Eight questions about semantic web annotations

Jérôme Euzenat

Eight Questions about Semantic Web Annotations

Jérôme Euzenat, *INRIA Rhône-Alpes*

Improving information retrieval is among annotation's central goals. However, without clear guidelines, annotation risks producing incoherent information. The author recommends answering eight key questions before annotation begins.

One prominent purpose of the Semantic Web is to improve information retrieval. To do this, we annotate informal Web pages on the basis of a formal description of their content. However, annotation isn't always productive; if it hasn't been designed in close relation to its use, it will produce limited benefits. To increase annotation's success, I single out eight questions to answer before the work begins.

These eight questions are not restricted to a particular context. In particular, they are independent of any document medium (documents can be texts, images, or multimedia documents) or representation language (the only concept appearing in the questions is that of genericity, which I discuss later). Issues raised here are relevant to any Semantic Web annotation by content.

This article introduces each of the eight questions and describes how they are applied as the foundation for developing a system to help biologists find articles relevant to their research. Using the content of abstracts, a domain ontology and annotations have been produced manually. This has been processed, with the idea that content annotations—but not an ontology—might be later extracted automatically from the abstracts.[1,2]

Annotating genetics abstracts

Escrire is a joint project between three INRIA teams comparing knowledge representation language families for content representation on the Web (see Figure 1). The compared families are conceptual graphs, object-based knowledge representation languages, and description logics. The comparison protocol involves building a bridge between a common pivot language (in which ontologies and annotations are encoded) and a particular representation language. Escrire takes queries in the pivot language, issues them in the three language families, and compares the results. Other projects also compare knowledge representation languages,[3] but Escrire compares them experimentally in context. (For more information on Escrire, see http://escrire.inrialpes.fr.)

The case study examined here concerns the domain of genetic interactions in the fruit fly (*Drosophila melanogaster*) embryo's early development (for more information, see the sidebar "Genes and Fruit Flies.") We chose this topic because we had already built a knowledge base on the subject.[4]

The annotated pages that Escrire processed in this experiment were abstracts of genetic articles from the National Institute of Health's Medline public database. Figure 2 shows an example abstract; I use examples related to it throughout this article.

I identify fragments of Medline abstracts by their PubMed Identifier in brackets (for example, [ID:1972684] for Figure 2). These abstracts are all available through PubMed (www.ncbi.nlm.nih.gov/PubMed), a site that provides public access to Medline. Biological journal abstracts have advantages from the viewpoint of our experiment. They are, like the biological articles themselves, relatively precise and tend to describe the article's conclusions (unlike abstracts for computer science articles, for instance).

An annotation system can answer queries such as "Does giant regulate some homeotic gene?" The query evaluation should be able to take into account the synonyms of "giant" (for example, `gt`) and that the abstract can evoke a particular homeotic gene without stating that it is homeotic (or mention inhibition and not regulation). This should be more efficient than a full-text search because content is normalized (see Figure 1 for two other interpretations

