The original publication is available at www.springerlink.com.
Albertoni R., Camossi E., De Martino M., Giannini F, Monti M, **Context Enabled Semantic Granularity**, Advanced Knowledge-Based Systems, 12th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Zagreb, Croatia, 2008

Book Series Lecture Notes in Computer Science
Publisher Springer Berlin / Heidelberg
ISSN 0302-9743 (Print) 1611-3349 (Online)
Volume Volume 5178/2008
Book Knowledge-Based Intelligent Information and Engineering Systems
DOI 10.1007/978-3-540-85565-1
ISBN 978-3-540-85564-4
DOI 10.1007/978-3-540-85565-1_84
Pages 682-688

Context Enabled Semantic Granularity

Riccardo Albertoni[1], Elena Camossi[2*], Monica De Martino[1], Franca Giannini[1], and Marina Monti[1]

[1] IMATI, Consiglio Nazionale delle Ricerche, Via De Marini, 6 - Torre di Francia - 16149 Genova, Italy
{albertoni,demartino,giannini,monti}@ge.imati.cnr.it
[2] School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland
elena.camossi@ucd.ie

Abstract. In this paper we propose a powerful ontology driven method that eases the browsing of any repository of information resources described by an ontology: we provide a flexible semantic granularity method for the navigation of a repository according to different levels of abstraction, i.e. granularities. The granularity is explicitly parameterised according to the criteria induced by the context.

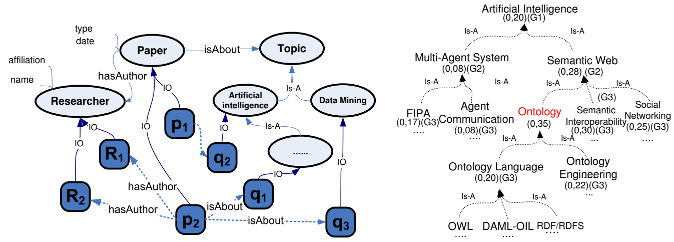
1 Introduction

Semantic granularity enables the browsing of information resources according to different levels of abstraction, i.e., granularities. Granularities have been already studied in the area of Information Systems, in particular for the spatio-temporal domain [1]. Some attempts to define semantic granularities have been made with respect to terminologies. However, in both cases, granularities are static and embedded in the data model or in the database schema. By contrast, semantic granularity [2] extracts dynamically the structure, namely the *granularity lattice*, which enables to organize the repository at different levels of abstraction.

Moreover, to fill the gap between Cognitive and Information spaces [2], it is mandatory to take into account the influence of the *context*. Thus, in this paper we propose a context dependent semantic granularity method. It originates from the research results presented in [3], where the *application context* has been formalized in order to parameterize the semantic similarity among ontology instances. The application context models the importance of ontology entities (i.e., classes, attributes and relations) that concur in the granularity assessment as well as the different operations used to analyse them. Herein we adapt this formalization to parameterize the semantic granularity we have proposed in [2]. The resulting instrument is a powerful ontology driven method that eases the browsing of a repository of information resources.

The advances of this work with respect to our previous results [3, 2] are: (i) the layered framework becomes a potential common frame for context dependent ontology driven methods: we demonstrate it is suitable for both the semantic similarity and the granularity; (ii) we propose an extension of the application context formalism for the granularity: we define new operations and functions to be adopted for the analysis of ontology entities; (iii) we illustrate a more flexible evaluation of semantic granularity

* The work of Elena Camossi is supported by the Irish Research Council for Science, Engineering and Technology.



(a) (b)

Fig. 1. (a) An ontology schema to organize information about scientific papers. (b) Topic taxonomy and semantic granularity results. For each quality, the values in brackets indicate its abstraction capability and the granularity (G1,G2, or G3) to which it is assigned.

throughout its context dependent parameterization. Overall, the main benefit of this work is to enable a user-oriented browsing: the user may formulate, learn and modify the granularity criteria induced by the context.

