ALL Place instances in KB
{
    IF Place_1 PART_OF Place_2 OR Place_1 SYNONYM_TO Place_2
    THEN
    {
        consider Place_1 = Place_2 when merging
        use Place_1
    }
}
```

Fig. 9. Pseudo-code for consolidating geographical information.

Place specificity: The part-of and sub-part relationships in WordNet are used to find any hierarchical links between the given places. WordNet shows that Leiden is part of the Netherlands, indicating that Leiden is the more precise information about Rembrandt's place of birth.

Shared place names: It is common for places to share the same name. For example according to the TGN, there are 22 places worldwide named London. This problem is less apparent with WordNet due to its limited geographical coverage.

In ArtEquAKT, disambiguation of place names is dependent on their specificity variations. For example after processing the three sentences about Rembrandt, it becomes apparent that he was born in a place named Leiden in the Netherlands. If the last two sentences were not available, it would have not been possible to tell for sure which Leiden is being referred to (assuming there is more than one). One possibility is to rely on other information, such as place of work, place of death, to make a disambiguation decision. However, this is likely to produce unreliable results.

2.5.3. Temporal consolidation

Dates need to be analysed to identify any inconsistencies and locate precise dates to use in the biographies. Simple temporal reasoning and heuristics can be used to support this task. ArtEquAKT's IE tool can identify and extract dates in different formats, providing them as day, month, year, decade, etc. This requires consolidation with respect to precision and consistency. Going back to our previous example from Section 2.5.1, to consolidate the first date (17th century), the process checks if the years of the other dates fall within the given century. If this is true, then the process tries to identify the more precise date. The date in the third sentence is favoured over the other two dates for entry to the ontology as they are all consistent, but the third date holds more information than the other two. Therefore, the third date is used for the instance of Rembrandt. If any of the given facts is inconsistent then it will be stored for future verification and use. Dates are stored internally in machine readable form, textual representations are used in the examples here for clarity. The pseudo-code for temporal consolidation can be found in Fig. 10.

At the end of the consolidation process, the knowledge extracted from the three sentences above will be stored in

```
FOR ALL Date instances in KB
{
    IF Date_1 is more specific than Date_2
    THEN
    {
        consider Date_1 = Date_2 when merging
        use Date_1
    }
}
```

Fig. 10. Pseudo-code for consolidating temporal information.

the KB as the following two triples for the instance of Rembrandt:

- Rembrandt date_of_birth 15 July 1606
- Rembrandt place_of_birth Leiden

2.5.4. Inconsistent information

Some of the extracted information can be inconsistent, for example an artist with different dates or places of birth or death, or inconsistent temporal information, such as a date of death that falls before the date of birth. The source of such inconsistency can be the original document itself, or an inaccurate extraction. Relation cardinality may be used to highlight inconsistencies in the KB. However, relations that are usually regarded to be of single cardinality may actually need to store more than one value, in cases where there is some disagreement in the community about certain facts. For example, Holbein the Elder's date of birth can be 1460 or 1465, depending on whom you believe.

Identifying which knowledge is more reliable is not trivial. Currently we rely on the frequency with which a piece of knowledge is extracted as an indicator of its accuracy; the more times a particular piece of information is extracted, the more accurate it is considered to be. For example, for Renoir, two unique dates of birth emerged; 25 Feb 1841 and 5 Feb 1841. The former date has been extracted from several web sites, while the latter was found in one site only, and therefore considered to be less reliable.

A more advanced approach can be based on assigning levels of trust for each extracted piece of knowledge, which can be derived from the reliability of the source document, or the confidence level of the extraction of that particular information. The knowledge consolidation process is not aimed at finding 'the right answers', however. The facts extracted are stored for future use, maintaining provenance to the original material.

2.6. Document generation in ArtEquAKT

The ArtEquAKT system uses biography templates to arrange the information in the KB into a narrative. It then renders that into a DHTML page so that the personalised biography can be displayed in a web browser.

The structures we use to arrange the story are human authored biography templates that contain queries into the KB. The templates are written in the Fundamental Open Hypermedia Model (FOHM) (Millard et al., 2000) and stored as XML in the Auld Linky contextual structure server (Michaelides et al., 2001). As the templates are stored in a structure server they can be retrieved in different contexts and thus may vary according to the user's preferences and experience.

The fact that fragments of text are associated with facts in the KB is useful as it allows real text to be used in the final biography in preference to generated text. Paragraphs and sentences are extracted by queries to the MYSQL DB. Images are extracted from the KB.

As the attributes of existing text might preclude it from being used, the ArtEquAKT system also allows the KB to be queried directly and basic natural language generation to be used to render them into the biography. This might also be useful for facts in the KB that have been inferred (and for which there is no corresponding text). Here, queries are made directly to the Protégé KB.

The resulting DHTML page is rendered for the user in a standard web browser.

We chose an adaptive representation as this provides a way to expose some of the choices made by the document generator to the reader, without disrupting the selected narrative.

In this section we will explore how we define our story structures and describe the mechanism that we use to adjust them according to varying user contexts. We will also look at how text fragments are selected to populate the structures while minimising repetition. The role of the ontology will be explored both as a vocabulary supporting the querying mechanism and as a part of the decision making process when selecting the fragments.

2.6.1. Biography templates

The ArtEquAKT story templates are human authored and contain queries into both the KB and DB. The FOHM (Millard et al., 2000) was chosen as an abstract representation because it is capable of handling many different types of hypermedia and document structure and has a convenient XML representation. The Auld Linky server (Michaelides et al., 2001) provides a neat HTTP-based query mechanism including the ability to make queries in a specified context, effectively changing the shape of the templates according to the user's preferences and experience. Fig. 11 illustrates a simple document template representing a biography.

Parts of the structure have context metadata attached to it in the form of tag/value pairs (for example 'art knowledge'/'expert'). When the system queries Linky for the template, it specifies metadata describing the user's context (effectively a user profile). When the profile does not match with the context on a part of the structure (e.g. novice versus expert) then that part of the structure is removed.

The template that remains is a high level representation of the story, personalised for the viewer. Each leaf of the structure is a query which resolves into either a statement

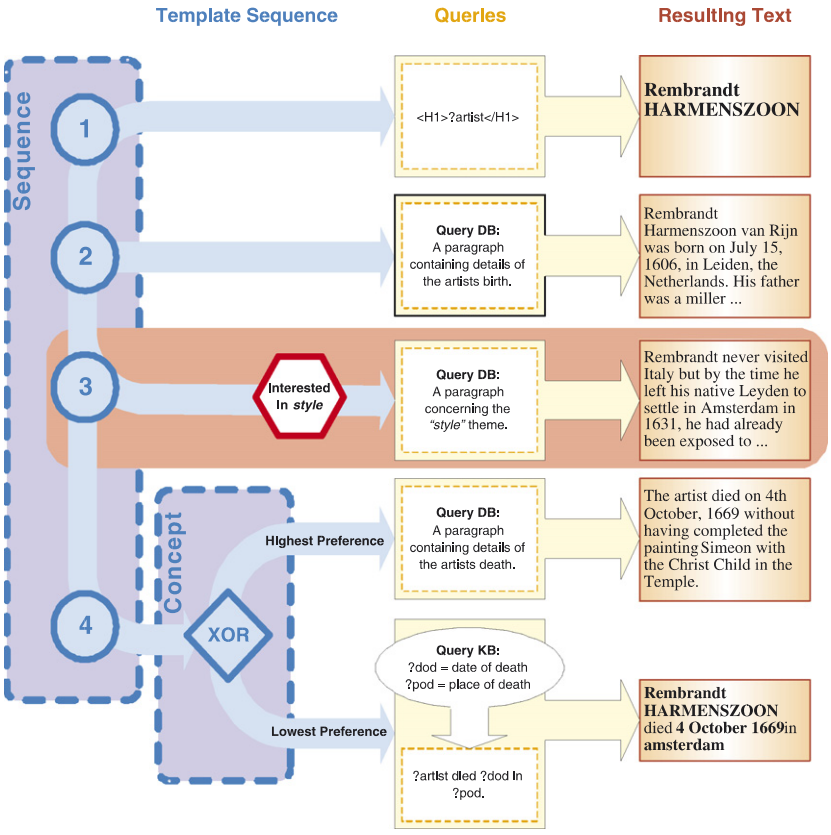

Fig. 11. A simple template structure.

from the extracted information (stored in the KB) or a reference to an original text fragment (stored in the DB). These leaves are organised into a hierarchy of sub-structures. The different types of structure help in the management of the final selection of text fragments.

While FOHM allows us to use an open set of structure types, we have found the following most useful within the story templates:

Sequence—This is the most common structure. It represents a list of queries that should be instantiated and rendered in order. The top level document template itself is normally a sequence.

Set—This represents a collection of queries that should be instantiated and rendered in any order (but with no repetition). It is often used to collect together multiple text fragments returned by a single query (for example, paragraphs about a particular topic).