1094-7167/02/$10.00 © 2002 IEEE

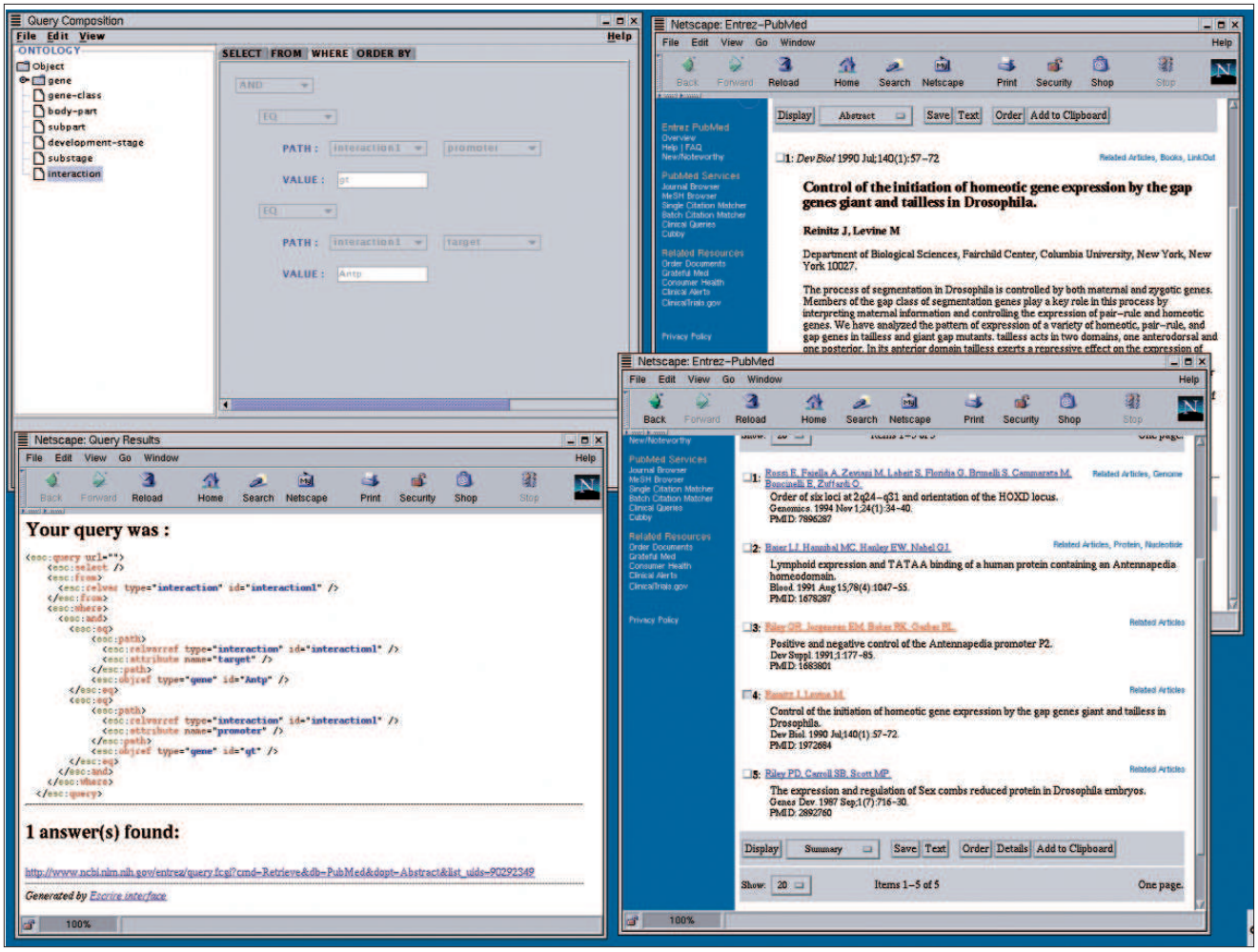

Figure 1: The Escrire system. The top-left window is a query application that displays a class hierarchy and a query to be composed. The bottom-left window is a browser displaying an XML representation of the query and its answers. The lower-right window features the answers provided by Medline (out of the five answers, only two, including the one that the formal system returned, displayed in the top-right window, are valid).

of "giant") and because the query can take advantage of taxonomical knowledge (such as that "Antennapedia" is a homeotic gene). It should also be better than hierarchical keyword search because it articulates terms in a relational language (here, the regulation of homeotic genes by "giant").

Terminology

Defining the terminology has two purposes: avoiding misunderstanding and, more notably, formulating the problems the eight questions raise.

An entity is *generic* if it generally applies to several individuals, and *individual* if it only concerns a particular individual within the domain of interpretation. A class, relation, or rule is a generic entity (for example, "gene"); an object or assertion about an object is an individual entity (for example, "Antennapedia").

A *schema* specifies the generic entities used for expressing content. It is opposed to a *description* (a set of individual entities).

Genes and Fruit Flies

An organism's genome is the set of genes found in all its cells. Genes are materialized by nucleic acid sequences, which the cellular machinery uses to produce proteins. It does this by transforming the DNA into RNA and interpreting the code that the sequences carry into proteins. The proteins in turn constitute new cellular machinery.

A protein can interact with other proteins and even nucleic acids (mainly by neutralizing one expression process or its product). An interaction results in the target gene's inhibition or activation. These interactions are important because they have a regulatory role. This role is evident in the embryo's early development: the cells, which come from the replication of a unique cell (and thus share the same genome), differentiate themselves as eye cells, nerve cells, bone cells, and so on.

