In the quest of specific-domain ontology components for the semantic web

JRG Pulido[*] SBF Flores PD Reyes RA Diaz JJC Castillo
Faculty of Telematics, University of Colima, México
{jrgp,medusa,damian,acosta,juancont}@ucol.mx

Keywords: Ontology Learning, Semantic Web, Self-Organizing Maps

***Abstract*—** This paper describes an approach we have been using to identify specific-domain ontology components by using Self-Organizing Maps. These components are clustered together in a natural way according to their similarity. The knowledge maps, as we call them, show colored regions containing knowledge components that may be used to populate an specific-domain ontology. Later, these ontology may be used by software agents to carry out basic reasoning task on our behalf. In particular, we deal with the issue of not constructing the ontology from scratch, our approach helps us to speed up the ontology creation process.

1 Introduction

The semantic web, requires that the information contained into digital archives is structured [4]. In the last few years a number of proposals on how to represent knowledge via ontology languages have paraded [42, 10, 17, 15, 30]. Now that OWL has become an standard [25], the real challenge has started. Slowly but surely the web is to be populated with structured knowledge that will allow software agents to act on our behalf. Converting the current web into the next generation one, the Semantic Web, is to take much longer if no semi-automatic approaches are taken into account to carry out this enterprise. This is what our paper is all about. The remainder of this paper is organized as follows. In section 2 some related work is introduced. Our approach is outlined in section 3. Results are presented in section 4, and conclusions and further work in section 5.

2 Related Work

Vast amounts of knowledge are currently available on the Internet and its quantity is growing rapidly. This has underlined the weakness of current mechanisms and techniques used to give users access to this knowledge. The difficulty of extracting, filtering, and organizing knowledge from expert domains has challenged the research community which is now extremely interested in reusing knowledge. The fundamental problem is how to extract formal and consistent knowledge representations suitable for specialised tasks such as inference. In the context of the semantic web, one of the most important challenges is the mapping of large amounts of unstructured information, suitable for humans,into formal representation of knowledge [4]. In the next subsections we have a brief look at some related work on Ontologies and Self-Organizing Maps which are the framework of our approach.

2.1 Ontologies

An ontology may be referred to as an agreed conceptualization. In other words, it is a set of elements that, as a whole, allow us represent real world domains, an academic one for instance. Must be said that representing knowledge about a domain as an ontology is a challenging process which is difficult to achieve in a consistent and rigorous way. It is easy to lose consistency and to introduce ambiguity and confusion [3]. The ontology life cycle usually requires the following [8, 29, 9, 7, 46] activities (Fig.1):

Gathering The acquisition and collection of the knowledge from the domain in which we are interested. It usually involves dealing with unstructured data in natural language from digital archives.

Extraction This requires background knowledge for creating taxonomies of the domain in a semi-automatic way. Learning techniques may be applied by the knowledge engineer for this task.

Organization Imposing a structure on the knowledge acquired and generating formal representations of it for later being used by software agents or humans.

Merging Defining mapping rules to facilitate interlingua exchange relating information from one context to another. This activity is as important as $Extraction$. It can be referred to as finding commonalities between two knowledge bases and deriving a new knowledge base.

Refinement Improving structure and content of the knowledge about the domain by eliciting knowledge from the domain experts. It amends the knowledge at a finer granularity level.

[*] Corresponding author, Tel/fax: +52 312 316 1075

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

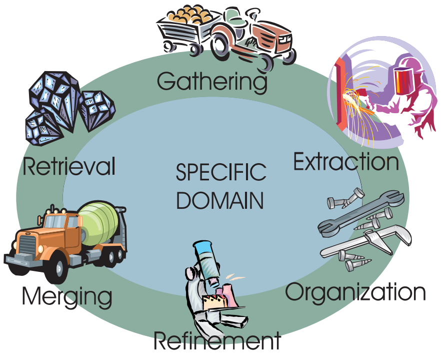


Figure 1: The ontology life cycle.

Retrieval Communicating the knowledge to users in such a way that computational mechanisms also can have access to it.

A number of interesting approaches can be found in the literature. For instance, in [16] the use of the so-called *Simple HTML Ontology Extension* (SHOE) in a real world internet application is described. A similar approach is presented in [2]. Most *tag-annotated* web pages tend to categorize concepts, therefore there is no need for complex inference rules to perform automatic classification. One of the most common uses of an ontology is to support the development of agent-based systems for web searching [31, 13]. It has also been use to characterize scientific Web communities [44], and in helping to give sense to unstructured text [27].