Concept—These structures are a sub-type of Set. They are used to group alternative queries together, any *one* of which may be successfully used at that particular point in the document. They are also used to store multiple results from a single query where only one result is wanted (for example, where a single representative paragraph is needed about a topic where there might be many suitable paragraphs available).

Level of detail (LoD)—These structures form a hybrid between Sequence and Concept. They are used to group a collection of alternative queries together but in an order such that the text fragment with the highest cardinality will contain the most detail. They are used in the template to indicate queries that have conceptual equivalence but different resolution.

The structures can each include other structures as their members. Certain combinations can be very useful, for example a LoD structure can have a sequence of paragraph queries as its highest order member, but if those fail then it can specify a single factual query of the KB to then construct a suitable sentence as a lower order alternative. Pseudo-code for the generation algorithm based on these template structures is given in Fig. 12.

The different structures and contextual behaviour provide a powerful mechanism for tailoring the documents and also provide flexibility in the instantiated templates (allowing the renderer to make choices without destroying any of the narrative flow encoded into the template).

2.6.2. Querying the KB and DB

The leaves of the template define the type of information which should be included at a particular point in the story. The template leaves can be instantiated with a piece of text extracted from the DB (re-use of an original text fragment), by constructing an original sentence from facts extracted from the KB, or by listing the facts in a more appropriate format, for example the dates of birth and death of an artist are often listed after their name in a title e.g. Rembrandt Harmenszoon Van Rijn (1606–1669). The fact

that fragments of text are associated with items in the ontology is useful as it allows real text to be used where possible in preference to generated text. For simple examples, such as dates and places of birth, sentence generation may be relatively straightforward but where the factual information is more complex the original sentences provide an easy shortcut and will often include additional information that was not extracted as part of the IE process.

```
Variables:
    structure: the template to be processed.
    e1: element in the structure

Repeat:
    foreach element <- structure
    {
        if(element is a substructure)
            recurse
        if(element is a set)
        {
            recurse to process each element
            add each processed element to biography
        }
        if(element is a concept)
        {
            recurse to process each element
            add first returned element to biography
            add remaining elements to alternatives
        }
        if(element is an LOD)
        {
            recurse to process each element
            add first position element to biography
            add remaining elements to expansion set
        }
        if(element is an index query object)
        {
            query database with values given
            if there is a result
            {
                construct structure from results
                recurse to process new structure
            }
        }
        if(element is a protege query object)
        {
            perform query with values given
            if there is a result
            {
                construct sentence and add to biography
            }
        }
    }
```

Fig. 12. Pseudo-code for the generation process.

The query specifies the relation (from the ontology) that is of interest at this point and the instance id that it applies to. This can be looked up in the DB (which acts as a fast index into the KB) and resolved into a set of text fragments. Behaviour metadata attached to the template structures define what is done with the query results, for example. attaching the results to the template in place of the query as either a Set or perhaps a Concept sub-structure.

Fig. 13 shows an example query, which we will examine by way of example. In this case the query (represented by the upper index_query event) will resolve into a paragraph that contains 'date of birth' and 'place of birth' relations for a particular artist identified using the artist ID. The 'forbest' event determines that the fragments retrieved for this query will be placed into a Concept structure.

One of the advantages of using the original paragraphs is that facts that the system fails to extract, and is therefore unaware of, are still included. Since human beings are extremely good at interpreting a narrative smoothly, even when adjacent fragments of text are slightly disjointed, this helps to create a more rounded biography.

However, the fragments have been extracted from existing larger texts and so already contain elements of discourse (focalisation, tense information, etc.). We are currently looking at how we might detect these attributes to ensure that the generated documents are consistent (e.g. to ensure that a document in the third person does not include a paragraph in the first person).

As the document generator populates the template with paragraphs, it keeps track of which paragraphs it has used and the urls from which they originated. This is so that it can avoid repeating itself in cases where a paragraph matches more than one query. Both pieces of information are required as our initial experimentation showed that the same paragraphs often appeared on several different web

```
<binding repeatable="true">
   <behaviour>
      <event>index_query</event>
      <behaviourvalue key="artistid">?artistid</behaviourvalue>
      <behaviourvalue key="keyword0"><![CDATA[date_of_birth]]></behaviourvalue>
      <behaviourvalue key="keyword1"><![CDATA[place_of_birth]]></behaviourvalue>
      <behaviourvalue key="queryname">pars</behaviourvalue>
   </behaviour>
   <behaviour>
      <event>forbest</event>
      <behaviourvalue key="queryname">pars</behaviourvalue>
   </behaviour>
</binding>
```

Fig. 13. A paragraph extraction query.

```
<binding repeatable="true">
   <behaviour>
      <event>protege_query</event>
      <behaviourvalue key="instanceID">?artistid</behaviourvalue>
      <behaviourvalue key="slot">date_of_birth</behaviourvalue>
      <behaviourvalue key="variablename">dob</behaviourvalue>
   </behaviour>
   <behaviour>
      <event>protege_query</event>
      <behaviourvalue key="instanceID">?artistid</behaviourvalue>
      <behaviourvalue key="slot">place_of_birth</behaviourvalue>
      <behaviourvalue key="variablename">pob</behaviourvalue>
   </behaviour>
   <reference>
      <data>
         <datacontent>[INST:?artistid] was born on ?dob in ?pob.</datacontent>
      </data>
   </reference>
</binding>
```

Fig. 14. A sentence construct query.

sites (either due to quotation, or perhaps plagiarism) and thus would appear twice in the DB.

In addition to using the existing text fragments, the system also allows the KB to be queried directly and basic natural language generation to be used to render the results into the document. This is important when brevity is required or is also useful for facts in the KB that have been inferred and for which there is no corresponding text, for example where a date of birth has been extracted from a heading which does not form a complete sentence.

Where facts are needed from the KB, the queries contain variable declarations that are dynamically assigned. These are then added to a blackboard of variables, which the document builder maintains as it traverses the template.

Fig. 14 shows a simple example of a sentence construct query that will build a sentence to say the same thing as the paragraph requested in Fig. 13. The words starting with a question mark (?) are variables that are replaced with the appropriate values from the blackboard. The INST environment tells the generator that the value of the variable is an ID from the KB and needs further resolution (by querying the KB) before it can be included into the sentence.

2.6.3. Using the KB to order fragments

One of the most common problems we have encountered when experimenting with the system is a tendency for the document generator to repeat the same facts in two different paragraphs. This is because paragraphs often contain more than one item of information and the system did not initially keep track of information it unintentionally included in the document, meaning that this information might be included again.

Using smaller text fragments (sentences rather than paragraphs) alleviates this problem somewhat as facts are rarely included unintentionally. However, the use of sentences can result in a terse, overly concise discourse, which is more difficult and less natural to read.

To overcome the problem and enable us to use paragraphs we have leveraged the existing KB of information. Each time a paragraph is added to the instantiating document the list of instance relations associated with the paragraph is pulled out of the KB and stored on a blackboard as a list of triples.

Instance relations are unique statements (for example, we might have the triple [Person_6, 'has_father', Person_15] which describes a relationship between two *Person* instances). As they are unique we never want to repeat them.

Each time a query results in a set of paragraphs, the set of triples for each paragraph is compared with the triples on the blackboard. If there is any overlap then one of the conflicting paragraphs is removed from the set. However, there may still be conflicts between members of the remaining set.

The document generator uses the triple information for the remaining set to produce a list of sub-sets, where each sub-set is guaranteed to contain no conflicting triples. This list is then converted into a LoD structure where each member is one of the sub-sets.

The triple comparison depends heavily on the effectiveness of the extraction algorithms to correctly identify specific relationships in the analysed text. This is because any relations that remain unidentified do not appear in the triples list and therefore cannot be reasoned about.

Once the document builder has instantiated the template, the semantics of the LoD and its member sets is such that the renderer will not use any of the conflicting paragraphs in the same discourse. However, the ArtEquAKT renderer does not just produce a flat document, instead it uses the various structures to create an adaptive document that exposes all of the matching text to the readers, but in a well-defined, interactive manner.