Drosophilists (those who study *Drosophila melanogaster*, the fruit fly) have studied these interactions through many different approaches, including observing a missing gene's effects on the grown-up drosophila. They have identified sets of genes, called *gene classes*, that take part in the anterio-posterior axis determination; in its subsequent differentiation into head, thorax, and abdomen; and in the differentiation of the abdomen into segments (which will determine the number of leg pairs). Some proteins come from the mother and constitute the egg's environment (the genes expressing these proteins are in the *maternal* class). The other genes, those which are not maternal, are *zygotic*. Because, in a class, genes tend to control each other, determining exactly which gene interacts with another is difficult. Instead, biologists consider that gene classes interact one with another.

Control of the initiation of homeotic gene expression by the gap genes giant and tailless in Drosophila.
Reinitz J, Levine M
Department of Biological Sciences, Fairchild Center, Columbia University, New York, New York 10027.

The process of segmentation in Drosophila is controlled by both maternal and zygotic genes. Members of the gap class of segmentation genes play a key role in this process by interpreting maternal information and controlling the expression of pair-rule and homeotic genes. We have analyzed the pattern of expression of a variety of homeotic, pair-rule, and gap genes in tailless and giant gap mutants. tailless acts in two domains, one anterodorsal and one posterior. In its anterior domain tailless exerts a repressive effect on the expression of fushi tarazu, hunchback, and Deformed. In its posterior domain of action, tailless is responsible for the establishment of Abdominal-B expression and demarcating the posterior boundary of the initial domain of expression of Ultrabithorax. giant is an early zygotic regulator of the gap gene hunchback: in giant-embryos, alterations in the anterior domain of hunchback expression are visible by the beginning of cycle 14. giant also regulates the establishment of the expression patterns of Antennapedia and Abdominal-B. In particular, giant is the factor that controls the anterior limit of early Antennapedia expression.
PMID: 1972684, UI: 90292349

Figure 2. An example abstract of a genetics article from the National Institute of Health's Medline public database.

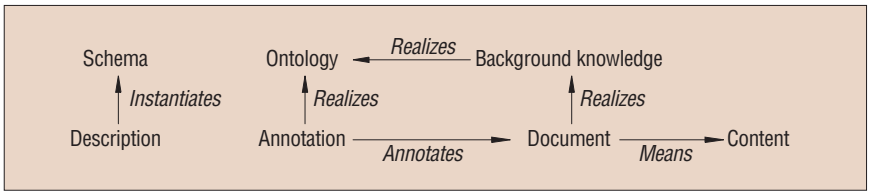

Figure 3. The various terms and their relationships, which form two disconnected graphs. The connections within each graph might or might not be completed by the answers to the eight questions in the main article. Here realizes means both instantiates and specializes.

Schemas and descriptions are only syntactic notions; they do not ascribe a particular role to what they denote (schematic entities could be in ontologies or annotations).

Background knowledge is a set of assertions that can be schematic or descriptive and that must be assumed when trying to assess a document's content. It corresponds to knowledge the author of a document thinks he or she shares with the readership. For the abstract in Figure 2, readers are supposed to know that "Antennapedia" is a homeotic gene; this isn't explicitly written but necessary for proper understanding. Of course, the background knowledge can be void.

An *ontology* is a set of assertions specifying the concepts involved in the domain. Background knowledge is common knowledge used for understanding the content. I distinguish the ontology, which is partly or wholly generic knowledge, from the background knowledge, which can only contain individual descriptions.

Document denotes the article abstracts extracted from Medline. More precisely, it corresponds to the Medline entries' abstract (without titles, authors, and journal names). The *content*, generally speaking, is the document's meaning. In the present context, *content* denotes the meaning that we want to represent. This is not the document's full meaning but a clearly circumscribed part of it. An *annotation* is the content represented in a formal language and attached to the document.

On documents and content representations, we can execute these operations:

- *Data extraction* of annotations from document content,
- *Generation* of a documentary representation of formal representations,
- *Indexing*, from a set of documents and a set of formal representations, which generates a function from the latter to the former enabling the retrieval of documents from representations,
- *Annotation*, from a set of documents and a set of formal representations, which generates a function from the former to the latter, enabling the interpretation of document content.

In this article, the difference between *indexing* and *annotation* is not relevant. The goal is to consider the relation between *document* and *annotation*. We do not use *annotation* for its operational meaning but for the object it denotes (that is, the formal representation of content).

We expect, from such an annotation scheme, that the combination of ontology (O), background knowledge (K), and annotation (A) enables the reconstruction of content (γ). In other words, if $\models$ denotes the logical consequence, $O \cup K \cup A \models \gamma$.