The paper is organized as follows. In Section 2 we introduce the semantic granularity method discussing an illustrative example. In Section 3 we describe the context dependent parameterization of the method. Finally, Section 4 concludes the paper outlining future research directions.

2 Semantic Granularity

Semantic granularities are built with respect to an ontology $\mathcal{O}$ representing the information resources, which are described by a structured set of *qualities* $\mathcal{Q}$. Information resources are instances of a class $\mathcal{S}$. The set of qualities $\mathcal{Q}$ are represented by ontology classes organized in a hierarchy $\prec_Q$ induced by relations IS-A and Part-Whole. We suppose that a top hierarchy class Q^T exists such that, for each quality Q, $Q \prec_Q Q^T$; moreover, each $Q \in \mathcal{Q}$ has at least one direct instance.

The user is expected to access the resources in the repository by using set of granules with increasing detail. Each granule belongs to a given granularity and corresponds to a quality Q according to which the corresponding resources are grouped. Granularities are defined dynamically, according to both the data model, represented by the ontology schema, and the data, given by ontology instances.

The method follows a two-phase process. In the first phase, namely *quality filtering*, it evaluates each quality with respect to its capability of abstracting information resources. The evaluation of the *abstraction capability of a quality* Q takes into account the attributes and the relations that characterize the resources in $\mathcal{S}$ as well as the attributes and the relations of their related instances. The quality filtering returns the qualities with a better value of abstraction capability which are promoted to be granules of some granularity.

Then, the *granularity building* phase distributes the granules among different granularities according to $\prec_Q$. It returns the set of granularities to employ for the repository navigation. Since not all the qualities in the hierarchy will be evaluated as good abstractors by the quality filtering phase, the browsing of the information resources according to semantic granularities will differ from the browsing driven by IS-A and Part-Whole.

Example 1. Fig. 1 shows an example of application of semantic granularity onto a repository of scientific papers represented by the ontology schema in Fig. 1(a). We use instances of Paper *as* *resources and of* Topic *as qualities in input for the semantic granularity. Fig. 1(b) is an excerpt of the topic taxonomy: the values in brackets are the results of the semantic granularity application. The first value is the abstraction capability of the topic resulting by the quality filtering phase: the lower is the value, the better the topic abstracts its subtopics in the hierarchy. Setting an abstraction threshold (for instance 0,31), the quality/topic* Ontology *is discarded. The second element in the brackets represents the result of the granularity building phase: the granularities G1, G2, G3, which correspond to distinct levels of abstraction, are identified and the granules are associated with them. For example, increasing the level of detail,* Artificial Intelligence *belonging to G1 is converted in* Multi-Agent System *and* Semantic Web*, which belong to G2. Furthermore,* Semantic Web *is converted in* Ontology Language, Ontology Engineering, Semantic Interoperability, *and* Social Networking *belonging to G3.*

3 Context Dependent Parameterization of Semantic Granularity

3.1 The ontology model and the layered framework

The ontology model gives the expressiveness of the ontologies defined according to the framework. Herein, we adopt the ontology model equivalent to an ontology with data types and defined in [3]. In addiction to δ_a, δ_r, δ_c that retrieve the attributes, the relations and the concepts reachable by a given concept or relation, we defined the function $\delta_{r^{-1}}\colon C \cup R \rightarrow 2^R$, such that $\delta_{r^{-1}}(c)$=$\{r\colon R \mid \exists c' \in C, \sigma_R(r) = (c', c)\}$ denotes the set of relations that reach $c \in C$; and $\delta_{r^{-1}}(r)$=$\{r'\colon R \mid r' \neq r, \exists c \in C, \exists c' \in \delta_c(r), \sigma_R(r') = (c, c')\}$ is the set of relations which differ from r and reaches the concepts reachable through the relation $r \in R$.

The framework is structured in terms of *data*, *ontology* and *context* layers plus the *domain knowledge* layer which spans all the others [4].

The *data layer* provides the *functions* onto the data type values (e.g., functions which filter the values of simple or complex data types, statistical and user defined functions).

