A Data Referencing Formalism for Information Exchange between Deafblind People and Databases

Carlos Seror
Independent researcher
Valencia, Spain
email: serorcarlos@gmail.com

***Abstract*—A categorizing feature of human language could be usefully exploited to facilitate an interactive access of deafblind people to database information. To that end, an interpretation of databases in terms of categories and instances is proposed, as well as the basic elements of a syntax consistent with it.**

Keywords-databases; categories; natural language; syntax; data referencing; deafblind people

I. INTRODUCTION

The access of DeafBlind (DB) people to information is generally limited to communication with other human beings through various, often non-standard, languages. Therefore, they require a human intermediary to query a database, or to be able to understand database reports expressed in some readable string form. Attempts to facilitate such communication have been described [1][2], including based on ontologies [16], sign languages [10], the Semantic Web [17], and even neural networks [9]. However, the formal aspect of data referencing has not been addressed in the literature in great depth. In this paper, based on a categorizing feature of natural languages [11], a formalism will be proposed that should make it possible to establish syntax correspondences between natural language expressions and databases.

In Section II, the concept of data aggregate will be introduced, to be subsequently endowed, in Section III, with a simple structure based on the logical connectives $\wedge$, $\vee$. In Section IV, a string syntax will be derived from the resulting expressions and shown to be consistent with both natural language and databases. In Section V, the scope of the connectives will be enlarged to include categories of data items, and the resulting n-tuple structure will be seen to be, in Section VI, a particular case of more complex two-dimensional structures. Based on such general structures, a referencing and updating language will be developed in Section VII, and used in Section VIII to establish equivalences with simple database operations, including a few examples showing the structures' potential for semantic representation. Finally, Section IX will assess the potential of the proposed approach for communication with deafblind users.

II. DATA AGGREGATES

Databases have been around for a number of decades now. They are just a particular way to organize data [12]. From an abstract standpoint, databases could be described as symbol aggregates endowed with a particular structure, usually in the form of tables, but sometimes as graphs, objects or other ways of organization [6]. Natural Language (NL), being a means to deliver information, could also be argued to use data but, except in specific, explicitly structured subject areas, its users are usually unaware of the structure of such data. Describing a data structure consistent with NLs would therefore be a first step to establish a working correspondence between NLs and databases. This paper formally describes one such structure, based on a categorizing feature of NLs.

For a general approach to a diversity of data structures, the term 'data aggregate' will be adopted here. A data aggregate is defined as a number of data items that could be represented as points on a surface. This is arguably the minimal structure that can be conceived of, and it does not exclude other additional structures. Thus, a number of colors could be considered as a data aggregate, irrespective of whether they are located within a rainbow or on a painter's palette. In a data aggregate, items can be pointed to but do not have to be distinctly labelled —you may know nearly everything about a wood and not have a name for any of its trees. An item in a data aggregate could also be identified through a number of instructions on how to locate it. Also, a data aggregate could be indefinitely updated, as long as any new item could be represented as an additional point on the same surface. Goats in a herd, or data in a form, are simple examples of data aggregates.

III. INTENSION AND EXTENSION IN A DATA AGGREGATE

Items in a data aggregate could be discriminated by applying criteria to them. A criterion is a notion more general than a property, because it encompasses properties as well as fancy choices and algorithms. Two of the simplest criteria that could be applied to a data aggregate are the ones associated to intension and extension. Aggregations of items in a data aggregate do not have any extension or intension connotation *per se*, and are therefore objects more general than sets. Extension and intension could be implemented on them by means of resp. the connectives $\wedge$, $\vee$, e.g.,

blue $\wedge$ red $\wedge$ yellow $\wedge$ ...	extension [colors]
blue $\vee$ red $\vee$ yellow $\vee$ ...	intension [color]

 ISBN: 978-1-61208-738-2

In general, therefore, a number of items u_1, u_2, u_3, ... in a data aggregate could be discriminated in two alternate ways, i.e.,

$$u_1 \wedge u_2 \wedge u_3 \wedge ... \quad (1)$$
$$u_1 \vee u_2 \vee u_3 \vee ... \quad (2)$$

Because mathematical sets are said to be describable both in extensional and intensional terms, we shall rather not mix things up and separately discern either description instead. Hence, any expression in the form (1) will be referred to as a **combination**, while any expression in the form (2) will be referred to as a **category**. This difference reflects the use of plural resp. singular in human language. Thus, 'colors' could be associated to a combination, while 'color' could be associated to a category. When the scope of the criteria is not specified, the expressions (1) and (2) could also be interpreted as reflecting the difference between resp. 'every' and 'any'.