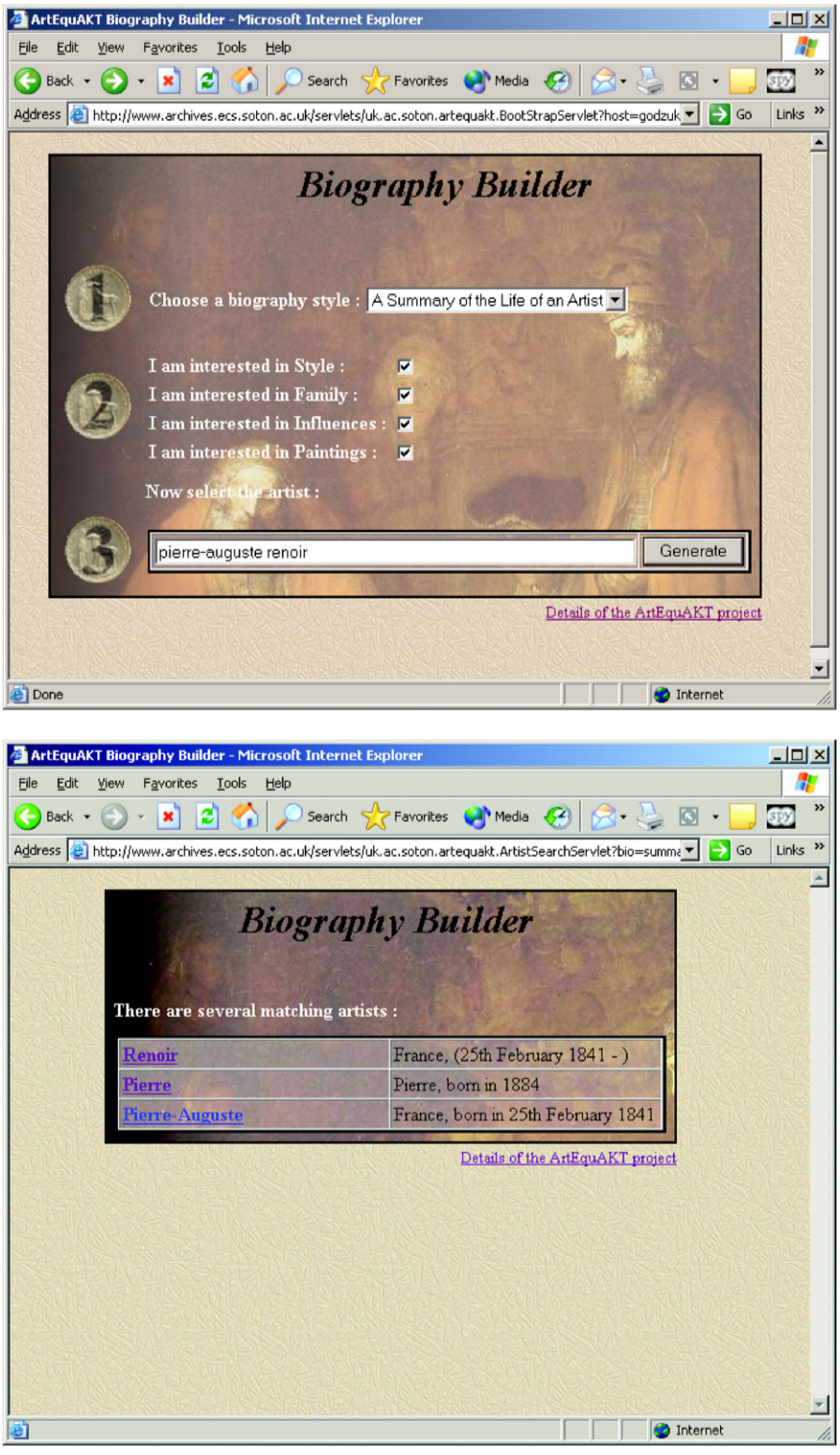

Fig. 15. Screenshots of the web interface.

2.7. The user interface

The ArtEquAKT server takes requests from a reader via a simple web interface. Fig. 15 illustrates a typical users interaction with the ArtEquAKT system. At the top is the initial screen, where the user enters the name of the artist they are interested in, selects a type of biography to generate (chronology, summary, etc.) and also enters any preferences for the biography, for instance stating that they are not interested in the artist's personal life.

If there is more than one artist that matches the search criteria, the user is presented with a choice, shown in the

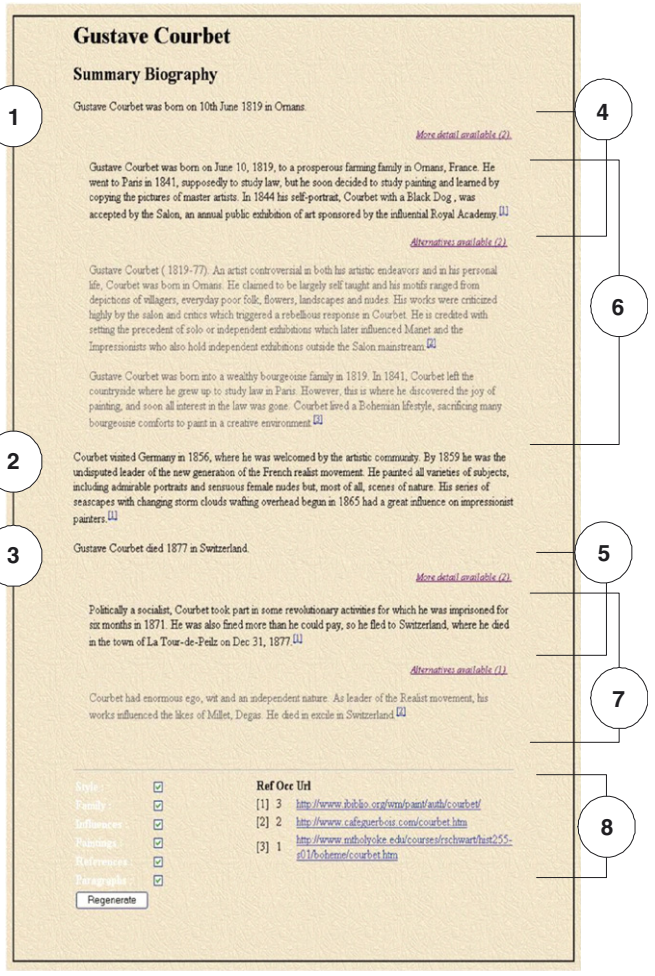

Fig. 16. An adaptive document.

second screen shot. Should there be no artist that matches the query, the user is asked if they would like to initiate a web search on that artist. This would launch the knowledge extraction process described in more detail previously. Finally, once the user has selected an artist the generated biography is displayed for them (see Fig. 16).

2.7.1. Applying adaptive hypermedia techniques

The adaptive hypermedia (AH) community has, over the years, devised a number of adaptive techniques that can be applied to the presentation of and interaction with hypermedia documents. Brusilovsky created a taxonomy of these techniques in 1996 (Brusilovsky, 1996), which he subsequently updated in 2001 (Brusilovsky, 2001).

The taxonomy focuses on the interface and user interaction, and has been divided into two distinct areas: adaptive presentation, the presentation of information based on user preferences, and adaptive navigation support, hyperlink presentation and generation. Brusilovsky foresaw a number of possible applications for these techniques, including that of electronic encyclopedias (for example, the PEBA-II system, Milosavljevic, 1997, can be seen to use some of the techniques). As part of the document generation we have implemented a number of the AH techniques described by the taxonomy.

When the document is being constructed, the renderer can make choices about which parts of the FOHM story structure it wishes to use. Fragments that have not been used in the initial rendering can be made available via some of the adaptive techniques.

2.7.1.1. Selecting fragments. In previous work we have criticised Brusilovsky for treating natural language adaptation as a separate branch of his taxonomy, independent of other content adaptation techniques (Bailey et al., 2002). We have evidence of this here, where in Fig. 16 the template has selected fragments 1,2 and 3 as part of a larger attempt to generate a natural language document. The renderer will be presented with a number of paragraphs which it can use that all convey the same information according to the KB in the form of a concept structure. Since the system considers all the fragments to contain similar information. The render just picks one of the fragments in the concept structure.

2.7.1.2. Dimming fragments. Where the renderer has selected one of the fragments from a concept structure, the remaining fragments are initially hidden from the reader (in Fig. 16 fragments 4 and 5) with a link provided which allows the reader to reveal the dimmed fragments (6 and 7). This can be useful to a reader as, although the system believes that these fragments are similar, there may be additional information that has not been identified during the extraction process.

2.7.1.3. Stretchtext. Where the renderer receives information in an LoD structure, it will display the first fragment in the list (the one with the least information). A link is provided to expand the LoD structure (in Fig. 16 this has occurred with fragments 1 and 3). When the link is selected, more of the structure is placed in-line in the document for the reader (fragments 4 and 5). This process can be repeated until all of the fragments in the LoD structure are revealed. This form of adaptation is known as stretchtext.

2.7.1.4. Generating links. The ArtEquAKT system maintains appropriate references to all the source material that it analyses. When this is used verbatim in the generated documents, a link is created to the original web resource and inserted at the end of the text fragment. Where multiple fragments from the same resource are used, the references are enumerated, collated, and presented at the end of the document (section 8 in Fig. 16).