The *ontology layer* provides the mechanism for processing semantic granularity by considering the way ontology's entities are related. It provides the implementation of the semantic granularity and of the *operations* (e.g., intersection, count) which may be recalled by the semantic granularity in a given application context.

The *context layer* provides the *application contexts*, i.e., the criteria for the computation of semantic granularity considering how ontology entities are used for specific purposes. Each application context specifies the attributes and the relations to consider likewise the operations and functions to apply on them.

3.2 Application Contexts

This section formalizes the application contexts used to parameterize the semantic granularity. It is an extension of the formalization illustrated in [3]. An application context is defined by an ontology engineer, according to specific application needs. Assuming the definition of *Sequence of elements* presented in [3] a *path of recursion* tracks the recursion during the assessment of the semantic granularity and represents the navigation path in the ontology to collect the information of interest. It is defined as follows.

Definition 1 (Path of Recursion of length n) *A* path of recursion p *of length* n *is a sequence of elements with length* n *whose elements are classes in* C *and relations in* R *(i.e.,*

$p \in S^n_{C\cup R}$), such that p starts from a class c and whose other elements are relations either starting from or ending in c or c', where c' is a class involved in some relation in p, that is $p(1) \in C \wedge \forall j \in [2,n]\; p(j) \in R \wedge (p(j) \in \delta_r(p(j-1)) \vee p(j) \in \delta_{r^{-1}}(p(j-1)))$.

P^n denotes the set of all paths of recursion with length n, whereas P denotes the set of all paths of recursion $P = \bigcup_{n\in N} P^n$.

The *application context (AC)* function is defined inductively according to the length of the path of recursion. It yields the set of attributes and relations to consider and the operations to apply when computing the semantic granularities, e.g., sum, average, minimum, maximum, which could indirectly recall the functions in the data layer, and different forms of count operations: $Count$, which evaluates the cardinality of a set of instances; $WCount$, which evaluates a weighted count of instances according to the cardinality of related attributes or relations; $InvCount$, which evaluates the inverse cardinality of a set of instances, (i.e., a set with less instances has more importance than a set with greater cardinality). The application context is formally defined as follows.

Definition 2 (Application Context AC) *Given the set P of paths of recursion, L the set of operations provided by the ontology layer (i.e. Count, WCount and InvCount for the semantic granularity), G the set of datatype functions available in the data layer, the application context for the semantic granularity is defined by the partial function AC: $P \rightarrow 2^{A\times(L\cup G)} \times 2^{R\times L}$.*

Note that each application context AC is characterized by the operators AC_A: $P \rightarrow 2^{A\times(L\cup G)}$ and AC_R: $P \rightarrow 2^{R\times L}$, which yield respectively the context AC related to the attributes and to the relations.

Example 2. Given the ontology schema in Fig. 1, two examples of application contexts AC_1 and AC_2 are defined. AC_1 corresponds to the hard coded context implicitly used in Example 1. It starts from the path of recursion $[Topic]$ and considers the instances of Paper *associated with each* Topic *to calculate the capability of abstraction. It is formalised as follows:*

$[Topic] \overset{AC_1}{\rightarrow} \{\{\phi\},\{(isAbout^{-1},Count)\}\}$.

AC_2 considers the date of publication, the number of authors, the type (i.e., journal, conference proceedings, or book) of papers. It is formalised as follows:

$[Topic] \overset{AC_2}{\rightarrow} \{\{\phi\},\{(isAbout^{-1},WCount)\}\}$

$[Topic.isAbout^{-1}] \overset{AC_2}{\rightarrow} \{\{(type,i(Paper,Book))(date,g(today))\}$
$\{(hasAuthor,InvCount)\}\}$.

AC_2 starts from the path of recursion $[Topic]$ and moves along to the inverse of relation isAbout *to focus on the attributes and relations of* Paper*. The change of focus is tracked by the path of recursion $[Topic.isAbout^{-1}]$. AC_2, when applied to the new path of recursion, returns the attributes* type *and* date *and the relation* hasAuthor. Type *and* date *are respectively processed by the two functions i and g provided by the data layer: $i(Paper, Book)$ returns the inverse of the cardinality of the papers associated with a given topic that are published in a book or a journal, whereas $g(today)$ counts only the papers published in the last three years. Finally, the inverse cardinality of the relation* hasAuthor *is considered.*

3.3 Context Dependent Quality Filtering

As mentioned above, the *quality filtering* evaluates the abstraction capability of each quality Q, selecting those more representative for the repository that will become granules. We explicitly parameterize the capability of abstraction of a quality Q to provide a semantic granularity according to an application context AC.

Definition 3 (Abstraction capability of a quality Q w.r.t. AC, $R_Q^{AC,p}$) *Given an ontology $\mathcal{O}$, representing the class of information resources $\mathcal{S}$, described with respect to the set of qualities $\mathcal{Q}$; the quality $Q \in \mathcal{Q}$; the partial order $\prec_Q$ on $\mathcal{Q}$; Q^T the most generic quality in $\mathcal{Q}$ according to $\prec_Q$; the starting path of recursion p initialized as $p = [Q^T]$; $R_Q^{AC,p}$, is defined as follows:*

$$R_Q^{AC,p} = \frac{\sum_{x \in AC_A(p) \cup AC_R(p)} R_Q^{AC,p,x}}{|AC_A(p)| + |AC_R(p)|}.$$

$R_Q^{AC,p,x}$ is the abstraction capability of Q according to the relation or attribute x mentioned in the application context AC for a path of recursion p. It is defined as follows:

$$R_Q^{AC,p,x} = \begin{cases} R_Q^{AC,p \circ x} & \text{if (x,} WCount) \in AC_R(p) \\ \frac{\max_{\{Q' \mid Q' \prec_Q Q\}} s_{AC,p,x}^{Q'*}}{\sum_{\{Q' \mid Q' \prec_Q Q\}} s_{AC,p,x}^{Q'*} + s_{AC,p,x}^{Q}} & \text{otherwise.} \end{cases}$$

In practice, for each relation x in the context whose associated operation is $WCount$, the evaluation of $R_Q^{AC,p}$ is defined recursively considering the instances related to the quality Q by the path of recursion $p \circ x$. That allows to assess the abstraction capability of Q also considering the relations and attributes belonging to instances that are not directly related to the quality Q. Otherwise, when the context does not prescribe a recursive assessment, the abstraction capability presented in [2] is parameterized according to the context defining $s_{AC,p,x}^{Q}$ and $s_{AC,p,x}^{Q^*}$ as follows:

$$s_{AC,p,x}^{Q} = \sum_{q \in Q} \sum_{\iota \in I(q,p)} f_{AC}^{p,x}(\iota) \qquad s_{AC,p,x}^{Q^*} = \sum_{\{Q' \mid Q' \prec_Q Q\}} s_{AC,x}^{Q'}.$$

$f_{AC}^{p,x}(\iota)$ measures the *weight* of the instance q of Q according to the relation or attribute x belonging to the set of instances $I(q,p)$, which are reachable through the recursion path p, and considering the operations indicated in AC. Thus, assuming: (i) X a placeholder that works as a metasymbol that may be replaced by R or A, whether x is respectively a relation or an attribute; (ii) $i_A(\iota, a) = \{v \in V \mid (\iota, v) \in l_A(a), \exists y \in C \ s.t. \ \sigma_A(a) = (y, T) \wedge l_T(T) = 2^V\}$ the set of values assumed by the instance ι for attribute a; (iii) $i_R(\iota, r) = \{\iota' \in l_c(c') \mid \exists c \ \iota \in l_c(c) \ \exists \ c' \ s.t. \ \sigma_R(r) \in (c, c') \wedge (\iota, \iota') \in l_R(r)\}$ the set of instances related to the instance ι by relation r; (iiii) g a function provided by the data layer, w the metasymbol that works as placeholder for the function parameters that have been already fixed in the application contexts; $f_{AC}^{p,x}(\iota)$ is defined as follows:

$$f_{AC}^{p,x}(\iota) = \begin{cases} g(w) & \text{if} (x, g(w)) \in AC_A(p),\ v \in i_A(\iota, x) \\ |i_X(\iota, x)| & \text{if } (x, \text{Count}) \in AC_X(p) \\ \frac{1}{|i_X(\iota, x)| + 1} & \text{if } (x, InvCount) \in AC_X(p) \end{cases}$$

The following example gives the flavour of circumstances where different contexts arise.

Example 3. We provide an example of semantic granularity application according to two application contexts. Let us considering a user who needs to browse a repository of scientific papers with two distinct purposes: (1) to get a first impression about the repository content and (2) to identify the hottest topics in the Computer Science research. These two aims correspond to two distinct contexts formalized respectively by the functions AC_1 and AC_2 given in the Example 2. We have extracted some real data from Faceted Dblp, a repository of Computer Science papers[3]*,*

[3] Faced Dblp is available at http://dblp.l3s.de/.

and organized them in the ontology of Fig. 1. The semantic granularity has been applied and a fragment of the result is illustrated in Fig. 2. Different granularities are obtained considering the contexts AC_1 and AC_2: comparing Fig. 2(a) and Fig. 2(b) granularities G2 and G3 contain different granules, i.e., topics. Indeed, the filtering phase results in different abstraction capabilities for the topics, according the two contexts. For example, Semantic Web *disappears in Fig. 2(b), as it has been discarded by the filtering, whereas* Ontology *increases its importance: moving from AC_1 to AC_2,* Semantic Web *decreases its abstraction capability as R_Q increases from 0.28 to 0.35, whereas* Ontology *increases its importance as R_Q decreases from 0.35 to 0.31. Adopting the threshold 0.31, the granularity building phase returns the granularities depicted in Fig 2.*

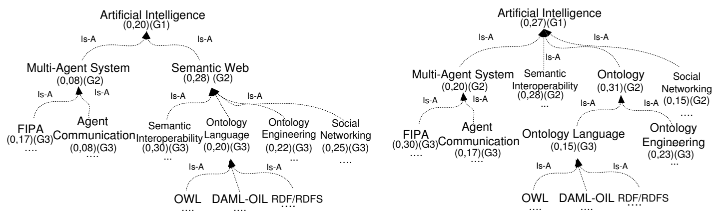


Fig. 2. Results of semantic granularity: according to contexts AC_1 (a) and AC_2 (b).

4 Conclusions and Future Works

In this paper we have described a context dependent parameterization of semantic granularity to browse any kind of information source described with respect to a set of qualities represented in ontologies. Even if the work is at a preliminary stage, the intermediate results indicate the validity of the undertaken approach towards the definition of a powerful ontology driven method allowing a user-oriented formulation of the granularity criteria induced by the context.
A more rigorous test case is in progress. So far, the context as explicitly parameterization of ontology driven methods has been demonstrated to be essential both for the semantic similarity [3] and the granularity.

References

1. E. Camossi, M. Bertolotto, E. Bertino: A Multigranular Object-oriented Framework Supporting Spatio-temporal Granularity Conversions. In *International Journal of Geographical Information Science*, 20(5):511-534, 2006.
2. R. Albertoni, E. Camossi, M. De Martino, F. Giannini, M. Monti: Semantic Granularity for the Semantic Web. In Proc. of the *2nd Int'l IFIP Workshop On Semantic Web & Web Semantics*, 4278:1863-1872, 2006.
3. R. Albertoni, M. De Martino: Asymmetric and Context-Dependent Semantic Similarity among Ontology Instances. In *Journal on Data Semantics* X, LNCS 4900:1-30, 2008.
4. M. Ehrig, P. Haase, M. Hefke, N. Stojanovic: Similarity for Ontologies - A Comprehensive Framework. In Proc. of the *13th European Conf. on Information Systems, Information Systems in a Rapidly Changing Economy*, 2005.