2.2 Self-Organizing Maps

We start this section, by describing some basic ideas related to Self-Organizing Maps (SOM). Clustering is the unsupervised process of grouping patterns, observations, data items, or feature vectors [18]. This problem has been addressed in different contexts and by researchers since the 60's in many disciplines, reflecting its broad appeal and usefulness as one of the steps in exploratory data analysis. A pattern set can be denoted as $S = \{d_1, .., d_m\}$. The i^{th} pattern in S is denoted as $d_i = \{a_{i1}, .., a_{in}\}$, $d_i \in \Re^n$. This pattern set is viewed as an *m*×*n* matrix. The individual scalar components a_{ik} are called $features$ or $patterns$. Some classic approaches to the problem include partitional methods [37], hierarchical agglomerative clustering [40], and unsupervised bayesian clustering [34]. A widely used partitional procedure is the k-means algorithm [19]. A problem with this procedure is the selection of k a priori. An alternative to these methods is SOM which does not make any assumptions about the number of clusters a priori, the probability distributions of the variables, or the independence between variables.

Perhaps the most well-known project is *WEBSOM2* [22]. This is an organization, searching and browsing system. In this case, a document map is presented as a series of HTML pages facilitating exploration. A specified number of best-matching points are marked with a symbol and can then be used as starting points for browsing. Some other approaches use SOM as a clustering and visualization software tool [24, 28, 39]. They have also been used to estimate mobile location [47], pattern recognition [1], and gene clustering [48].

3 Methods

Our software is written in Java, which offers robust, multiplatform, and easy networking functionalities. Being a object-oriented programming language, it also facilitates reuse as well. Speed is not an issue anymore as computer processors are faster and faster. Java and its various APIs are powerful enough for constructing ontology software systems. The idea of combining ontologies and semantic maps has motivated our work. For the semantic web to become a reality, we need to transform the current web into a web where software agents are able to negotiate and carry out trivial tasks for us. Doing this manually, would mean a bottleneck for the semantic web. We need software tools that help us accomplish this enterprise.

Our system consists of two applications: Spade and Grubber [6, 5]. The former pre-processes html pages and creates a document space. The latter is fed with the document space and produces knowledge maps that allow us visualize ontology components contained from a digital archive. They may later be organized as a set of $Instances$, $Relations$, and $Functions$. Problem solvers may use those for inferring new data [11, 12, 46, 33].

3.1 The Algorithm

SOM can be viewed as a model of unsupervised learning and an adaptive knowledge representation scheme [38]. Adaptive means that at each iteration a unique sample is taken into account to update the weight vector of a neighbourhood of neurons [21]. Adaptation of the model vectors take place according to the following equation:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \qquad (1)$$

where t $\in \mathcal{N}$ is the discrete time coordinate, $m_i \in \Re^n$ is a node, and $h_{ci}(t)$ is a neighbourhood function. The latter has a central role as it acts as a smoothing kernel defined over the lattice points and defines the stiffness of the surface to be fitted to the data points. This function may be constant for all the cells in the neighbourhood and zero elsewhere. A common neighbourhood kernel that describes a natural mapping and that is used for this purpose can be written in terms of the Gaussian function:

$$h_{ci}(t) = \alpha(t) \exp(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}) \qquad (2)$$

where $r_c, r_i \in \Re^2$ are the locations of the winner and a neighbouring node on the grid, $\alpha(t)$ is the learning rate ($0 \leq \alpha(t) \leq 1$), and $\sigma(t)$ is the width of the kernel. Both $\alpha(t)$ and $\sigma(t)$ decrease monotonically.

The major steps of our approach are as follows:

a) Produce a *document space* A document space is created with the individual vector spaces.

b) Construct the SOM By using a suitable number of cells and iterations the map is trained with the $docuspace$.

Once the SOM has been trained, ontology components can be seen and examined clustered together. One important difference between our approach and Kohonen's is that we do not use *average context* [38, 20] to create the $docuspace$. In other words, Kohonen uses phrases for the creation of the lexicon which turns out into a much bigger docuspace. We have used one-word terms for the lexicon. This helps us reduce the dimensionality of the dataset, as contextual information is clustered together anyway. Preliminary results were surprisingly close to our intuitive expectations. After this, some other ontology tools such as editors can be used to organize this knowledge. Finally, it can be embedded into the digital archive where it was extracted from by means of any of the ontology languages that exist.