Figure 3 summarizes the relationships between these terms.

The eight questions

The eight questions relate to the use of annotations and the content's epistemological status. Some are common to natural language understanding, knowledge representation, and information retrieval research. However, the questions take on a new dimension in the context of the Semantic Web because they interact with one another. The questions fall into three categories depending on the entities they affect: annotation, background knowledge, and the Web.

Annotation

At the beginning are the actual (or future) documents carrying the content. Assigning annotations to this informal knowledge requires deciding about the kind of information to represent and the representation's form. This is closely related to the kind of queries to be answered. The annotations must represent the part of the content that the user wants to query, which therefore relates to the specific application. However, the content itself calls for some representational features that we cannot ignore.

Question 1: What aspect of the content must be represented?

Annotations can include various status

information—for example,

- *Media data* (date of creation, length, encoding, or format)
- *Metadata* (data about data—for instance, authors or production date)
- *Indexes* (document object identifier or MedLine unique identifier)
- *Content descriptors* from a predefined set (keywords and phrases, or categories)
- *Content representation* (preview or abstract)

Medline already provides metadata about the articles, including languages, ISSN, publication types, and annotation by medical subject headings (a hierarchical organization of medical terms). Annotations are not always content representations. For example, if a biogenetics article does not specify that it deals with biogenetics, this must be added as metadata.

Escrire deals with the representation of content in a formal language. This representation can be considered complex, compared to a list of terms, because it is expressed in a recursive language. So, you can use it to create representations instead of just using predefined ones. It does not deal with metadata but deals strictly with the content of the abstracts (which themselves can be considered informal representations of article content).

Even in complex content representations, several types of structures can be described:

- The text's *grammatical structure* decomposes sentences into units such as noun phrases (`np`) or verbs (`v`). Figure 4 shows the grammatical tagging for "The process of segmentation in Drosophila is controlled by both maternal and zygotic genes."
- The *rhetorical structure* is the text's argumentative structure. In general, the used abstracts can be represented as "'state of the art' and 'experiment' entails 'conclusion.'" For instance, the abstract in Figure 2 can be represented as "Assertion: sentence #1–2; Experiment description: sentence #3; Results: sentence #4–9." The representation can take deeper rhetoric into account (for example "in particular" in the abstract's last sentence links sentences 8 and 9 by a generality relation).
- The *logical structure* deals with the content's conceptual and relational representation.

Each of these structures is useful for a particular purpose and enables the characterization (and then the retrieval) of particular features and not others. Escrire uses the logical structure.

```
<np opr="sbj">
    <ad sem="df.sg">The</ad>
    <n sem="sg">process</n>
    <adp><ad opr="arg">of</ad>
        <nph><n sem="sg" root="segment">segmentation</n></nph>
    </adp>
    <adp><ad opr="loc">in</ad><np><name>Drosophila</name></np></adp>
</np>
<vph fun="gov">
    <v root="be" fun="aux" sem="sg">is</v>
    <v root="controll" fun="gov" sem="past">controlled</v>
</vph>
<adp><ad opr="mns">by</ad>
    <np fun="comp">
        <ajp syn="p"><io>both</io>
            <aj>maternal</aj>
            <io>and</io>
            <aj>zygotic</aj>
        </ajp>
        <n root="gene" sem="pl">genes</n>
    </np>
</adp>
```

Figure 4. The grammatical tagging for the sentence "The process of segmentation in Drosophila is controlled by both maternal and zygotic genes."

```
<objref name="Antennapedia" type="gene"/>
(a)

<objref name="Antennapedia" type="homeotic"/>
(c)

<relation type="interaction">
    <role name="promoter">
        <objref type="gap" id="gt"/>
    </role>
    <role name="target">
        <objref type="homeotic" id="Antp"/>
    </role>
</relation>
(d)
```

```
<objref name="homeotic"/>
(b)

<relation type="interaction">
    <role name="promoter">
        <objref type="gap" id="gt"/>
    </role>
    <role name="target">
        <objref type="homeotic" id="Antp"/>
    </role>
    <attribute name=
        "effect">inhibition</attribute>
    <attribute name=
        "location">anterior</attribute>
</relation>
(e)
```

Figure 5. Representation of different depths of content: (a) a reference to the *Drosophila* genes mentioned in a document (some research stops here[7]); (b) a reference to the *Drosophila* gene classes in the document; (c) a reference to the *Drosophila* genes and an assertion of their class; (d) an assertion of interactions between the genes; (e) an assertion of interactions and their circumstances (location and effect).