3. Discussion and evaluation

The ArtEquAKT system presented in this paper could be evaluated in a number of ways. The emphasis in the project was on connecting various techniques into a process chain

that enabled extracted information to be consolidated and recomposed into web page biographies of artists in different contexts. As such, our focus in evaluating the system has been on the success of the integration of the various technologies and the suitability of ontologies as a co-ordinating structure. The IE tools were largely taken 'off the shelf' and as such do not represent new developments in the field in themselves although their integration and application in the way described previously is itself novel. A brief evaluation of the extraction tools is presented in the section below, followed by a section discussing the effectiveness of the consolidation tools. Some qualitative evaluation of the generated biographies is then covered and finally a more general discussion of the architecture is presented.

3.1. IE evaluation

We used the system to populate the KB with information for five artists, extracted from a total of around 50 web pages. Precision and recall were calculated as percentages for a set of 10 artist relations (listed in Table 2).

Recall is taken to be the number of facts of the given type that the system correctly extracted compared to the number of facts of that type that were actually in the document. Where recall is listed as N/A there were no facts of the given type in the selected documents. Precision refers to the percentage of facts extracted by the system of a given type that did indeed turn out to be facts of that type. Where no facts were extracted the precision will be listed as N/A. As is the case with the place of work of Courbet, the system extracted what it thought were facts about place of work but missed those facts in the document that did indeed refer to the place of work. Here then, the recall was 0 and the precision also 0 as those facts extracted turned out to be false positives.

The experiment results given in Table 2 shows that precision scored higher than recall with average values of 85 and 42, respectively.

Inaccurately extracted knowledge may reduce the quality of the systems output. For this reason, our extraction rules were designed to be of low risk levels to ensure higher extraction precision. Advanced consistency checks could help to identify some extraction inaccuracies; e.g. a date of marriage is before the date of birth, or two unrelated places of birth for the same person!

The preference of precision versus recall could be dependent on the relation in question. If a relation is of single cardinality, such as a place of birth, then recall could be regarded as less significant as there can only be one value for each occurrence of this relation. A single accurate capture of the value of such a relation could therefore be sufficient for most purposes. However, multiple cardinality relations, such as places where a person worked, can have several values. Higher recall in such cases could be more desirable to ensure capturing multiple values. One possible approach is to automatically adjust the risk level of extraction rules with respect to cardinality, easing the rules if cardinality is high while restricting them further when the cardinality is low.

In Table 2, Goya is an example where only a few short documents were found. The amount of knowledge extracted per artist could be used as an automatic trigger to start gathering and analysing more documents.

As would be expected, the nature of the sources had a large bearing on the effectiveness of the IE process. Our IE process is designed to parse text, so the worst documents to analyse tended to be those that were heavily structured. Information in HTML tables or lists was not always successfully extracted. The system was also occasionally confused by more complex punctuation structures in titles and headings, for example *Gustave Courbet (France 1819–Switzerland 1877)* did not conform to our more simple extraction rules. The fact that the structuring is often concerned with presentation rather than necessarily the organisation of content means that for the IE process the additional information is simply an added complication. As we move towards information sources that are structured to indicate more semantic information, this may help IE processes.

The style of writing also had a large affect on the extraction process, with more complicated sentence structures obviously requiring more complex extraction rules. The use of anaphora (it, he, she, etc.) posed difficulties for IE. Finally, our IE was geared towards the extraction of triples from single sentences. Achieving the extraction of triples across multiple sentences is much harder.

Better results would also have been achieved if the KB had been prepopulated with, for example, a gazetteer of artists names. By bootstrapping the KB, the IE processes would have had a better idea of what they were looking for. The lack of boot strapping also impacted on the consolidation process as there was no 'trusted' list to base the consolidation on.

3.2. Consolidation evaluation

Table 3 shows the reduction rate in number of instances and relations after consolidating the KB. Applying the heuristics described earlier in the paper led to the reduction in number of instances of the Person and Date classes by 90% and 64%, respectively. Before consolidation, 283 instances representing Rembrandt were stored. The un-

Table 3
Consolidation rates.

Class	Before consolidation	After consolidation	Rate %
Person instance	1475	152	−90
Date instance	83	30	−64
Place instance	30	505	+94
Person relations	4240	1562	−63

ique-name consolidation heuristic was the most effective with no identified mistakes.

When place instances are fed to the KB, they are expanded using WordNet and stored alongside their synonyms, holonyms (part of), and meronym (sub parts). The number of Place instances created in the KB has therefore increased significantly (94% rise). This gave the consolidation the power to identify and consolidate relationships to places as described in the geographical consolidation section. Some instances (mainly dates) were not consolidated due to slight syntactical differences, e.g. "25th/2/1841" versus "25/2/1841". This highlights the need for an additional syntactic-checking process that could eliminate such noise.

Table 4 gives a more detailed picture on the consolidation results with respect to the instances created for the five artists listed in Table 2. The table gives the total number of instances the system created for each artist based on the information it received from the extraction process. The number of instances that remained after applying the consolidation process is given in the third column. For example, the 346 instances created for Rembrandt boiled down to only 3. The instances that remained in the KB after consolidation are listed in the fourth column. The last column gives some comments and explains the reasons why an instance remained separate (i.e. was not merged into the main instance for the relevant artist).

Table 4
Detailed consolidation results

Artist	Extracted/ consolidated	Remaining instances	Comment
Rembrandt	346/3	Rembrandt Harmenszoon	Main instance
		Rembrandt van Rijn	Slight name mismatch and insufficient information overlap with main instance (only year and place of death, and year of marriage are known)
		van Rijn	Name was not matched with Rembrandt
Renoir	250/2	Renoir	Main instance. IE failed to extract the full name for Renoir
		Pierre-Auguste	Name was not matched with Renoir
Cassatt	196/2	Mary Stevensen Cassatt	Main instance
		Cassatt	Name mismatch and insufficient information overlap with main instance (only date of birth is known)
Goya	97/3	Fransisco Jose de Goya	Main instance
		Fransisco Goya	Slight name mismatch withinsufficient information overlap (only place of death is extracted)
		Goya Fransisco de Goya	Name mismatch due to IE error. Insufficient information overlap (only place of death is extracted)
Courbet	80/2	Gustave Courbet	Main instance
		Courbet	Insufficient information overlap (only date of birth)

We can here see the effect of the order in which the consolidation stages are applied on the results of this process. As mentioned earlier, our consolidation starts with the name of the artist first. This will lead to merging all artist instances that bear the exact same artist name. Secondly, instances with similar, but not identical names will be analysed for information overlap. For example *Rembrandt van Rijn* is similar to the full name of the artist (the main instance) and hence will be compared against the main instance with respect to certain attribute values. The two instances will be left separate if there is insufficient knowledge to judge whether they are duplicates or not.

In some cases, the instance name for the artist is *too* different from that of the main instance, such as the case with *van Rijn* (i.e. no mention of Rembrandt). Such instances will not be merged with the main Rembrandt instance in spite of whether there is sufficient information overlap or not. This would have not been the case if the consolidation process ignores the artist names when comparing their instance for information overlap. However, this may result in merging any two artists that happened to have, for example, the same date and place of birth.

One possible strategy to deal with situations like the above is to increase the amount of minimum information overlap required for two instances to be merged *if* they have a greater deal of name mismatch. In our examples, this would have allowed *Renoir* and *Pierre-Auguste* to be merged.

3.3. Biography generation evaluation

The consolidated knowledge was used to guide the generation of biography documents. It is difficult to evaluate the biographies separately from the extraction and consolidation on which they so heavily depend, so here we shall reflect on the suitability of templates and our use of AH techniques.

In general the template approach seems to have been effective. The biographies generated from our test summary template ranged in size from being constructed from three paragraphs from two different sources to being composed of 21 paragraphs derived from 10 different sources. Overall the biographies averaged 10 paragraphs from five different sources.

However, templates are not very flexible and in cases where there is little information they tend to become underpopulated (one or two paragraphs) and lose any sense of narrative cohesion. Perhaps a better approach would be to ask the user to make more abstract choices and then map that onto a template given the quantity of

information available. This would ensure that the biographies degraded gracefully into less and less sophisticated structure.

In terms of selecting paragraphs to fill slots in a template there were persistent issues with facts that are repeated (sometimes several times) in the document. This repetition disrupts the reading experience and draws attention to the automated nature of the writing. There were several causes, some more easily remedied than others.

Some sites use exactly the same text (perhaps because pages have been quoted or plagiarised) such that two different paragraphs from different sites might be identical. This means that it is not sufficient to check for repeating paragraph instances, instead a simple text comparison must be used to catch different instances that have the same text.

Dealing with multiple paragraphs that have different text but contain the same facts is a more complex aspect of the same repetition problem and is related to that of choosing between different paragraphs that fill a given slot. Currently the system will choose the paragraph that contains the least information beyond the facts that are wanted. For example, to fill a 'date of birth' slot a paragraph that describes the date of birth will be chosen over one that contains the date of birth *and also* some other information. This is to decrease the likelihood of a clash later down the document.