4 Results

This section presents two experiments that we have carried out. Firstly, we compare our results, from a very small data set, to those of other authors. In [38, 36] this *dataset* is presented, rather poorly compared to our approach, and analysed. Our approach uses a 4x4 SOM and presents the same data by using colored areas. Then the results from applying our approach to a bigger digital archive, for identifying ontology components, are shown. Both subsections present the data in two ways, what we have called the Entity Map and the Attribute Map. The former shows $entities$ clustered together as main features and their corresponding attributes as subfeatures. The latter, exhibits $attributes$ as main features and their corresponding entities as subfeatures.

4.1 Animals

This scenario, as mentioned, is real simple. Basically it contains two kinds of animals, namely birds and mammals. These have been clustered together. A brief definition of the classes of this scenario is given as follows.

Birds These are creatures with feathers and wings. Most birds can fly. The following are the birds that are part of this scenario: dove, hen, duck, goose, owl, hawk, eagle. Birds In this scenario are described in terms of their attributes, so a number of subclasses may be identified. For instance, some of these birds fly, some other do not. It is interesting to note that in this particular dataset all the birds have feathers and have two legs. Are these the most important attributes of birds? That is for the experts to decide for we all know that humans have also two legs and penguins have no feathers at all. But we also know that they are not part of this dataset.

Mammals These on the other hand are creatures that give birth to their young and feed them with milk. The following are the mammals that are part of this scenario: cow, fox, dog, wolf, cat, tiger, lion, horse, zebra. Mammals are also described in terms of their attributes such that new subclasses may be identified. For instance, some of these mammals hunt, some other do not. Note that in this dataset all the mammals have four legs and hair. Are these attributes the ones that define mammals? Probably not. But again an expert will decide. We all know that mammals feed their young with milk and this attribute does not appear in the dataset.

Background knowledge would allow even children to classify, or at least identify, some of these animals and perhaps draw a basic taxonomy of the domain. Experts on the other hand may elaborate some more complex taxonomies from this small scenario by means of browsing our knowledge maps, either the entity map or the attribute map. Other more complex scenarios from the animal kingdom would include reptiles, fish, amphibian, insects, or mollusc, not included here [41]. From the experiments we have found that one dominant characteristic amongst the animals is their size, e.g. birds are small, mammals come in two sizes. On the other hand, birds of prey and hunting mammals, small animals with feathers, big animals with hooves, and the ones with four legs and hair are also clustered together. This is consistent with earlier tests carried out on the dataset. Both SOMs are shown in Figure 2. It must be noticed that the vector spaces for *zebra and horse, and owl and hawk* are equal. The ones for *hen and duck* are approximately equal. Similarly, the vector spaces for the Attributes *feather and two legs, and hair and four legs* are equal. That is why some areas overlap and produce a combination of colorings.

4.2 Digital Archives

For our second analysis a set of web pages of the Computer Science Department[1]at Nottingham University has been used. A University consists of a number of entities, for instance $school$ and $person$. One School has generally more that one *research group*. A person usually plays more that one $role$ within the school, $member$ of a research

[1] http://www.cs.nott.ac.uk

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

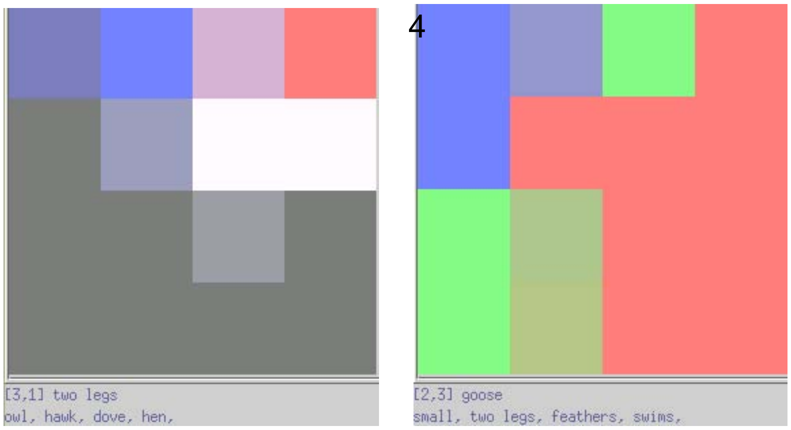


Figure 2: An animal dataset and its ontology components. **Left**. Attribute Map: main feature and subfeatures of cell [3,1]. **Right**. Entity Map : main feature and subfeatures of cell [2,3]. Every single cell contains its own data.

group for instance, and has a number of $publications$. A specific domain such an academic domain requires knowledge from the experts in order to produce a complex ontology. Before going any further, the definitions of some of the domain ontology components that we are looking for are briefly given.