Question 2: What are the subject and form of the knowledge to represent?

In the current state of the art, formally representing the complete content of even simple text does not seem reasonable or straightforward. A directly extracted predicate calculus representation of the abstract in Figure 2 is four pages long. So, the Escrire team restricted the experiment's scope to the expression of simple statements (à la SHOE[5] or OntoBroker[6]). However, this allows representation of several depths of content (see Figure 5).

Because the information displayed in Figure 5e corresponds to what biologists expected, Escrire does not go deeper. You could go further, for instance, by adding information about the experimental context or the phenotypic consequences of inter-

action between the genes (that is, the physical consequence on the grown-up drosophila).

The Escrire experiment thus restricted the representations to that of genes, gene classes, and gene interactions in the knowledge base. This reduces the abstract's content to the point that some abstracts describing articles related to interactions have no formal content. However, the abstract in Figure 2 contains references to seven genes, three gene classes, and nine interactions. Its actual annotation is two XML pages long.

Question 3: Are annotations only descriptions?

So far, annotating seems simple. It relies on identifying specific elements in the document and including the corresponding description as annotation. The common understanding of annotations is that they constitute individual descriptions or data.

A quick natural language syntactic analysis would reveal that "Antennapedia" is a proper noun and might denote an individual. However, "Antennapedia" is not an individual gene but the concept of a particular gene of drosophila (which should be present in all cells of all drosophila). Representing it as an individual does not raise problems as long as the biologist does not describe an individual gene. In fact, the journal abstracts scanned from Medline are remarkably homogeneous in their syntax and content: they concern the same set of objects, use the same set of tools, and make the same kind of claims.

Conversely, "homeotic genes" is a noun phrase denoting a set of individuals, and "gap" is a proper noun denoting a class. (Here, the common noun "gap" stands for the proper noun of a class of genes; suppressing one introduces a gap in the segments of the drosophila's body.) Mentioning classes is convenient for asserting the described object's type, such as "homeotic":

```
<objref name="Antennapedia" type="homeotic"/>
```

However, some classes stand for themselves. For example, the abstract sentence "Polycomp (Pc), acts as a repressor of the ANT-C and BX-C" [ID:2563569] treats the Antennapedia and Bithorax complexes (sets of genes) like individual genes. Representing this assertion requires the ability to use classes as objects.

Moreover, applications exist in which only generic features of the represented objects are important. For example, if you annotate a set of documents describing a vita, the person's identification will never be queried directly and the relevant features are class-like. For instance, the described person is a programmer who has a certain number of years of experience and has mastered particular programming languages.

So, it isn't true that the ontology will provide the type structure and that the content only refers to individuals. The content can be about the classes, and it can assert very strong properties of a class (for instance, the class can only contain one element or can be a subclass of another class).

The application designer decides whether these classes (or defined set of individuals) must be expressible. In Escrire, the classes are expressible in annotations because the gene classes were found everywhere in the abstracts. In the following sections, "Antennapedia" is still an individual and "homeotic" is its class.

Question 4: Must we reify some classes in descriptions?

The classes can be represented in two ways: as classes (higher-level constructors) or individuals (*reified*). This leads to the following syntactical difference:

```
<classref name="homeotic"/>
```
or
```
<objref name="homeotic" type="class"/>
```

It seems natural to express directly the classes in annotations, but this requires a more elaborate manipulation and query language. Moreover, the knowledge's schema is then dynamically modified (preventing knowledge compilation).

Reification is less natural but does not require introducing generic structures in annotations. However, if these generic structures also exist in the ontology ("homeotic" is represented both as the class of the "Antennapedia" object in the ontology and as its reification as a "homeotic" object in the annotations), the coherence between both representations must be maintained. Moreover, if you choose to reify the classes, you must provide the interpretation of the represented statements (does the control concern all the individual instances of the maternal gene class or only one?) and the adequate constructions (the proper quantifiers).

Some languages, such as RDF Schema,[8] allow considering sets, sets of sets, and so on through a reification mechanism. Some others deliberately separate objects from classes, such as the Abox/Tbox of early description logics. These approaches have been debated for years and are still discussed to determine whether an annotation language must deal with reification. The only certainty is that you must pick an option and consistently apply it.[9]

Modeling becomes difficult because we can choose between instances and classes, classes and subclasses, or classes and classes of classes for the "Antennapedia–homeotic" couple. Escrire reifies the notion of gene class, maintained as classes of genes and as named set of genes that can be manipulated in annotations and queries.