Paragraphs that contain only the required information are not always available so the system will choose the best case and then record the facts that it has inadvertently included. For example, if a paragraph is chosen because it mentions the date of birth but it also contains details about the artists marriage then this is recorded, if the template later requires a paragraph describing marriage then the system knows it already has inserted this information and does not include another paragraph (i.e. does not repeat itself). Unfortunately this is not very effective because of the system's low accuracy in identifying facts in paragraphs. This accuracy is high enough to extract the information from a set of sources, but not high enough to successfully mark all facts. This means that the system does not know that it has already included some bits of information and thus repeats itself despite the algorithm described above.

A complete repetition algorithm would not make definitive paragraph decisions until it had parsed the entire template and constructed a set of alternative paragraph arrangements which it would then choose based on some preference metric (i.e. a preference for including facts in their slots over a more concise narrative that contains the same information but in a more flexible order). Due to the accuracy of the extraction system it was decided that such an algorithm would not be worth pursuing in this version of the system.

In terms of the document presentation the use of AH seemed to be very successful, not only as a reading aid (for example to open up more details if the reader chooses) but also as a way of providing an audit trail for the knowledge (as all paragraphs are sourced it is possible to compare what different sites say on a particular topic). They also help to bypass some of the failings of the extraction system in spotting facts, as users who are looking for particular information can browse the generated document and may find it even if the system did not spot it and explicitly display it to them.

3.4. General evaluation

Our main aim with the ArtEquAKT system has been to investigate the possibilities of coupling IE, consolidation and document generation through the use of ontologies. We have not yet carried out extensive user trials on the system to evaluate in detail its user interface, or the perceived quality of the biographies being generated by it. However, we have analysed the biographies generated from the five artists previously discussed and can make some broad statements about the biographies that the system is capable of generating.

By extracting paragraphs as well as explicit relations and facts, the generated biographies do appear better than might be assumed from the previous IE evaluation. Often information not explicitly extracted may be presented in a biography paragraph which has been selected based on the simpler keyword matching.

For example, in the IE evaluation section above (3.1), we saw how the system was unable to successfully extract detailed marriage information about the five artists yet was able to supply paragraphs which included this detail. A brief examination of this effect shows how complex the problem can actually be and the reason for combining the cruder cut of keyword recognition with the more specific, yet more problematic, fact extraction.

For Renoir, the system detected the date of his marriage but was unable to identify the name of his spouse. The more complex sentence construction observed in the paragraph about marriage in the generated biography may give us a clue as to why the name was not extracted.

> 'In 1880, he met Aline Charigot, a common woman, whom he would marry in 1890, they had 3 sons...'

The phrase 'marry in 1890' was clearly extracted but not linked with the name previously stated. Co-referencing information is extracted but the linking of related clauses is only carried out at a basic level currently. More complex extraction techniques could be used to resolve this problem.

For Mary Cassatt, the system thinks that she was married but was unable to establish to whom or when. In actual fact, she never married and it may be the following sentence in an analysed document that caused the confusion.

> 'Despite the concerns of her parents, Cassatt chose career over marriage, and left the United States in 1865 to travel and study painting in Europe.'

With Goya, it is again the construct 'whom' that causes problems:

> 'Bayeu was also the brother of Josefa Bayeu, whom Goya married in 1773.'

Later on in the biography the system supplies a paragraph which it incorrectly thinks concerns the death of Goya based on simple keyword matching but turns out to be about the death of his friend.

> 'Later, after the death of his friend and brother-in-law Francisco Bayeu, he took over his duties as Director of Painting in the Royal Academy from 1795 to 1797, when he resigned due to ill health...'

These issues stand out when comparing the generated documents to their templates, but in normal reading are obscured by the use of whole paragraphs and only disrupt a reading when there are obvious markers in the paragraphs that do not fit in with the surrounding material (for example, a date out of chronological sequence). Otherwise the biographies generated by the system are quite readable, even if sometimes identifiable as automatic constructs.

The inevitable comparison to be drawn would be between the biographies generated by the ArtEquAKT system and those that could be found by carrying out a web search using Google.

Firstly, it is worth reiterating that we are not expecting our generated biographies to provide a comparable reading experience to those that are hand written. ArtEquAKT biographies do suffer from repetition, although AH techniques have significantly reduced this by providing selections of alternative paragraphs where it can tell there is significant overlap in content. In the current system there is no attempt to deal with co-referencing although the information is collected during the extraction process. There is also no smoothing of the narrative in terms of tense or style which can occasionally be jarring. And although our templates attempt to provide a broad brush chronology (birth, marriage, death), inevitably the paragraphs do not always reflect a smooth chronology that would occur in a hand crafted biography.

Where we are able to observe some benefits of the ArtEquAKT system over Google is in the consolidation of information and the presentation of that consolidation to a human reader. A Google search for 'Goya biography' throws up many results. Just looking at the facts we are initially extracting we can see that the first result listed contains his birthplace but not a full date and only a date of death. The second result provides places and dates for his birth and death but no other information. The third result is a very long and comprehensive biography but even this does not contain the date of his marriage. So the objective of consolidating multiple sources of information into one place appears to be met. This is harder to show with Rembrandt and Renoir as the web already contains very many comprehensive biographies of these artists but we can at least claim to avoid the need to wade through false hits on art shops, guest houses and even dentists (unbelievably, www.rembrandt.com is actually Rembrandt Oral Care).

4. Background and related work

An ontology is a shared conceptualisation of a specific domain in a machine–understandable format (Guarino and Giaretta, 1995). Ontologies will play a major role in deploying the Semantic Web, by facilitating knowledge representations, inference, sharing, etc. (Berners-Lee et al., 2001). The use of ontologies to support knowledge extraction has been previously touched upon (Vargas-Vera et al., 2001; Handschuh et al., 2002) and indeed statements about their usefulness have been made (Fensel, 2001). Nevertheless the full potential of this approach is not yet explored.

4.1. Information extraction

IE can be broadly described as the extraction or pulling out of pertinent information from volumes of texts. This could involve either the extraction of factual information or the use of summarisation techniques.

Traditional IE tools rely on templates to direct the extraction process. The aim of these templates is to provide the IE process with limited representations of the concepts and relations of interest and restrict the IE search and extraction to a defined vocabulary. This vocabulary normally contains concept types and names, synonyms, verbs, etc. Templates represent the basic ontological facts that an IE system is supposed to extract. However, templates normally lack the required infrastructure for hierarchical and conceptual reasoning.

Templates are sometimes built by manually stripping down an existing ontology (Vargas-Vera et al., 2001). The detachment of the IE process from the ontology itself reduces the amount and level of possible reasoning and inference. Maintaining a direct contact between the IE tools and the ontology may improve the extraction performance. However, this may require the ontology to be extended with additional vocabulary to satisfy the needs of the IE process, such as synonyms and alternative terms that are often found missing or hard to access in ontological representations.

IE tools often deploy a set of extraction rules to identify and extract the entities of interest (e.g. rules to extract person names, others to extract organisations). Such rules are often handcrafted (e.g. GATE, Cunningham et al., 2002b) or learnt semi-automatically in training sessions (e.g. Melita, Ciravegna et al., 2003). The function of these rules is to identify and classify terms within sentences based on the lexical construction of these sentences. Extraction rules are often designed to extract generalised classifications of terms (e.g. van Gogh is a Person). Extracting more specific classifications (e.g. van Gogh is an Artist) requires

a more complex analysis. Using an ontology to support the extraction rules and to help extracting more specific classifications has been explored in Popov et al. (2003).

Most traditional IE systems are domain dependent due to the use of linguistic rules designed to extract information of specific content (compare the systems participating in the evaluation campaigns of the Message Understanding Conference (MUC) as described in Grishman and Sundheim, 1996, text corpuses, earthquake news White et al., 2001, sports matches Reidsma et al., 2003). Adaptive IE systems (Ciravegna, 2001) can ease this problem by identifying new extraction rules induced from example annotations supplied by users. However, training such tools can be difficult and time consuming. Promising results are offered by more advanced adaptive IE tools, such as Armadillo (Dingli et al., 2003), which discovers new linguistic and structural patterns automatically, thus requiring limited bootstrapping.

We currently use IE techniques to extract knowledge directly from unstructured web documents. Where the ArtEquAKT system can take advantage of existing annotations to retrieve the knowledge it requires it will, but currently annotations are rare and will most likely not be rich or detailed enough to cover all the knowledge contained in these documents. Annotating existing web documents forms one of the basic barriers to realising the Semantic web (Kahan and Koivunen, 2001). Manual annotation is impractical and unscalable, while automatic annotation tools are still in their infancy. Hence advanced knowledge services may require tools able to search and extract the required knowledge from the web, guided by a domain conceptualisation (ontology) that specifies what type of knowledge to harvest. Previous work on annotation has demonstrated the value of coupling natural language processing (NLP) with ontologies (Vargas-Vera et al., 2001; Maedche et al., 2002). The ontology can guide the annotation task by restricting it to a specific domain and providing it with knowledge inference and conceptual browsing facilities (Maedche et al., 2002). An ontology-based approach for annotation needs to deal with a wide range of issues such as the problems of duplicate information across documents, managing ontology change, and redundant annotations (Staab et al., 2001). Several tools have been developed based on IE systems to semi-automate the process of document annotation (e.g. MnM, Vargas-Vera et al., 2002; Melita, Ciravegna et al., 2003; OntoMat, Handschuh et al., 2003). These tools help to annotate web pages with respect to ontologies. Such annotations can be added as instantiations in the ontology. Most ontology languages allow multiple inheritance which permits concepts to be derived from multiple parents, thus giving polysemy a more flexible structure. It also provides flexible term expansion through hierarchies.

4.1.1. Relation extraction

The task of relation extraction is to extract pre-defined relation types between two identified entities. Techniques such as probabilistic methods or machine learning (e.g. Inductive Logic Programming, ILP) are often applied as well as simple linguistic analysis. In addition, systems like (Katz, 1997; Litkowski, 1999) made use of such semantic relations in retrieving answers in a response to natural language questions demonstrating the benefits of exploiting structural information about sentences in establishing linkages between words.

Roth presented a probabilistic method for recognising both entities and relations (Roth and Yih, 2002). The method measures the inter-dependency between entities and relations and uses them to restrain the conditions under which entities are extractable given relations and vice versa. An evaluation showed over 80% accuracy on entities and a minimum 60% on relations. However, the computational resources for generating such probabilities are generally intractable. Aitken (2002) applied ILP to learn relation extraction rules where associated entities are symbols (e.g. 'high', 'low'). It is more concerned with discovering hidden descriptions of entity attributes than creating binary relations between two entities which we are interested in. REES, developed by Aone and Ramos-Santacruz (2000) is a lexicon-driven relation extraction system aiming at identifying a large number of event-related relations. Similarly to the approach here, it depends on a verb for locating an event-denoting clue and uses a pre-defined template which specifies the syntactic and semantic restrictions on the verb's arguments.

4.2. KB population

Storing knowledge extracted from text documents in KBs offers new possibilities for further analysis and reuse. Ontology instantiation refers to the insertion of information into the KB, as described by the ontology (sometimes referred to as ontology population). Instantiating ontologies with knowledge is one of the important steps towards providing valuable ontology-based knowledge services. Manual ontology instantiation is very labour intensive and time consuming. A number of approaches have been studied to speed up this process using a variety of techniques, such as using IE on raw text, harvesting information from structured documents, gathering knowledge from existing annotations, accessing online DBs and gazetteers, etc. These approaches come with different strengths and weaknesses.

Relying on existing annotations to instantiate a *specific* ontology may ensure fast access to good quality knowledge as it had already been curated and made available in a semantic format. However, as mentioned earlier, annotations may not exist in sufficient amounts for the desired type of knowledge. Furthermore, annotations are most often crafted based on different ontologies, which may require some effort to map to local domain representations. Exposé (Luke et al., 1997) is an example of a system developed to build a KB from online information encoded in SHOE (Luke et al., 1996).

KIM (Popov et al., 2003) is an annotation framework which relies on an ontology that has already been instantiated with large amounts of general purpose data (e.g. locations, organisations, people names). These instantiations were derived from a set of existing online DBs and gazetteers. Such online resources often contain a large amount of good quality data, and hence can be suitable for ontology bootstrapping. However, online DBs and gazetteers cover certain areas of knowledge which might not be of interest to some ontologies. KIM applies IE techniques to get hold of additional knowledge.

IE techniques are suitable for extracting knowledge from text documents regardless of structure or annotations. However, IE is usually domain dependent. Craven et al. (2000) instantiated an ontology with knowledge extracted from web pages using IE methods that have been trained to extract specific types of information. The SemTag system (Dill et al., 2003) uses the TAP (Guha and McCool, 2003) KB to locate and annotate instances within web pages. However, SemTag is not capable of identifying new instances, but rather relies on a pre-instantiated ontology.

The PANKOW system Cimiano et al. (2004), uses an unsupervised, pattern-based approach to identify the type or category of an instance (e.g. England is a Country). This system searches Google for linguistic patterns made from pairing an extracted instance with a class in the ontology. The system then decides which category to select based on the number of hits returned by Google when searching for that specific pattern. Note that PANKOW does not attempt to extract relations between instances, which is one of the roles of ArtEquAKT. Furthermore, it is not clear whether PANKOW's approach applies to ArtEquAKT's domain of artists (searching for the pattern 'Rembrandt is an artist' returns less results in Google than for 'Rembrandt is a place').

Scraping text from well-structured web pages forms another approach for instantiating ontologies from web documents (e.g. Snoussi et al., 2002; Davulcu et al., 2003). Such approaches are useful for extracting large quantities of information from domain specific pages. However, such tools can only extract from well-structured documents. Change in the structure of documents often results in a considerable change in the harvesting scripts to maintain their functionality. Other approaches (e.g. Dingli et al., 2003) make use of induced wrappers to extract from less-structured web pages.

4.2.1. Knowledge consolidation

Automatically instantiating an ontology from diverse and distributed resources poses significant challenges. One persistent problem is that of the consolidation of duplicate information that arises when extracting similar or overlapping information from different sources. Tackling this problem is important to maintain the referential integrity and quality of results of any ontology-based knowledge service. Reidsma et al. (2003) relied on manually assigned object identifiers to avoid duplication when extracting from different documents. Little research has looked at the problem of information consolidation in the IE domain. This problem becomes more apparent when extracting from multiple documents. Comparing and merging extracted information is often based on domain dependent heuristics (Radev and McKeown, 1998; Reidsma et al., 2003; White et al., 2001). Dill et al. (2003) relied on statistical measures to disambiguate instances. Our approach attempts to identify inconsistencies and consolidate duplications automatically using a set of heuristics and term expansion methods based on WordNet (Voorhees, 1998).

4.3. Document generation

While a KB with a defined ontology will ease problems of machine interaction, many applications will be attempting to sort, arrange and present information to people. Ontologies are appropriate vocabularies for machines, but human beings need a more natural interface.

Story telling provides a simple, intuitive mechanism for presenting such information. There is a great deal of existing work regarding narrative, both critical and philosophical, which may be drawn on to assist the construction of a story from 'raw' structured facts.

We can consider ontologically structured information (in this case extracted and consolidated from the web into a KB) as the underlying story, waiting to be told. The fragments of text in the KB can be re-ordered and combined with generated sentences to produce an eventual discourse, personalised to a particular reader and drawing on many different published sources.

Previous work in the area of dynamic story generation has highlighted the difficulties of maintaining a rhetorical structure across a dynamically assembled sequence (Rutledge et al., 2000). As a consequence of this, there has been a focus on dynamic presentation decisions as opposed to narrative ones (Mancini, 2000). Here, the narrative language of cinema has been used as a mechanism to maintain coherence in hypertext narratives. Where dynamic narrative is presented it has often been based around domain specific story-schema such as the format of a news program (a sequence of atomic bulletins), which has a dependable rigid format (Lee et al., 1999). By setting out the narrative structure in advance, the system does not require the same depth of understanding of the individual narrative components collected.

4.4. Related systems

By its very nature as an integrating project, influences can be derived from the various areas that are being integrated. In this section we will look at ontologies in general and systems that use ontologies in similar ways to the ArtEquAKT system. Related work specific to the component parts of the architecture will be discussed in more detail in the relevant later sections.

The closest work we found to ArtEquAKT is in the area of text summarisation. A number of summarisation techniques have been described to help bring together important pieces of information from documents and present them to the user in a compact form.

Even though most summarisation systems deal with single documents, some have targeted multiple resources (Radev and McKeown, 1998; McKeown et al., 2002; White et al., 2001). Radev and McKeown (1998) developed the SUMMONS system to extract information and generate summaries of individual events from MUCs text corpora. The system compares information extracted from multiple resources, merges similar content and highlights contradictions. However, like most IE based systems; information merging is often based on linguistics and timeline comparison of single events (e.g. Radev and McKeown, 1998; White et al., 2001) or multiple events (e.g. Reidsma et al., 2003).

The Topia project (TOPic based Interaction with Archives) (Rutledge et al., 2003) generates documents based upon discourse structures derived from the underlying domain semantics. Unlike the ArtEquAKT system, the starting point for Topia is a semantically annotated set of material (Paintings in the Rijksmuseum) which is then formed into documents by clustering material and organising it into discursive segments. Like ArtEquAKT the aim was to produce documents that were coherent, plausible and hopefully pleasant for the reader. Topia does not have an IE component, as the knowledge is already marked up, defining the concepts and their metadata using RDF. The documents are constructed dynamically using clustering techniques and ordering based around discourse derived from the metadata. There are no pre-defined templates as in the ArtEquAKT system.

The MIAKT (Medical Imaging and Advanced Knowledge Technologies) project aimed to apply the capabilities of the knowledge management and the intelligent analysis and handling of medical data to collaborative medical problem solving in the domain of breast cancer screening and diagnosis (Dupplaw et al., 2004). The research focussed on the use of ontology services combined with annotation and enrichment services in order to support Multi-Disciplinary Meetings between medical practitioners with different expertise. The supporting tools used a common ontology for annotation and sharing of data, trying to encapsulate the overlapping domains involved. In addition, reasoning and GRID-services were used to augment the activities.

The KnowItAll system developed at the University of Washington is constructing a large DB of facts extracted automatically by web crawing (see Downey et al., 2005). The system is domain independent and operates autonomously.

5. Conclusions and future work

The system discussed here integrates a variety of tools in order to automate an ontology-based knowledge acquisition process and maintain a knowledge base with which to generate customised biographies.

The ontology is placed at the heart of the system, thus controlling incoming and outgoing knowledge from and to the system components. The IE component uses the vocabulary of the ontology to guide the extraction process. The extraction process tries to match words in web documents with class and relation names in the ontology to determine how to extract from a sentence and where to insert the extracted triple in the ontology.

Skeleton rules for sentence-based relation extraction are currently handcrafted within the system at design-time, which are then executed automatically at run-time. However, these rules are dynamically linked to the relation labels in the ontology. In order for this to work, the ontology terminology has to be set in a clear fashion, thus avoiding obscure relation and class names. It would also help if the labels of relations in particular consist of one main term (e.g. *born_in* instead of *location_of_birth*). This enables the extractor process to easily identify the verb or noun to check for in the text documents.

The initial idea of ArtEquAKT was to build an ontology-independent knowledge extraction system. However, in practice there will always be some element of domain dependency when it comes to information extraction, generation, and consolidation. Nevertheless, certain types of knowledge are not limited to any specific domain, such as personal information, dates of personal events, locations, etc. Such knowledge will be treated more or less the same no matter if the person in question is an artist, a painter, a football player, or a politician. On the other hand, information about paintings is domain specific (domain of art), and thus requires special treatment when it comes to identification and extraction (e.g. identifying names and styles of paintings).

As an example we could replace the current artist ontology with a researcher ontology, where the extraction is expected to focus on information about research activities. Since the relation extraction between entities in our artists ontology is mostly determined by the main type of verb used in the source text, we can expect similar extraction performance from a research ontology if the research activities are also identifiable from such a main verb. With respect to entity recognition tools, domain specific entities (e.g. publication styles) need specialised extraction rules that have to be modified when the domain changes.

The important point is that the system is flexible enough to accept new extraction rules and can adapt to some extent to changes in the ontology without having to regenerate affected rules. For example if the relation *date_of_birth* was changed to *born_on* in our ontology, then the extraction process will not be affected because both terms (birth and born) are linked in WordNet which is heavily used by ArtEquAKT for term expansion.

Our consolidation tools also make use of the ontology relations to consolidate duplicate instances. For example,

multiple instances of the same artist can be consolidated if they have common attribute values as described by the ontology. For this to work the ontology must clearly identify relations that may accept multiple values (e.g. places visited) and those of singular values (e.g. places of birth). But in some cases, conflicting information is extracted from the web, and hence there will be a requirement to store multiple values for such relations, even if they normally take singular values.

Domain knowledge helps in consolidation. For example, paintings are considered the same if they are painted by the same artists with the same name *and* date. Gazetteers can also be used to support consolidation. If a gazetteer of artist names is available then the consolidation process can check whether there is a unique Rembrandt or not, and if so then all Rembrandts can be merged together. More work needs to be done to develop more knowledge dependent consolidation methods.

Even after the knowledge base has been built using the extraction and consolidation processes, the ontology remains as a unifying presence. The templates used for biography generation contain queries into the paragraph database indexed using the terms of the ontology. They also contain direct queries into the knowledge base which are used to generate simple sentences in cases where paragraphs are not available or not required. The templates are thus implicitly structured around the ontology and the types of relations that have been modelled (family, marriage, etc.)

Currently extensions to the ontology require manual additions to the templates in order to be reflected in the biographies. A more integrated approach would be to represent the templates inside the ontology, so that sections of the template would be directly related to the classes in the ontology with whose instances they would be populated. As an example this would mean that if the family classes were extended to included uncles and aunts then this would be automatically reflected in the templates and thus the biographies.

In general the ontology provides a point of focus and cohesion throughout the system, even though the actual component parts are loosely coupled. The ontology not only provides a common vocabulary but also a common understanding of what relations might exist that is drawn on at every stage.

Further work will look at removing the hand crafted elements from each stage of the system and deriving more information from the ontology itself. This will facilitate a more generic system, where changing the ontology is all that is required to change the domain. Once this has been completed, more detailed user evaluations of the system can be carried out as the project moves forward from evaluating the individual components and their interface with the ontology through to more overarching evaluations of the biographies produced by the system as a whole.

As the Semantic Web grows, changing an ontology will increasingly become a job of collation and integration rather than authorship. We are not suggesting that using an ontology as the central part of this kind of extraction and generation system is a magic solution, but rather that it a point of focus that affords a degree of modularity. The ontology needs to be relevant to the processes that use it, for example by having associations to IE services, or relations to domain specific document templates.

Ontologies provide a solid underpinning for the automatic extraction, consolidation and re-representation of knowledge. They encourage consistent modelling, enable more modular and generic systems, and provide a focal point for complex chains of knowledge manipulation processes.

Acknowledegments

This research is funded in part by the EU project "eCHASE" under the contract EDC11262, EPSRC IRC project "Equator" GR/N15986/01 and EPSRC IRC project "AKT" GR/N15764/01.

References

Aitken, J.S., 2002. Learning information extraction rules: an inductive logic programming approach. In: Proceedings of the European Conference on Artificial Intelligence (ECAI), France. pp. 335–359.

Alani, H., Jones, C., Tudhope, D., 2000. Associative and spatial relationships in thesaurus-based retrieval. In: Proceedings of the Fourth European Conference on Digital Libraries, Lisbon, Portugal. pp. 45–58.

Alani, H., Kim, S., Millard, D.E., Weal, M.J., Hall, W., Lewis, P.H., Shadbolt, N.R., 2003. Automatic ontology-based knowledge extraction and tailored biography generation from the web. IEEE Intelligent Systems 18 (1), 14–21.

Aone, C., Ramos-Santacruz, M., 2000. Rees: A large-scale relation and event extraction system. In: Proceedings of the Sixth Applied Natural Language Processing Conference, U.S.A. pp. 76–83.

Bailey, C.P., Hall, W., Millard, D.E., Weal, M.J., 2002. Towards open adaptive hypermedia. In: Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, Springer, Berlin, vol. 2347. pp. 36–46.

Berners-Lee, T., Hendler, J., Lassila, O., 2001. Semantic web. Scientific American. May, 35–43.

Brusilovsky, P., 1996. Methods and techniques of adaptive hypermedia. Journal of User Modelling and User-Adaptive Interaction 6 (2), 87–129.

Brusilovsky, P., 2001. Adaptive hypermedia. User Modeling and User-Adapted Interaction. Ten Year Anniversary Issue 11, 87–110.

Cimiano, P., Handschuh, S., Staab, S., 2004. Towards the self-annotating web. In: Proceedings of the 13th International Conference on World Wide Web (WWW '04). ACM Press, New York, NY, USA, pp. 462–471.

Ciravegna, F., 2001. Adaptive information extraction from text by rule induction and generalisation. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI). pp. 1251–1256.

Ciravegna, F., Dingli, A., Iria, J., Wilks, Y., 2003. Multi-strategy definition of annotation services in MELITA. In: Workshop on Human Language Technology for the Semantic Web and Web Services. Proceedings of the Second International Semantic Web Conference (ISWC), Sanibel Island, FL.

Ciravegna, F., Chapman, S., Dingli, A., Wilks, Y., 2004. Learning to harvest information for the semantic web. In: Proceedings of the First

European Semantic Web Symposium, Heraklion, Greece, May, pp. 312–326.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S., 2000. Learning to construct knowledge bases from the world wide web. Artificial Intelligence 118 (1–2), 69–113.

Cunningham, H., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M., 2002a. Developing language processing components with GATE (user's guide). Techical Report, University of Sheffield, U.K., available in ⟨http://www.gate.ac.uk/⟩.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., 2002b. GATE: a framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, Philadelphia, USA.

Davulcu, H., Koduri, S., Nagarajan, S., 2003. OntoMiner: bootstrapping and populating ontologies from domain specific web sites. In: Proceedings of the First International Workshop on Semantic Web and Databases, Berlin, Germany.

Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y., 2003. SemTag and seeker: bootstrapping the semantic web via automated semantic annotation. In: Proceedings of the World Wide Web (WWW) Conference, Budapest, Hungary, pp. 178–186.