Department The computer science department is one of the many departments in the University of Nottingham. The department comprises the *School of Computer Science and Information Technology (CSiT)* and the *Informatics Institute of Information Technology* (formerly the ICL Institute of IT). They both offer a number of postgraduate programs and courses for undergraduate students.

Lectures These are talks that some members of the school give to teach people about a particular module, for instance from the Module called *Introduction to Artificial Intelligence* (G5AIAI) some lectures are: Blind Searches, Game Playing, and Neural Networks. For easy referencing, lectures have a course code, shown above in parenthesis.

Tutorials These are regular sessions between a tutor and a number of students for discussion of a subject being studied. PhD students usually play the role of tutors and undergraduates are tutees. Sometimes exercises are solved during these sessions.

Coursework Coursework are assignments that students do during the course of Modules. These count towards their final grades. Each module includes at least one coursework. Exercises and sometimes essays are also considered as part of coursework.

Laboratories Laboratories are rooms containing specific equipment for students to actually put in practice what they have learnt during their courses. Computers are the main equipment needed in the field of Computer Science. Professors and Lectures arrange a number of practicals for the students to be carried out in the labs to reinforce knowledge about a specific subject.

Exercises Exercises are particular pieces of tasks that students work out. They are designed to help students learn particular skills about a subject. They usually are supervised by PhD students during tutorials.

Surveys Surveys are detailed accounts about topics. They are usually written by researchers, including PhD students, in the form of literature reviews before research proposals or as introductory chapters in theses.

Industry Industries make and supply particular products, or provide and distribute particular services for the community. Industries are interested in improving their processes of producing and providing better products and services. In order to improve those processes, Industries support financially a number of academic research initiatives within the Universities for research to be done for them.

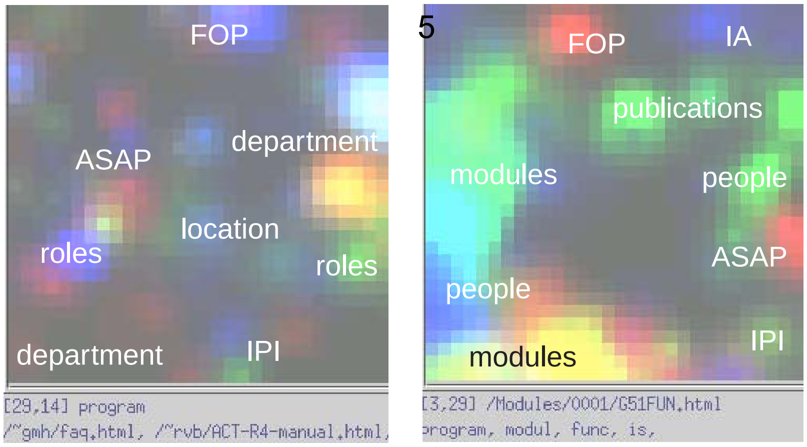
Again, background knowledge is very important. Browsing the SOMs gives us a clear idea and helps us understand what the domain is all about. For instance we can readily identify $people$ within the domain, their $roles$[2], $modules$[3] that are taught, and $research$[4] interests of the members of the school. Terms like *ieee, confer(ence), proceed(ings), workshop, journal, spring(er)*[5], and even the location of the school (wollaton, jubil(e), campu(s), nottingham), and how to reach it (driv(e), rout(e), map, direc(tion),

[2] Professor, student, head, tutor, assistant, lecturer.

[3] Databases, java, data structures, artificial intelligence.

[4] Scheduling, software agents, functional programming.

[5] Terms truncated by the *stemmer*.



```
[29,14] program
/~gmh/faq.html, /~rvb/ACT-R4-manual.html,
```

```
[3,29] /Modules/0001/G51FUN.html
program, modul, func, is,
```

Figure 3: An academic domain and its ontology components. **Left**. Attribute Map: main feature and subfeatures of cell [29,14]. **Right**. Entity Map : main feature and subfeatures of cell [3,29]. Every single cell contains its own data.

guid(e)) are clustered together. Further subcategories are also visualized, for instance, within the *Image Processing and Interpretation Research Group* we found terms like *text, vision, ai, colour, recognition, image* grouped (Fig.3). The domain expert, by using these ontology components, is able to construct a basic ontology of the domain. By using some other software tools, for organizing the ontology and the found components (fig.1), the knowledge engineer may tailor specific-domain ontologies for software agent applications by means of ontology languages [42]. For instance, this ontology can be embedded later into web pages for agents to use and perform intelligent information retrieval.

5 Conclusions

A common ontology enables collaborators, software agents and humans for instance, to work together with a minimal risk of misunderstanding. Background knowledge allow us to classify or at least identify some of these elements and extract a basic taxonomy from the domain. But, it is for the experts to decide whether the elements that have been identified are valid ontology components from the domain. This is very important and must be emphasized as there is not a formal evaluation technique other than some ontology validation software tools. Once those components have been validated, more complex taxonomies may be elaborated. The use of some other software tools, for organizing the ontology and the found components, we can tailor specific-domain ontologies for software agent applications. Must be said that the acquisition and representation of knowledge needs to take into account the complexity that is present in domains as well as the needs of users or agents carrying out the search. Principled techniques that allow the ontological engineer to deal with the problems caused by such complexity need to be developed, and the ideas in this paper have shown promise as avenues of investigation to populate the new web, the so called semantic web, where software agents will carry out reasoning and inference tasks in our behalf.

References

[1] N Allison and H Yin. Self-organizing maps for pattern recognition. In E Oja and S Kaski, editors, *Kohonen maps*, pages 111–120. Elsevier Sci, Amsterdam, 1999.

[2] R Benjamins et al. $(KA)^2$: Building ontologies for the internet: A midterm report. *Int.J.Human-Computer Studies*, 51(3):687–712, 1999.

[3] R Brachman. What is-a and isn't: An analysis of taxonomic links in semantic networks. *IEEE Computer*, 16(10):10–36, 1983.

[4] L Crow and N Shadbolt. Extracting focused knowledge from the Semantic Web. *Int.J.Human-Computer Studies*, 54:155–184, 2001.

[5] D Elliman and *JRG Pulido*. Visualizing ontology components through self-organizing maps. In D Williams, editor, *6th International Conference on Information Visualization (IV02), London, UK*, pages 434–438. IEEE Computer Soc.Press, Los Alamitos, 2002.

[6] D Elliman and *JRG Pulido*. Self-organizing maps for detecting ontology components. In H Arabnia et al., editors, *The 2003 Int.Conf.on Artificial Intelligence (IC-AI), Las Vegas, USA*, pages 650–653. CSREA Press, 2003.

[7] J Euzenat. Eight questions about Semantic Web annotations. *IEEE Intelligent Systems*, 17(2):55–62, 2002.

[8] D Fensel et al. OIL: An ontology infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16:38–45, 2001.

[9] J Fernandez and R Martinez. A cooperative tool for facilitating knowledge management. *Expert Systems with Applications*, 18:315–330, 2000.

[10] A Gómez and Oscar Corcho. Ontology languages for the Semantic Web. *IEEE Intelligent Systems*, 2002.

[11] A Gómez et al. Knowledge maps: An essential technique for conceptualisation. *Data & Knowledge Engineering*, 33:169–190, 2000.

[12] J Gordon. Creating knowledge maps by exploiting dependent relationships. *Knowledge-Based Systems*, pages 71–79, 2000.

[13] N Guarino et al. OntoSeek: Content-based access to the web. *IEEE Intelligent Systems*, pages 70–80, 1999.

[14] V Guerrero et al. Document organization using Kohonen's algorithm. *Information Processing and Management*, 38:79–89, 2002.

[15] J Heflin et al. Applying ontology to the web: A case study. *Engineering Applications of Bio-Inspired Artificial Neural Networks*, 1607, 1999.

[16] J Heflin and J Hendler. Dynamic ontologies on the web. In *American Association For Artificial Intelligence Conf.*, pages 251–254. AAAI Press, California, 2000.

[17] I Horrocks et al. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of web semantics*, 1(1):7–26, 2003.

[18] A Jain et al. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[19] R Johnson and D Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, New Jersey, 4th edition, 1998.

[20] S Kaski et al. WEBSOM - Self-organizing maps of document collections. *Neurocomputing*, 6:101–117, 1998.

[21] T Kohonen. *Self-Organizing Maps*. Information Sciences Series. Springer-Verlag, Berlin, 3rd edition, 2001.

[22] T Kohonen et al. Self organization of a massive text document collection. In E Oja and S Kaski, editors, *Kohonen Maps*, pages 171–182. Elsevier Sci, Amsterdam, 1999.

[23] S Laakso et al. Self-organizing maps of Web link information. In N Allinson et al., editors, *Advances in Self-Organizing Maps*, pages 146–151. Springer-Verlag, Berlin, 2001.

[24] J Laaksonen et al. PicSOM - content-based image retrieval with self-organizing maps. *Pattern recognition letters*, 21:1199–1027, 2000.

[25] L Lacy. *OWL: Representing Information Using the Web Ontology Language*. Trafford Publishing, USA, 2005.

[26] K Lagus. Text retrieval using self-organized document maps. *Neural Processing Letters*, 15:21–29, 2002.

[27] CS Lee et al. Automated ontology construction for unstructured text documents. *Data & Knowledge Engineering*, 60(3):547–566, 2007.

[28] S Li. A web-aware interoperable data mining system. *Expert systems with applications*, 22:132–146, 2002.

[29] A Maedche and S Staab. Ontology learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.

[30] P Martin and P Eklund. Embedding knowledge in web documents. *Computer Networks*, 31:1403–1419, 1999.

[31] J McCormack and B Wohlschlaeger. Harnessing agent technologies for data mining and knowledge discovery. In *Data Mining and Knowledge Discovery: Theory, Tools and Technology II*, volume 4057, pages 393–400, 2000.

[32] D Merkl. Text classification with self-organizing maps: Some lessons learned. *Neurocomputing*, 21:61–77, 1998.

[33] E Motta et al. Ontology-driven document enrichment: principles, tools and applications. *Int.J.Human-Computer Studies*, 52:1071–1109, 2000.

[34] J Principe. *Neural and Adaptive Systems, Fundamentals through Simulations*, chapter 7. Wiley, USA, 2000.

[35] D Pullwitt and R Der. Integrating contextual information into text document clustering with self-organizing maps. In N Allinson et al., editors, *Advances in Self-Organizing Maps*, pages 54–60. Springer-Verlag, Berlin, 2001.

[36] A Rauber and D Merkl. Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets. In *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 228–237, 1999.

[37] B Ripley. *Pattern Recognition and Neural Networks*, chapter 1,9. University Press, Cambridge, 1996.

[38] H Ritter and T Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989.

[39] G Tambouratzis et al. Evaluating SOM-based model in text classification tasks for the greek language. In N Allinson et al., editors, *Advances in Self-Organizing Maps*, pages 267–274. Springer-Verlag, Berlin, 2001.

[40] H Tanaka et al. An efficient document clustering algorithm and its application to a document browser. *Information Processing and Management*, 35:541–557, 1999.

[41] *JRG Pulido* et al. Identifying ontology components from digital archives for the semantic web. In *IASTED Advances in Computer Science and Technology (ACST)*, 2006. CD edition.

[42] *JRG Pulido* et al. Ontology languages for the semantic web: A never completely updated review. *Knowledge-Based Systems*, Elsevier volume 19, issue 7:489–497, 2006.

[43] *JRG Pulido* et al. Semi-automatic derivation of specific-domain ontologies for the semantic web. In Gelbukh and Reyes-Garca, editors, *5th Mexican Intl Conf on Artificial Intelligence*, pages 253–261. IEEE Computer Soc.Press, Los Alamitos, 2006.

[44] P Velardi et al. A taxonomy learning method and its application to characterize a scientific web community. *IEEE Trans.on Knowledge And Data Engineering*, 19(2):180–191, 2007.

[45] S Wang and H Wang. Knowledge discovery through self-organizing maps: Data visualization and query processing. *Knowledge and Information systems*, 4:31–45, 2002.

[46] A Waterson and A Preece. Verifying ontological commitment in knowledge-based systems. *Knowledge-Based Systems*, 12:45–54, 1999.

[47] J Xu et al. Mobile location estimation for ds-cdma systems using self-organizing maps. *Wireless Communications & Mobile Computing*, 7(3):285–298, 2007.

[48] LD Yin et al. Clustering of gene expression data: performance and similarity analysis. *BMC Bioinformatics*, 7(4), 2006.