Background knowledge

Interpreting documents requires prior knowledge. This background knowledge makes explicit what is implicit in the document body. Ontologies can be considered as the background knowledge. As I mentioned in the "Terminology" section, it is necessary to clarify the respective roles of ontologies and background knowledge in document annotations.

Question 5: Is some background knowledge necessary?

In an article concerning drosophila, no biologist will consider anything about plants or mammals. Moreover, biologists will focus on those genes that are most often studied in the species (namely, the antero-posterior axis development factors). Identifying the knowledge required to read the abstract in Figure 2 is difficult. The biology professor, the undergraduate biology student, and the computer scientist do not learn the same things from the abstract. The biology professor learns that "giant controls the expression of Antennapedia in the anterior part of the egg." The undergraduate learns that "some zygotic genes influence the segmentation of drosophila." The computer scientist learns that "interaction exists between genes in drosophila." The obtained knowledge depends heavily on the available background knowledge.

For instance, the abstract in Figure 2 draws conclusions about the gap's influence on homeotic classes, although the experiments show evidence of interaction between individual genes without mentioning their classes. Understanding the content requires knowledge about instances (for example, that "Antennapedia" is a homeotic gene).

The lack of common knowledge about

individuals prohibits queries involving proper nouns such as "Antennapedia." It also prohibits answering the question "Is X inhibited by Z?" across articles. (When one article tells that "X enhances Y" and the other that "Y inhibits Z," no answer is possible if the "Y" is not shared.)

Finding and characterizing background knowledge is thus an important issue for the Semantic Web.

Question 6: Is the background knowledge part of the ontology?

One issue is the distinction between ontology and background knowledge. If generic entities can be part of annotations, we cannot consider ontologies to be the type system of individuals found in content. Because ontologies are already common knowledge, they can be the content's background: the necessary knowledge for understanding the content. Ontologies can provide more context to knowledge, such as in document or query expansion.[10]

If all background knowledge is part of the ontology, individuals such as "Antennapedia" must be part of the ontology. However, some applications demand different background knowledge. For instance, a system identifying gene expression pathways for expert biologists cannot have the same background knowledge as a system for teaching high school biology. So, it is convenient to be able to change background knowledge for interpreting documents, while ontologies, because they are shared, can be considered more stable.

On the contrary, if the ontology and background knowledge are separate, and given that individuals are found in background knowledge, adding constraints involving instances to the ontology is impossible.

In summary, ontologies can play two different roles: schema providing types for annotations and context providing background knowledge. In some languages, such as object-based languages, the constructors help specify the difference between schema and individuals. Others, such as logic languages, do not separate them. In any case, separating background knowledge from the ontology enables better separation of these roles.

Question 7: Can the background knowledge and ontology evolve?

In some abstracts, the authors introduce new concepts or at least new classes of a thing—for example, "a new set of genes described here, which we call tube expansion genes" [ID:10887083]. If these new concepts become consensual, they must be added to the ontology. This requires that ontologies evolve.

However, this issue can lead to a complete relativism stating that no ontology exists; rather, concentric circles of more or less accepted background knowledge exists. Such a vision would reconcile the idea that a query's context can be assembled from several compatible domain ontologies, that some knowledge introduced at some point becomes part of another background knowledge circle, and that the "ontology" can grow out of the Web. This organization is quite close to the microtheory ideas involved in CyC.[11]

More radically, this could lead to the consequence that the Semantic Web has no background knowledge (or that it is its own context): it must gather all its knowledge from the Web. Moreover, if classes are defined in the content, no ontology would be needed—the Semantic Web must spin one!

Concerning these questions, Escrire merges the ontology and the background knowledge under the name of ontology. This ontology contains the descriptions of the common objects, and it cannot evolve.

The Web

The global level is the Semantic Web itself. It is made of distributed knowledge resources (documents, annotations, ontologies, and background knowledge). An application must handle these resources with a clear idea of the gathered knowledge's status. So, we come to Question 8.

Question 8: Is the Semantic Web common knowledge or distributed knowledge?

One problem during the modeling phase is determining the correct query evaluation strategy. A basic approach tries to answer a query (q) with the background knowledge and only the annotations of each document. More precisely, it will return references to the documents that, joined with the background knowledge, provide an answer to the query: $K \cup O \cup A \models q$. But, by doing so, the system does not take advantage of the information the rest of the Web can provide. More specifically, if you ask for interactions between the genes "giant" and "spalt major," the system will not return an answer because no abstract mentions these genes together.

So, a second approach consists of finding these minimal sets of documents whose annotations ($A_1, \ldots, A_n$), together with the background knowledge, provide an answer ($K \cup O \cup A_1 \cup \ldots A_n \models q$). For example, a biologist might like to know that a document exists evoking interactions between "giant" and "Antennapedia" and another evoking interactions between "Antennapedia" and "spalt major."

This change of approaches changes query interpretation. With the first approach, the query is "Which documents mention the regulation of spalt major by giant?" With the second, the query is "Does (this Semantic Web tell that) giant regulate spalt major?" The first approach yields a set of documents; the second produces an answer to the query.

This raises the consistency problem: a powerful language will contain sets of inconsistent document annotations entailing any assertion. If these sets are small enough, they will contain candidate answers for any query. The first approach has the merit of confining inconsistencies.

Beside this technical problem, an epistemological problem exists: annotating a set of documents for information retrieval (the first approach) and gathering knowledge for building a giant knowledge base (the second one) are not the same thing. The first approach considers the documents independently, in the same way they were designed. They don't have to be consistent with other documents and can be assumed self-sufficient. The second approach puts the documents in the context of many other documents. Care must be taken to ensure coherence of the annotations. These other documents could have been unknown by the author and could change, restrict, or contradict the content.

Furthermore, the first approach doesn't let you replace one term with another equivalent term defined elsewhere. The second approach lets you freely use equivalent terms. These problems already occur in the Web, and humans are accustomed to solving them. Machines are not, so we must help them. This is the main reason for defining the expected application before beginning an annotation project.

Applying the questions to existing systems

The answers to these eight questions should provide a framework enabling to compare Semantic Web applications. Table 1 summarizes the answers for six systems.

Table 1: How different systems deal with the eight questions.

Question	Medline	MUC	Escrire	Escrire NG	SHOE	KA2
What aspect of the content must be represented?	Organisms, subjects, journals, and authors	Products, people, and organizations,	Genes, gene classes, and interactions		Researchers and laboratories	Researchers, laboratories, and publications
What are the subject and form of the knowledge to represent?	Objects	Objects, relations, and actions	Objects and relations		Objects and relations	Objects and relations
Are annotations only descriptions?	Y	Y	Y	N	Y	Y
Must we reify some classes in descriptions?	N	N	Y	Y	N	N
Is some background knowledge necessary?	Y (list of journals…)	Y (hidden)	Y	Y	Y	N
Is the background knowledge part of the ontology?	N	—	Y	N	Y	N
Can the background knowledge and ontology evolve?	N	N	N	N	N	N/Y
Is the Semantic Web common knowledge or distributed knowledge?	Distributed	Distributed	Distributed		Common	Common

Medline, the first system, annotates the same articles as Escrire with metadata and attaches them under hierarchically organized terms. Unlike other systems, Medline serves as a witness—that is, a system substantially different from the others.

The second system is the evaluation scheme for the *MUC* (message understanding conferences)—specifically, the one prevalent until MUC-5**.** It assesses the capacity of information extraction systems; I view this assessment as an annotation process because the competing systems must fill a template similar to an object, event, or structure. MUC experiments take advantage of a restricted ontology that is a generalization of the templates. The scheme also takes advantage of background knowledge—for instance, country names. However, the ontology and background knowledge remain implicit.

I've already described Escrire in this article; *Escrire NG*, Escrire's new generation, could also take advantage of the remarks made in this paper: separating background knowledge from ontology and allowing generic knowledge in annotations.

SHOE (Simple HTML Ontology Extensions) is an annotation language for guiding agents on the Web.[5] Its typical application enables Web page markup by a frame-like language. I consider it here in its pilot application to computer-science-researcher-and-laboratory representation. SHOE separates ontologies and annotations, but refining the ontologies is possible and common knowledge can be part of ontologies.

KA2 is the application of the Ontobroker system for building a knowledge base about knowledge acquisition from annotated Web pages.[6] Like SHOE, it gathers distributed knowledge and provides answers to queries. Unlike SHOE, the ontology is part of the application, and Ontobroker does not allow background knowledge in the ontologies: the only knowledge sources are the annotations.

Current systems seem to offer only a common minimal platform. These systems are relatively similar, especially regarding the knowledge representation format. The most important distinctions concern query interpretation. As Table 1 shows, these systems are not internally prepared for the evolution of ontologies and background knowledge.

The eight questions presented here do not delimit the entire annotation by content design space. They are a first contribution to the methodology of annotating with formal content. Practical worldwide experiments should help refine some of these and raise other questions. In the context of the Semantic Web, there are two important issues: the context dependence of the annotations, and the evolution of annotations and ontologies.

The approach I've described here raises a question about the Semantic Web's unity: if so many significant combinations of answers to these questions exist, can there be a unique Semantic Web? The systems described in this article demonstrate that there can be several slightly different Semantic Webs.

One challenge, once the options each system takes are made explicit, consists of exploiting the relationships between these systems and to take advantage in one particular context of the knowledge from another context—consistently. ■

Acknowledgments

The INRIA Escrire concerted research action, supported by France Telecom, partially funded this work. I thank the Escrire team members (Rim Al-Hulou, Olivier Corby, Rose Dieng, Carolina Medina, Emmanuel Nauer, Amedeo Napoli, Yannick Toussaint, and Raphaël Troncy), Gwendal Auffret, Jeff Heflin, Yannick Prié, and Heiner Stuckenschmidt for many fruitful discussions and comments.

References

1. D. Proux, F. Rechenmann, and L. Julliard, "A Pragmatic Information Extraction Strategy for Gathering Data on Genetic Interactions,"

Proc. 8th Int'l Conf. Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, Calif., 2000, pp. 279–285.

2. M. Erdmann et al., "From Manual to Semi-automatic Semantic Annotation: About Ontology-Based Text Annotation Tools," *Proc. Coling Workshop Semantic Annotation and Intelligent Content*, 2000, pp. 79–86.

3. O. Corcho and A. Gómez Pérez, "A Roadmap to Ontology Specification Languages," *Knowledge Engineering and Knowledge Management: Methods, Models, and Tools*, Lecture Notes in Computer Science, no. 1937, Springer-Verlag, Heidelberg, 2000, pp. 80–96.

4. J. Euzenat, C. Chemla, and B. Jacq, "A Knowledge Base for *D. melanogaster* Gene Interactions Involved in Pattern Formation," *Proc. 5th Int'l Conf. Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, Calif., 1997, pp. 108–119.

5. J. Heflin, J. Hendler, and S. Luke, *SHOE: A Knowledge Representation Language for Internet Applications*, tech. report CS-TR-4078, Dept. of Computer Science, Univ. of Maryland at College Park, 1999.

6. D. Fensel et al., "Ontobroker: The Very High Idea," *Proc. 11th FLAIRS Conf.*, 1998, pp. 131–135.

7. Y. Tateisi et al., "Building an Annotated Corpus in the Molecular Biology Domain," *Proc. Coling Workshop Semantic Annotation and Intelligent Content*, 2000, pp. 28–34.

8. D. Brickley and R. Guha, "Resource Description Framework (RDF) Schema Specification 1.0," W3C candidate recommendation, http://www.w3.org/TR/2000/CR-rdf-schema-20000327 (current Nov. 2001).

9. W. Woods, "Understanding Subsumption and Taxonomy: A Framework for Progress," *Principles of Semantic Networks: Exploration in the Representation of Knowledge*, J. Sowa, ed., Morgan Kaufmann, San Francisco, 1991, pp. 45–94.

10. D. McGuinness, "Ontological Issues for Knowledge-Enhanced Search," *Formal Ontology in Information Systems,* N. Guarino, ed., ISO Press, Amsterdam, Netherlands, 1998, pp. 302–316.

11. D. Lenat and R. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project,* Addison-Wesley, Reading, Mass., 1989.

The Author

Jérôme Euzenat is a research officer at INRIA Rhône-Alpes. He leads the Exmo action investigating the exchange of formal knowledge mediated by computers. He is particularly interested in the relationships between representations including abstraction, granularity, versioning and transforming representations. He holds a PhD and habilitation in computer science, both from Grenoble 1 University. Contact him at INRIA Rhône-Alpes, 655 avenue de l'Europe, Montbonnot Saint-Martin, 38334 Saint-Ismier, France; jerome.euzenat@inrialpes.fr.