Any item *u* encompassed by a category *C* —i.e., complying with the criteria that define *C*— will be referred to as an **instance** of *C*. A category *C* encompassing the instances u_1, u_2, u_3, ... will therefore be expressed as

$$C \equiv u_1 \vee u_2 \vee u_3 \vee ...$$

The definitions of category and instance could be used as a means to locally refer to an item in a data aggregate. Indeed, in a data aggregate *E* where a category *C* has been identified, any instance of *C* could be expressed as a disambiguation of *C*. That is, if we denote a category as *C()* and an instance *u* of that category as *C(u)*, we could refer to *u* as

$$C() \rightarrow C(u)$$

The expression above may be interpreted as a path in *E*, i.e., "select *C*, then select the instance *u* of *C*", where *u* could be identified by means of either a label or a number of instructions. In the following sections, an enlarged notion of disambiguation will be shown to be a powerful device to refer to data items in a data aggregate —arguably, the basic addressing device used by natural language users.

A data item in a data aggregate could also be referred to through a disambiguation of categories. For example, the word 'green' may denote either a color or a political affiliation. In the absence of additional cues, they could be disambiguated resp. as either *color(green)* or *political_affiliation(green)*. In formal terms, if *H* is a category having the category *C* as an instance, then the instance *C(u)* could be referred to as

$$H(u) \rightarrow C(u)$$

IV. SYMBOL SYNTAX AND STRING SYNTAX

The notation used thus far, based on symbols such as connectives or arrows, will be referred to as **symbol syntax**. An alternative syntax, which shall be referred to as **string syntax**, will express categories and instances as single words, and disambiguations as strings, as follows:

symbol syntax	string syntax
C()	C
$C() \rightarrow C(u)$	C [δ u]

where the symbol δ denotes the fact that *u* complies with the criterion that defines *C*. The correspondence between symbol expressions and string expressions will be denoted as >>, e.g.,

color() >> color
color() → color(blue) >> color [δ blue]

Note that, in practice, if we deem it obvious that, e.g., the word 'blue' refers to a color, we will not precede it with the word 'color', which will have to be guessed by the receiver. This data compression feature hinges on an implicit operation that pervades human language —and arguably also human thought—, i.e., spontaneous categorization.

V. COMBINED CATEGORIES

The connective ∧ could also be used to discriminate combinations of categories in a data aggregate. For example, from the categories

mass, electric charge, spin

a combined category could be derived, which in turn would give rise to a number of objects, e.g.,

mass ∧ electric_charge ∧ spin >> particle
mass ∨ electric charge ∨ spin >> observable
$\text{mass}(9.1\times10^{-31}\ \text{kg}) \wedge \text{electric\_charge}(-1.6\times10^{-19}\ \text{C}) \wedge \text{spin}(1/2)$ >>
particle [δ electron]

As the latter example shows, a combination of categories is itself a category, having as instances combinations of instances of its component categories. The latter example is a full disambiguation of the category 'particle'. However, combined categories could also be partially disambiguated by specifying just some instances of its component categories, e.g.,

bearing() ∧ altitude(7000 ft) ∧ No._of_passengers(80)

Component categories in a combined category could also be discriminated by means of the connective ∨. The resulting object could be used to identify a path within the data aggregate leading to any of such component categories. For example, the category 'color' could also be construed as an attribute, i.e., an instance of the category

color ∨ shape ∨ size ∨ ...

 ISBN: 978-1-61208-738-2

Categories pervade language, a fact which is obscured by the spontaneous categorization mechanism, of which language users are usually unaware. In human languages, spontaneous categorization is a run-of-the-mill feature, associated not only to adjectives such as 'blue' or 'big', but to virtually any kind of meaningful component. Thus, the sentence 'birds fly' could be inaccurately categorized as implying that hens fly, or meaningfully categorized by instead interpreting 'birds' as an instance of some category, e.g., birds ∨ mosquitos ∨ bats ∨ ... having 'fly' as an attribute. Similarly, the meaning of 'a through person' could only be captured by evoking a category of concepts having 'through' as an instance.

In general, therefore, a **combined category** G^{η} will be defined as the general expression

$$G^{\eta} = O1 \wedge O2 \wedge O3 \wedge ... \quad (3)$$

where *O1*, *O2*, *O3*, ... are categories, whether they have been disambiguated or not, and η uniquely identifies that particular combination of categories. The definition (3) is consistent with a number of concept theories [13][14], that describe concepts as n-tuples of symbols representing attributes.

VI. CATEGORY CLUSTERS

An n-tuple is just a one-dimensional combination of categories, and therefore a particular case of the more general concept of **category cluster**, where complex spatial relations could be incorporated as additional discrimination criteria in a data aggregate. A data form is a familiar example of data cluster. In general terms, therefore, a **representation** can now be defined as a data aggregate together with any number of category clusters. Now, given a category cluster *G* and one of its component categories *C*, any set of instructions *r* to uniquely identify *C* within *G* will be referred to as a **relation** *r(G, C)*.

As in the one-dimensional case, specific category clusters could also be referred to by specifying one or more of its component categories. For example, an employee's record might be uniquely identified by specifying just the employee's name, or his age and height. This will be formally described as follows. