XML Topic Maps and Semantic Web Mining

Benedicte Le Grand, Michel Soto

Laboratoire d'Informatique de Paris 6
8, rue du Capitaine Scott 75015 Paris, France
Benedicte.Le-Grand@lip6.fr, Michel.Soto@lip6.fr

Abstract. Navigation and information retrieval on the Web are not easy tasks; the challenge is to extract information from the large amount of data available. Most of this data is unstructured, which makes the application of existing data mining techniques to the Web very difficult. However, new semantic structures which improve the results of Web Mining are currently being developed in the Web. This paper presents how one of these semantic structures - XML topic maps – can be exploited to help users find relevant information in the Web. This paper is organised as follows: first, we introduce XML topic maps in the context of Tim Berners-Lee's Semantic Web vision. Then, we show how topic maps allow to characterise and "clean" Web data through the definition of a profile; this is achieved by the analysis of a lattice generated by a classification algorithm - called Galois algorithm. This profile may be used to evaluate the relevance of a web site with regard to a specific request on a traditional search engine. We finally explain how data on the Web can be clustered, organised and visualised in different ways so as to enhance users' navigation and understanding of these documents.

1 Introduction

Navigation and information retrieval on the Web are not easy tasks; the challenge is to extract information from the large amount of data available. Most of this data is unstructured, which makes the application of existing data mining techniques to the Web very difficult. However, new semantic structures which improve the results of Web Mining are currently being developed in the Web. This paper presents how one of these semantic structures - XML topic maps – can be exploited to help users find relevant information in the Web. This paper is organised as follows: first, we introduce XML topic maps in the context of Tim Berners-Lee's Semantic Web vision [2]. Then we show how topic maps allow to characterise Web sites through the definition of a profile. This profile may be used to evaluate the relevance of a web site with regard to a specific request on a traditional search engine. We finally explain how data on the Web can be clustered, organised and visualised in different ways so as to enhance users' navigation and understanding of these documents.

2 XML Topic Maps and the Semantic Web

Finding information on the Web is very difficult. Search engines may return hundreds or more links to users' queries – provided that the right keywords are used. Choosing the most relevant Web sites to explore is not trivial, because no semantics help evaluate the relevance of each hit. The next step is not easier: once a link is chosen, navigation is not always intuitive. Users can get lost easily: they may not find the information they are looking for even though it does exist. Sometimes they do not manage to go back to a page they have already visited. This is due to the lack of structure of the Web. Therefore it is necessary to add structure and semantics as well as to provide a mechanism which allows a more precise description of data on the Web. According to Tim Berners-Lee from W3C [2]:

"The Semantic Web is a vision: the idea of having data on the web defined and linked in a way that it can be used by machines – not just for display purposes, but for using it in various applications."

This Semantic Web can be achieved by adding semantic structures to the current Web. Many candidate techniques were proposed, such as semantic networks [16], conceptual graphs [5], the W3C Resource Description Framework (RDF) [14] and XML Topic Maps [11]. Semantic networks are basically directed graphs (networks) consisting of vertices linked by edges. Edges express semantic relationships between the vertices.

The conceptual graphs theory developed by Sowa [10] is a language for knowledge representation based on linguistics, psychology and philosophy.

RDF data consists of nodes and attached attribute/value pairs. Nodes can be any web resource (pages, servers, basically anything for which you can give a URI), or other instances of metadata. Attributes are named properties of the nodes, and their values are either atomic (character strings, numbers, etc.), metadata instances or other resource. This mechanism allows us to build labelled directed graphs.

Topic maps, as defined in ISO/IEC 13250 [8], are used to organise information in a way that can be optimised for navigation. Topic maps were designed to solve the problem of large quantities of unorganised information. Information is not useful if it cannot be found or linked. In the paper publishing world, there are several mechanisms to organise and index the information contained within a book or document. Indexes allow readers to go directly to the portion of the document that is relevant to their information needs. Topic maps can be thought of as the online equivalent of printed indexes. Topic maps are also a powerful way to manage link information, much as glossaries, cross-references, thesauri and catalogs do in the paper world. Topic Maps allow users to create a large quantity of metadata and tightly interconnected data. They constitute a kind of semantic network above the data themselves.

A new specification which aims at applying the topic map paradigm to the Web is currently being written; this initiative is called XTM (XML Topic Maps) [11]. XML Topic Maps allow to structure data on the Web and therefore make Web mining more efficient.

It was recently proven that the RDF and Topic Map models could inter-operate at a fundamental level [9]. Both standards are concerned with defining relationships between entities with identity. Each language can be used to model the other.

All the techniques described previously have the same goals and many of them are compatible. We decided to further investigate XML Topic Maps and study how they could enhance Semantic Web Mining.

We aim at helping users find relevant information and we contribute at three levels:

1. by evaluating Web sites relevance to users needs based on semantic criteria,
2. by filtering the topic map; the topic map profile constitutes a reference that can be used to select the most semantically significant objects (called *regular* objects). This allows to identify the major subjects which the topic map deals with and to discard less relevant topics.
3. by enhancing navigation on the Web through the aggregation of conceptually related topics and through the visualisation with different scales – or levels of details.

The different steps of topic maps – or Web sites[1] - analysis are represented in figure 1:

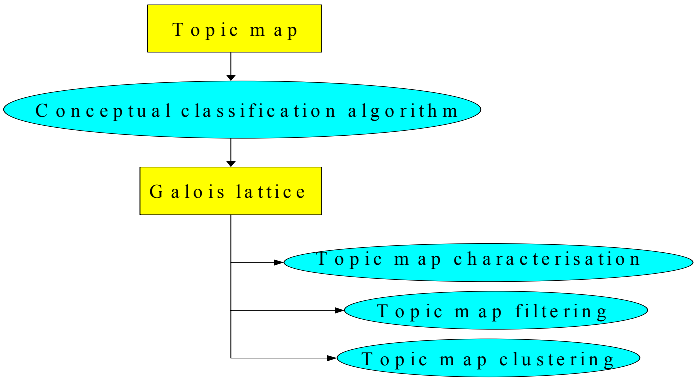


Fig. 1. Web sites analysis algorithm

We propose to achieve the first goal by defining profiles of topic maps – and consequently Web sites. These profiles characterise topic maps – or web sites - and help evaluate their relevance to users' information needs. The computation of this sort of topic map "DNA" and its interpretation are described in section 4.

The topic map may contain topics which are not semantically significant or not much related to others. We call them *singular* topics. They may be eliminated from the topic map so as to clean it, as explained in section 4.2.

[1] In the following, we will use the term "topic map" which is more general than "Web site". Topic maps may apply to any kind of data.

Our third contribution consists in enhancing navigation and information retrieval in a Web site. Information retrieval varies according to the needs of the user. If he looks for an answer to a specific question, query languages (like "tolog" [6]) are adapted. Their strength is to exploit the relationships between objects, which allows to answer questions better. For example, one may seek the Beatles' songs which were not written by John Lennon. This kind of information would be difficult to find with a traditional search engine.

If the subject of interest is clearly identified, it is easy to explore the corresponding topic in the topic map. This topic can be reached through a list of topics, for example an alphabetical list. Tools to navigate in topic maps have been designed so that any topic can be reached in 7 mouse clicks at most.

If the user has no precise question nor any clear subject of interest, none of the search modes described above can apply. This is the case of a beginner user who wishes to have a global understanding of the topic map so as to decide where to start his navigation. Therefore he first needs a simplified view of the topic map, with no detail, then he can decide to see more precise information as his subject of interest gets clearer. Let us compare this to geographical maps: there is no point in displaying very specific data on a map of the world. However, more and more details may be added as the user focuses on some part of the map. We propose to use clustering algorithms to group semantically related topic together at different abstraction levels. Clusters computation and visualisation are described in section 5.

The figure 1 shows that topic maps – or Web sites – characterization, filtering and clustering are deduced from the results of a conceptual classification algorithm based on Formal Concept Analysis and Galois connections. This algorithm is presented in section 3.

3 Conceptual classification algorithm

The starting point of our Web analysis is a conceptual classification algorithm based on Formal Concept Analysis and Galois connections. FCA is a mathematical approach to data analysis which provides information with structure. FCA may be used for conceptual clustering as shown in [4] and [12]. Let us first define a few terms:

- an object is a topic or an association of the topic map,
- the objects have characteristics called properties. We describe how these properties are determined in 3.1.

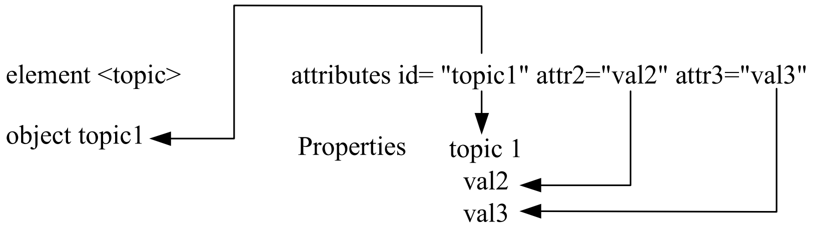
A profile allows to characterise a topic map in a structural way. With this footprint, one can tell if the topic map is specific or general. We can also tell if the objects of a topic map are similar or very different. In order to characterise objects, we use a Galois algorithm to classify the objects conceptually. This algorithm groups objects in concepts according to the properties they have in common. It is very powerful because it performs a semantic classification without having to express semantics explicitly. We will first describe how the objects and their properties are generated from a topic map. Then we will describe Galois lattices and detail the statistical computations made on the objects. We will finally explain how the profile is determined.

3.1 Objects and properties generation

The generation of objects and properties is a 2-steps process.

- First step:
  Every time there is an element with an identifier (that is an *id* attribute), a new object is created. The name of the object is the value of the identifier. As stated in the DTD (Document Type Definition), all topics and associations of the topic map have an identifier, so there will be the same number of objects as the number of topics and associations.
  An object's properties correspond to the values of this object's attributes (including the value of the *id* attribute), as well as the values of his children' attributes. These properties are weighted (for instance, the weight of the values of *instanceOf* attributes may be greater than the weight of the values of *href* attributes).

Generation of object and properties (first step):



Example: consider the following extract of a topic map about music, written by Kal Ahmed[2]:

```
<topic id="t-the-clash">
      <instanceOf>
         <topicRef xlink:href="tt-band"/>
      </instanceOf>
      <baseName>
         <baseNameString>The Clash</baseNameString>
         <variant>
            <parameters>
               <topicRef xlink:href="http://www.topicmaps.org/xtm/1.0/psi-sort"/>
            </parameters>
            <variantName>
               <resourceData>clash the</resourceData>
            </variantName>
```

[2] Kal Ahmed works for Ontopia, http://www.ontopia.net

```
    </variant>
    <variant>
      <parameters>
        <topicRef xlink:href="http://www.topicmaps.org/xtm/1.0/psi-sort"/>
      </parameters>
      <variantName>
        <resourceData>Clash, The</resourceData>
      </variantName>
    </variant>
  </baseName>
</topic>
```

An XML document is made of elements limited by tags and is hierarchically structured. In the example we studied, *topic*, *instanceOf* and *baseName* are elements. An element may have characteristics called attributes. The attributes of an element are declared inside the opening tag of the element. The element *topic* has an attribute *id* with a value *tt-clash*. The element *instanceOf* has no attribute.

When parsing the topic map, we find a topic which has an identifier with the value *t-the-clash*. An object *t-the-clash* is thus created.

In order to determine the properties of these objects, we look for all the attributes of this element. In this case, the only one is the identifier.

Then, we have a look at the children of this element (that is all the XML elements included in the element) to find their attributes. We repeat this for all the children.

In this example, the analysis of this abstract of the topic map creates an object *t-the-clash* with the properties *t-the-clash (*weight e.g. 0.5*)*, *tt-band* (weight e.g. 2) and *http://www.topicmaps.org/xtm/1.0/psi-sort (weight e.g. 0.2)*. The weights shown here correspond to one possible scenario - in which the type of a topic (weight 2) is more important than its name (weight 0.5), its occurrences (weight 0.2) or the associations it is involved in (weight 1).

In the same way, the analysis of the following abstract:

```
<topic id="tt-band">
    <instanceOf>
      <topicRef xlink:href="tt-music"/>
    </instanceOf>
    <baseName>
      <baseNameString>Band</baseNameString>
    </baseName>
</topic>
```

leads to the creation of an object *tt-band* with the properties *tt-band* (weight e.g. 0.5) and *tt-music* (weight e.g. 2).

The last example concerns an association:

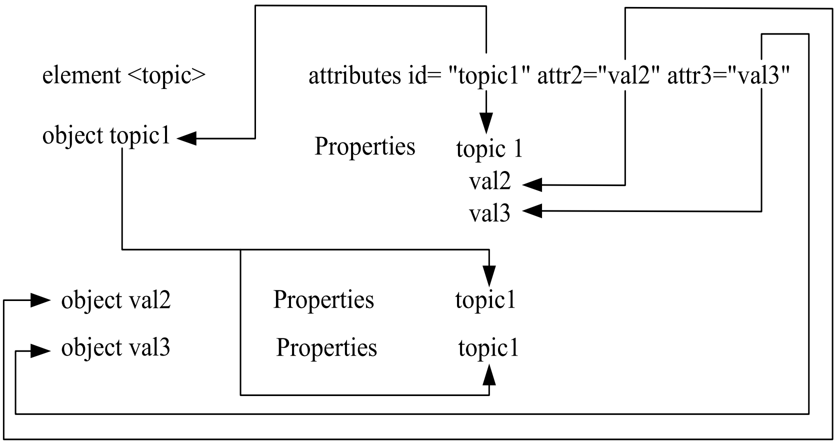
```
<association id="assoc6">
    <instanceOf>
      <topicRef xlink:href="at-recorded"/>
    </instanceOf>
    <member>
      <instanceOf>
        <topicRef xlink:href="tt-band"/>
      </instanceOf>
      <topicRef xlink:href="t-the-clash"/>
    </member>
    <member>
      <instanceOf>
        <topicRef xlink:href="tt-track"/>
      </instanceOf>
      <topicRef xlink:href="t-i-fought-the-law"/>
    </member>
</association>
```

The object *assoc6* is created and has the properties *assoc6* (weight e.g. 0.5*)*, *at-recorded* (weight e.g. 2), *tt-band*, *t-the-clash*, *tt-track* and *t-i-fought-the-law*.

So far, the properties of an object are only intrinsic properties. Indeed, the object *t-the-clash* takes a part in the association *assoc6*, but this does not appear in its properties yet, since the association is not declared inside the topic which has the identifier *t-the-clash*. The second step takes these characteristics into account.

- Second step:

Generation principle of the objects and properties (second step):



The second steps adds non intrinsic properties to the objects by “crossing” the data. In fact, for an object O with a set of properties P, each property P becomes an object with O (amongst others) as a property. The properties of an object are its intrinsic properties and all the properties that were added recursively.

In the previous examples, the object *assoc6* has the properties *assoc6*, *tt-band* and *t-the-clash*. The property *assoc6* is added to the objects *tt-band* and *t-the-clash*. So all the objects know the associations they appear in.

Moreover, the object *t-the-clash* has the property *tt-band*. The data is crossed by adding *t-the clash* to the object *tt-band*. This example illustrates a new type of information, which was not present in the first step: the object *tt-band* knows it has an instance of *t-the-clash*. In the preceeding scenario, *t-the-clash* was the only one to know its superclass.

In the end, *tt-band* has the properties *tt-band* (weight e.g. 0.5)*, tt-music*(weight e.g. 2)*, t-the-clash* (weight e.g. 1), *assoc1* (weight e.g. 1), *assoc2* (weight e.g. 1) and *assoc6* (weight e.g. 1). The object *t-the-clash* has the characteristics *http://www.topicmaps.org/xtm/1.0/psi-sort* (weight e.g. 0.2), *tt-band, t-the-clash* (weight e.g. 0.5)*, assoc1* (weight e.g. 1)*, assoc2* (weight e.g. 1) and *assoc6* (weight e.g. 1).

Note that the properties *assoc1* and *assoc2* correspond to other associations in which *tt-band* and *t-the-clash* appear. These associations are present in the topic map but not in the extracts we presented.

3.2 Introduction to Galois lattices

The notion of Galois lattice for a relationship between two sets is the basis of a set of conceptual classification methods. This notion was introduced by Birkhoff in [3] and by Barbut and Monjardet in [1]. Galois lattices consist in grouping objects into classes that materialise concepts of the domain under study. Individual objects are discriminated according to the properties they have in common. This algorithm is very powerful as it performs semantic classification. Topic maps are semantic structures themselves, but they may be very large and complex, so this algorithm is interesting to extract more semantics from them. The algorithm we implemented is based on the one that was proposed in [7].

Let us first introduce Galois lattices basic concepts.

Let two finite sets E and E' (E consists of a set of objects and E' is the set of these objects' properties), and a binary relation R ⊆ E x E' between these two sets. Figure 2 shows an example of binary relation between two sets. According to Wille's terminology [13], the triple (E, E', R) is a formal context which corresponds to a unique Galois lattice. It represents natural regroupings of E and E' elements.

Let P(E) be the powerset of E and P(E') the powerset of E'. Each element of the lattice is a couple, also called concept, noted (X, X'). A concept is composed of two sets X ∈ P(E) and X' ∈ P(E') which satisfy the two following properties :

$$\begin{gathered} X' = f(X) \text{ where } f(X) = \{ x' \in E' \mid \forall x \in X, xRx' \} \\ X = f'(X') \text{ where } f'(X') = \{ x \in E \mid \forall x' \in X', xRx' \} \end{gathered} \qquad \textbf{(1)}$$

A partial order on concepts is defined as follows :

Let C1=(X1, X'1) and C2=(X2, X'2),

$$C1 < C2 \Leftrightarrow X'1 \subseteq X'2 \Leftrightarrow X2 \subseteq X1 \qquad \textbf{(2)}$$

This partial order is used to draw a graph called a Hasse diagram, as shown on figure 2. There is an edge between two concepts C1 and C2 if C1<C2 and there is no other element C3 in the lattice such as C1<C3<C2. In a Hasse diagram, the edge direction is upwards. This graph can be interpreted as a representation of the generalisation / specialisation relationship between couples, where C1<C2 means that C1 is more general than C2 (and C1 is above C2 in the diagram).

R	a	b	c	d	e	f	g	h
1	1	0	1	0	0	1	0	1
2	1	0	1	0	0	0	1	0
3	1	0	0	1	0	0	1	0
4	0	1	1	0	0	1	0	1
5	0	1	0	0	1	0	1	0

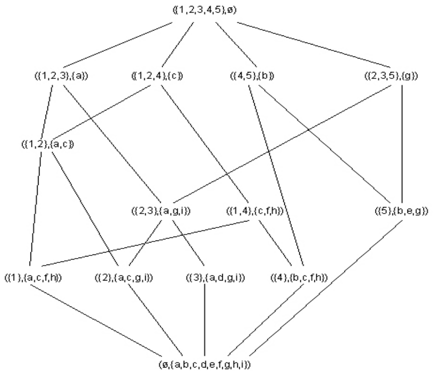


Fig. 2. Binary relationship and associated Galois lattice representation (Hasse diagram)

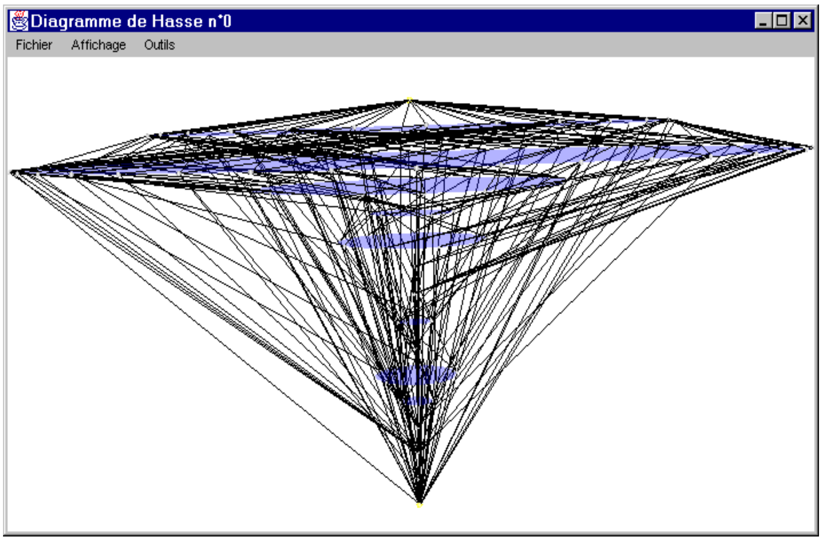


Fig. 3. Concept lattice of a topic map about music

The concept lattice shows the commonalities between the concepts of the context. The first part of a concept is the set of objects. It is called "extension". The second set – the intention - reveals the common properties of the extension's objects. The figure 3 shows the concept lattice generated from our example topic map about music.

4 Topic maps characterisation: conceptual profile

4.1 Calculating the statistics for every object.

We calculate statistics for every object of the topic map. We compute a weighted mean of these statistics. Each object has a weight which is assigned according to its importance in the topic map (the number of occurrences of the object in the XML source file).

Consider an object O. It is characterised by a vector with 6 components:

- The first component (A1) is the percentage of concepts of the sub-lattice where the object is present in the list of extensions. This value tells if O is present in many concepts of the lattice. A low value for A1 may indicate that O has few common characteristics with other objects. However, the other components allow to increase our knowledge.
- The second component is the maximum number of objects with which O is grouped, divided by the total number of objects. We have to select the concept containing O and with the largest number of objects. We add a constraint on this concept: it must contain at least one property. Indeed, we wish to group objects with common properties. The component A2 shows if O is grouped with many other objects. However, this value is a maximal value. The validity of A2 must be checked using A3.
- A3 is the mean number of objects with which O is grouped divided by the number of objects. This time we can tell if O is linked to a large number of objects and determine the significance of A2. If A3 is high, then there is a concept with O and many other concepts. On the other hand, if A3 is low, O is grouped with very few objects. The selected concept is thus an exception and we should not base our analysis on it.
- Let S be the set of objects which are grouped with O in one –or more- concepts of the lattice; these objects have at least one of O's properties. A4 is the maximum number of properties O shares with the objects contained in S, divided by the total number of objects. This component is deduced from the concept containing the object O and which has the greatest number of properties. Again, we add a constraint on this concept: it must contain at least two objects, that is at least one object different from O. We want to evaluate the number of shared properties, thus we need at least one object with which O shares them. A4 tells if the objects which are close to O share many common properties with O or not. Objects are more similar when they share an increasing number of properties. This similarity

can either be structural or conceptual. However, this value is a maximum number which must be validated with A5.

- A5 is the mean number of properties O shares with other objects, divided by the total number of properties. This tells the degree of significance of A4.
- Finally, A6 is about the topic map itself, and not about the lattice. It is the number of occurrences of the object in the topic map divided by the number of occurrences of objects of the same type (topic or association). A6 is used to compute the topic map's profile. This profile represents the characteristics of a mean object. Each component of this vector is the mean of the components of each object in the topic map, with a weight A6 given to each of these objects. Thus, objects with a high number of occurrences in the topic map will influence the profile much more than objects with few occurrences.

Note that the five first components are deduced from an analysis of the lattice whereas the last component only depends on the XML document.

4.2 Topic map – Web site - profile and selection of objects

When the statistics have been computed for every topic and association, the profile can be deduced. It is a vector for which each component is a mean of the components of all the objects with the weight A6 of each object. For N objects O_1, O_2, … O_N, each component A_i of the profile vector P is computed as follows:

$$P.A_i = \sum_{j=1}^{N} O_j.A_i * O_j.A_6 \quad \textbf{(3)}$$

where $O_j.A_i$ is the component A_i of the j-th object.

We wish to keep the most relevant objects, that is the ones which share "many" common properties with "many" other objects. These objects are called *regular* objects, they are semantically more significant than others. The significance of the words "many" (properties) and "many" (objects) is given by the topic map profile. A regular object is associated to at least as many objects and shares as many properties as the profile.

Among the statistics presented in section 4.1, the values A3 and A5 are more relevant than A2 and A4: maximum values may not give a reliable information because they may correspond to an exception. The comparison between the objects and the profile is thus done using the components A3 and A5.

A regular object O must verify the following conditions:

$$\begin{aligned} O.A_1 &\geq profile.A_1 \\ O.A_2 &\geq profile.A_2 \end{aligned} \quad \textbf{(4)}$$

This should be refined using the standard deviation. The standard deviation for A3 is the mean distance between an object's value of A3 and the profile's value of A3.

$$std.dev.A_3 = \frac{\sum_{j=1}^{N} |O_j.A_3 - P.A_3|}{N} \tag{5}$$

For A5, the standard deviation is computed in the same way:

$$std.dev.A_5 = \frac{\sum_{j=1}^{N} |O_j.A_5 - P.A_5|}{N} \tag{6}$$

Thus, a regular object is defined as follows:

$$\begin{gathered} O.A_1 + std.dev.A_1 \geq P.A_1 \\ O.A_2 + std.dev.A_2 \geq P.A_2 \end{gathered} \tag{7}$$

The regularity conditions can be changed (to be more or less restrictive) with a coefficient (C). Thus, a regular object meets the two following requirements:

$$\begin{gathered} O.A_1 + C \times std.dev.A_1 \geq P.A_1 \\ O.A_2 + C \times std.dev.A_2 \geq P.A_2 \end{gathered} \tag{8}$$

A non regular object is called a *singular* object –it conveys little semantics. When the objects of the topic map are submitted to these conditions, singular objects are eliminated. When C increases, more objects are suppressed since the conditions are harsher.

After this selection, we have a new list of objects which are used as an input for the Galois classification algorithm. A new lattice is generated and the statistics computed on this new panel of objects provide a new profile. We can thus select once again the regular objects for this new footprint of the topic map. The new regular objects are used again as an input for the Galois algorithm, etc. until all the objects become regular. This happens when no object is eliminated. The algorithm stalls and we get a stable list of regular objects which we must group together.

4.3 Results

Several topic maps – of different sizes and subjects - were analysed. The figure 4 displays the distribution of objets in three topic maps. The coordinates of the center of a disk correspond to the values of A3 and A5 attributes. The diameter of a circle is proportional to the number of objects which have these values for A3 and A5. All the objets of the *simple* topic map are very close. This topic map is qualified of "homogeneous", which means that all topics have the same semantic significance. *Music* and *icc* are "heterogeneous" structures. The objects in the lower left corner have low values for A3 and A5: they are "singular" - not much related to other objects in the topic map. These topic maps can be filtered easily by eliminating these singular objects.

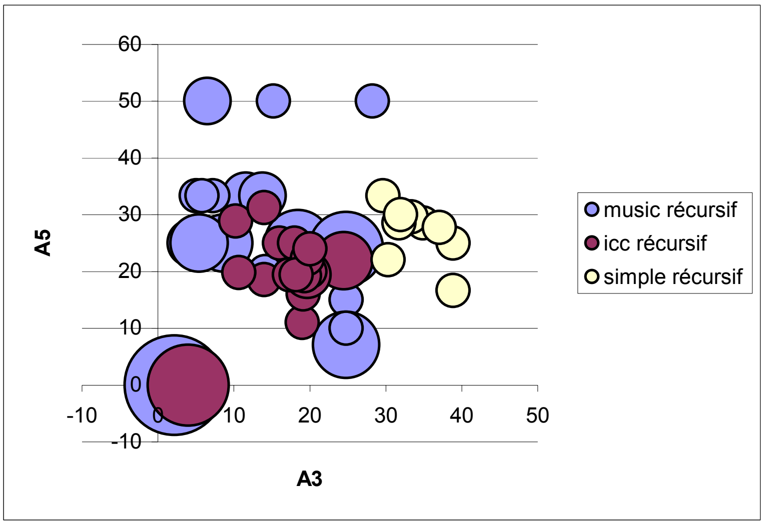


Fig. 4. Topics conceptual distribution

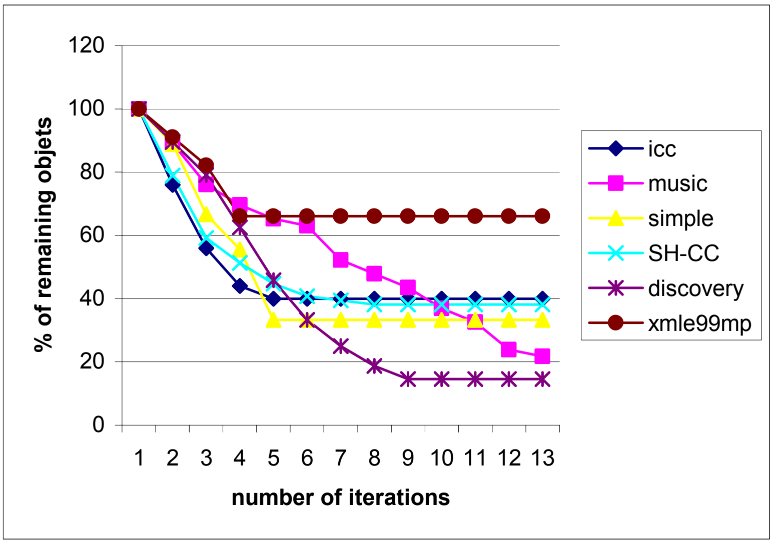


Fig. 5. Topic map filtering

The figure 5 illustrates the filtering of six topic maps. Some topic maps can be simplified a lot; this is the case of *discovery*. On the other end, after the last iteration, *xmle99mp* still contains almost 70% of its topics. This means that it is more difficult to filter this topic map: all topics have the same semantic value.

5 Topic maps clustering and visualisation

5.1 Clustering algorithm

The Galois lattice which is generated from a topic map contains some concepts which are made up of a set of topics which share common properties. The lattice gives an exhaustive description of the input data and the number of concepts generated may be very high. The concept lattice shown in the figure 3 is quite complex although it was generated from a small topic map (which contains 46 objects). We wish to group topics together into clusters in order to provide different level of detail (or scales) of the topic map. We propose to extract a tree from the Galois lattice. The concepts contained in this tree are the clusters. Thus, we have a hierarchy of clusters. The root of the tree contains all the topics; it is a gross cluster which provides no additional information. The next level groups some topics together, the next level executes a finer grouping of topics, etc. The number of levels of detail is given by the depth of the tree.

Many clustering algorithms exist; we chose to implement a clustering algorithm based on Galois conceptual classification. The clusters we generate are thus conceptually and semantically relevant. This algorithm also allows us to use the generalisation/specialisation relationship inherent to the Galois lattice.

To build the tree of clusters, we start from the representation which provides the greatest level of detail. Every cluster corresponds to an object: the objects are not grouped together. We begin to construct the leaves of the tree: these clusters correspond to the fathers of the upper bound of the lattice (which is represented at the bottom of the lattice on the Hasse diagram). This is the most specific level.

For each leaf, we select one unique father which is a generalisation of the concept. This selection is done according to a hierarchy of criteria which will be developed in the following. One father is selected for each selected node, and so on until the lower bound of the lattice is reached. At the end of this process, a tree is created. Each level of the tree contains clusters which correspond to a level of detail.

We defined a hierarchy of selection criteria when a concept has several fathers in the lattice.

- first, we consider the distance between each father and the lower bound of the lattice (this distance corresponds to the minimum number of edges between them).

- if one of the fathers' distance to the lower bound is smaller than the others, this node is selected. Being at lower distance from the lower bound means that this concept is semantically richer.
- if several nodes are at a minimum distance from the lower bound, we compare the sum of the weights of the properties contained in their intention. The node with the highest value is selected.
  - if several fathers meet this requirement, the algorithms chooses the one which minimises the total number of branches in the tree. If this condition is not unique, different scenarios are considered, one for each possible father.

5.2 Clusters analysis

Once the tree of clusters is generated, different measures may be computed, e.g. the proportion of concepts of the initial lattice which were not selected to be clusters.

The depth of the tree is interesting because it indicates the number of navigation levels that may be provided to the user. We also study the distribution of clusters at each abstraction level. If a cluster has no father, it means that it cannot be generalised. On the other hand, a cluster with no children corresponds to the most specific level.

We may also compute the distances between clusters. The distance between two clusters may be the average – or minimum, or maximum – distance between two objects (one in each cluster). Let O1 and O2 be two objects. Let P1 be the set of properties of O1 and P2 the set of properties of O2. Let INTER be the intersection of P1 and P2, and UNION the union of P1 and P2. The similarity between O1 and O2 is defined as:

$$S(O1,O2)=\frac{\sum_{i=1}^{card(INTER)} wi}{\sum_{j=1}^{card(UNION)} w'j} \quad \textbf{(9)}$$

The distance between O1 and O2 is given by (2):

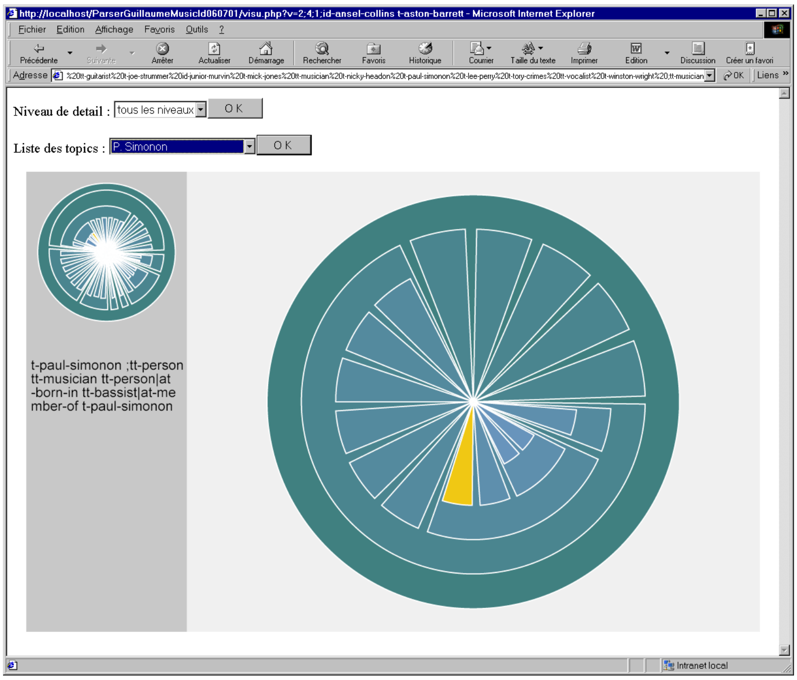
$$D(O1,O2)=100-\frac{1}{S(O1,O2)} \quad \textbf{(10)}$$

5.3 Clusters representation

The levels of detail are symbolised by different colours. At each abstraction level, clusters are represented by portions of a disk, as shown in figure 6. Each cluster's size is proportional to the number of children this concept has. When the pointer of the mouse is over a cluster, its extension – the set of topics contained in this cluster – or intention – the set of these topics' properties – is displayed. When the user clicks on a part of the disk, this cluster becomes the current context – i.e. the whole disk -

and its content is displayed in greater detail. The disk in the upper left corner represents a global view of the topic map before focusing on a specific cluster.

The figure 6 shows the results of this clustering algorithm on our example topic map about music. These representations are SVG (Scalable Vector Graphics) graphics [15]. SVG is a language for describing two-dimensional graphics in XML. SVG drawings can be interactive and dynamic. SVG leverages and integrates with other W3C specifications and standards efforts. By leveraging and conforming to other standards, SVG becomes more powerful and makes it easier for users to learn how to incorporate SVG into their Web sites.



http://localhost/ParserGuillaumeMusicId060701/visu.php?v=2;4;1;id-ansel-collins t-aston-barrett - Microsoft Internet Explorer

Adresse %20tt-guitarist%20t-joe-strummer%20id-junior-murvin%20t-mick-jones%20tt-musician%20t-nicky-headon%20t-paul-simonon%20t-lee-perry%20t-tory-crimes%20tt-vocalist%20t-winston-wright%20;tt-musician

Niveau de detail : tous les niveaux

Liste des topics : P. Simonon

```
t-paul-simonon ;tt-person
tt-musician tt-person|at
-born-in tt-bassist|at-me
mber-of t-paul-simonon
```

Fig. 6. Clusters visualisation

6 Conclusion and further work

This article presented how XML topic maps could be exploited to help users find relevant information on the Web. This contribution is at several levels: first, we characterise Web sites by defining their profile. This may be used to evaluate Web sites relevance with regard to a specific query. Second, our analysis identifies topics that have no interest – semantically speaking – which allows to "clean" the topic map. Finally we showed how we could enhance navigation by clustering Web pages and displaying them with different levels of details.

These results were deduced from the analysis of Galois lattices generated from Web sites with a conceptual classification algorithm. This algorithm is very powerful as it groups topics semantically.

In the future, we will study Web sites clusters in more details. For example, we noticed that some of the clusters are less relevant than others; it may thus be possible to further filter the Web site if it is really too large.

We will also investigate how ontologies may be used to characterise our clusters. Galois algorithm generates clusters which have a semantic value without expressing this semantics explicitly. Ontologies may help us make this information explicit.

7 Références

1. Barbut, M., Monjardet, B., *Ordre et classification*, Algebre et combinatoire, Tome 2, Hachette, 1970.
2. Berners-Lee, T., *A roadmap to the Semantic Web*, **http://www.w3.org/DesignIssues/Semantic.html**, Sept 1998.
3. Birkhoff, G., *Lattice Theory*, First Edition, Amer. Math. Soc. Pub. 25, Providence, R. I., 1940.
4. Carpineto, C., Romano, G., *Galois: An order-theoretic approach to conceptual clustering*, Proc. Of the 10th Conference on Machine Learning, Amherst, MA, Kaufmann, pp. 33-40, 1993.
5. Chein, M., Mugnier M.-L., *Conceptual Graphs : Fundamental Notions*, Revue d'intelligence artificielle, Volume 6 - n°4/1992, pp365-406, 1992.
6. Garshol, L. M., *"tolog" – A Topic Map Query Language*, XML Europe 2001, Berlin, Germany, 21-25 May 2001.
7. Godin, R, Chau, T.-T., *Incremental concept formation algorithms based on Galois Lattices*, Computational intelligence, 11, n° 2, p246 –267, 1998.
8. International Organization for Standardization, ISO/IEC 13250, *Information Technology-SGML Applications-Topic Maps*, Geneva: ISO, 1998.
9. Moore, G., *RDF and Topic Maps, An Exercise in Convergence*, XML Europe 2001, Berlin, Germany, 21-25 May 2001.
10. Sowa, J. F., *Conceptual Information Processing in Mind and Machine*, Reading, Massachusetts, Addison-Wesley, 1984.
11. TopicMaps.Org XTM Authoring Group, *XTM: XML Topic Maps (XTM) 1.0: TopicMaps.Org Specification*, 3 March 2001.
12. Wille, R., *Line diagrams of hierarchical concept systems*, Int. Classif. 11, pp. 77-86, 1984.
13. Wille, R., Concept lattices and conceptual knowledge systems, Computers & Mathematics Applications, 23, n° 6-9, pp. 493-515, 1992.
14. World Wide Web Consortium, Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, 22 February 1999.
15. World Wide Web Consortium, Scalable Vector Graphics (SVG) 1.0 Specification, W3C Candidate Recommendation, 2 November 2000.
16. Woods, W.A., What's in a link: foundations for semantic networks, In D.G. Bobrow and A.M. Collins, (Eds.), Representation and Understanding: Studies in Cognitive Science,. New York: Academic Press. p. 35-82, 1975.