Dingli, A., Ciravegna, F., Guthrie, D., Wilks, Y., 2003. Mining web sites using unsupervised adaptive information extraction. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary, pp. 75–78.

Downey, D., Etzioni, O., Soderland, S., 2005. A probabilistic model of redundancy in information extraction. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 1034–1041.

Dupplaw, D., Dasmahapatra, S., Hu, B., Lewis, P., Shadbolt, N., 2004. Multimedia distributed knowledge management in MIAKT. In: Handshuh, S., Declerck, T. (Eds.), Proceedings of Knowledge Markup and Semantic Annotation, Third International Semantic Web Conference, pp. 81–90.

Fensel, D., 2001. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer, Berlin.

Grishman, R., Sundheim, B., 1996. Message understanding conference—6: a brief history. In: Proceedings of the 16th International Conference on Computational Linguistics, Copenhagen.

Guarino, N., Giaretta, P., 1995. Ontologies and knowledge bases: towards a terminological clarification. Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing. IOS Press.

Guha, R., McCool, R., 2003. TAP: a semantic web platform. Computer Networks: The International Journal of Computer and Telecommunications Networking 42 (5).

Handschuh, S., Staab, S., Ciravegna, F., 2002. S-cream semi-automatic creation of metadata. In: Proceedings of the Semantic Authoring, Annotation and Markup Workshop, 15th European Conference on Artificial Intelligence (ECAI'02), France, Lyon, pp. 27–33.

Handschuh, S., Staab, S., Volz, R., 2003. On deep annotation. In: Proceedings of the 12th International World Wide Web Conference (WWW'03), Budapest, Hungary, ACM Press, New York, pp. 431–438.

Harpring, P., 1997. Proper words in proper places: the thesaurus of geographic names. MDA Information 2 (3), 5–12.

Hill, L.L., Frew, J., Zheng, Q., 1999. Geographic names: the implementation of a gazetteer in a georeferenced digital library. Digital Library Magazine 5 (1).

Kahan, J., Koivunen, M.-R., 2001. Annotea: an open RDF infrastructure for shared web annotations. In: Proceedings of the 10th International Conference on World Wide Web, Hong Kong, pp. 623–632,

Katz, B., 1997. From sentence processing to information access on the world wide web. In: Proceedings of AAAI Spring Symposium on Natural Language Processing for the World Wide Web, U.S.A., pp. 77–86.

Kim, S., Alani, H., Hall, W., Lewis, P.H., Millard, D.E., Shadbolt, N.R., Weal, M.J., 2002. Artequakt: generating tailored biographies with automatically annotated fragments from the web. In: Proceedings of the Workshop on Semantic Authoring, Annotation and Knowledge Markup in Conjunction with the 15th European Conference on Artificial Intelligence (ECAI), pp. 1–6.

Lee, K., Luparello, D., Roudaire, J., 1999. Automatic construction of personalised TV news programs. In: Proceedings of the Seventh ACM Conference on Multimedia, Orlando, FL, pp. 323–332.

Litkowski, K.C., 1999. Question-answering using semantic relation triples. In: Proceedings of Text REtrieval Conference, U.S.A., pp. 349–356.

Luke, S., Spector, L., Rager, D., 1996. Ontology-based knowledge discovery on the world-wide web. In: Franz, A., Kitano, H. (Eds.), Working Notes of the Workshop on Internet-Based Information Systems at the 13th National Conference on Artificial Intelligence (AAAI'96). AAAI Press, pp. 96–102.

Luke, S., Spector, L., Rager, D., Hendler, J., 1997. Ontology-based web agents. In: Johnson, W.L. (Ed.), Proceedings of the First International Conference on Autonomous Agents (Agents97). Association for Computing Machinery, New York, pp. 59–66.

Maedche, A., Neumann, G., Staab, S., 2002. Bootstrapping an ontology-based information extraction system. Intelligent Exploration of the Web. Springer/Physica Verlag, Berlin.

Mancini, C., 2000. From cinematographic to hypertext narrative. In: Proceedings of the 11th ACM Conference on Hypertext and Hypermedia, San Antonio, TX, USA., pp. 236–237.

McKeown, K., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J., Nenkova, A., Sable, C., Schiffman, B., Sigelman, S., 2002. Tracking and summarizing news on a daily basis with Columbia's Newsblaster. In: Proceedings of the Human Language Technology Conference, San Diego, CA, USA.

Michaelides, D., Millard, D., Weal, M., DeRoure, D., 2001. Auld leaky: a contextual open hypermedia link server. In: Hypermedia: Openness, Structural Awareness, and Adaptivity (Proceedings of OHS-7, SC-3 and AH-3), Lecture Notes in Computer Science, vol. 2266, Springer, Heidelberg, pp. ,59–70.

Millard, D., Moreau, L., Davis, H., Reich, S., 2000. FOHM: a fundamental open hypertext model for investigating interoperability between hypertext domains. In: Proceedings of the 11th ACM Conference on Hypertext and Hypermedia, San Antonio, Texas, USA. pp. 93–102.

Millard, D., Alani, H., Kim, S., Weal, M., Lewis, P., Hall, W., De Roure, D. C., Shadbolt, N., 2003. Generating adaptive hypertext content from the semantic web. In: Proceedings of the First International Workshop on Hypermedia and the Semantic Web, HyperText'03, Nottingham, UK.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K., 1993. Introduction to wordnet: An on-line lexical database. Technical Report, University of Princeton, U.S.A.

Milosavljevic, M., 1997. Augmenting the user's knowledge via comparison. In: Proceedings of the Sixth International Conference on User Modeling, UM97. Springer, Wien, pp. 119–130.

Musen, M.A., Fergerson, R.W., Grosso, W.E., Noy, N.F., Grubezy, M.Y., Gennari, J.H., 2000. Component-based support for building knowledge-acquisition systems. In: Proceedings of the Intelligent Information Processing (IIP 2000) Conference of the International Federation for Processing (IFIP), World Computer Congress (WCC'2000), Beijing, China.

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., Goranov, M., 2003. Towards semantic web information extraction. In: Workshop on Human Language technology for the Semantic Web and Web Services. Proceedings of the Second International Semantic Web Conference (ISWC2003), Sanibel Island, Florida. Springer, Berlin, pp. 484–499.

Radev, D.R., McKeown, K.R., 1998. Generating natural language summaries from multiple on-line sources. Computational Linguistics 24 (3), 469–500.

Reidsma, D., Kuper, J., Declerck, T., Saggion, H., Cunningham, H., 2003. Cross document annotation for multimedia retrieval. In: Proceedings of the EACL Workshop on Language Technology and the Semantic Web (NLPXML), Budapest, Hungary.

Roth, D., Yih, W.T., 2002. Probabilistic reasoning for entity and relation recognition. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING 02),Taiwan.

Rutledge, L., Bailey, B., Ossenbruggen, J.V., Hardman, L., Geurts, J., 2000. Generating presentation constraints from rhetorical structure. In: Proceedings of the 11th ACM Conference on Hypertext and Hypermedia, San Antonio, TX, USA, pp. 19–28.

Rutledge, L., Alberink, M., Brussee, R., Pokraev, S., van Dieten, W., Veenstra, M., 2003. Finding the story—broader applicability of semantics and discourse for hypermedia generation. In: Proceedings of the 14th ACM Conference on Hypertext and Hypermedia, Nottingham, UK, August, pp. 67–76.

Sekine, S., Grishman, R., 1995. A corpus-based probabilistic grammar with only two non-terminals. In: Proceedings of the Fourth International Workshop on Parsing Technology, pp. 216–223.

Snoussi, H., Magnin, L., Nie, J.-Y., 2002. Toward an ontology-based Web data extraction. In: Proceedings of the 15th Canadian Conference on Artificial Intelligence, University of Calgary, Calgary, Altas, Canada.

Staab, S., Maedche, A., Handschuh, S., 2001. An annotation framework for the semantic web. In: Proceedings of the First International Workshop on MultiMedia Annotation, Japan.

Vargas-Vera, M., Motta, E., Domingue, J., Buckingham Shum, S., Lanzoni, M., 2001. Knowledge extraction by using an ontology-based annotation tool. In: Proceedings of the Workshop on Knowledge Markup and Semantic Annotation, First International Conference on Knowledge Capture (K-CAP'01), Victoria, B.C., Canada, pp. 5–12.

Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., Ciravegna, F., 2002. MnM: ontology driven semi-automatic and automatic support for semantic markup. In: Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW), Spain, pp. 379–391.

Voorhees, E., 1998. Using wordnet for text retrieval. In: Fellbaum, C. (Ed.), WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA, pp. 285–303.

White, M., Korelsky, T., Cardie, C., Ng, V., Pierce, D., Wagstaff, K., 2001. Multidocument summarization via information extraction. In: Proceedings of Human Language Technology Conference (HLT 2001), San Diego, CA.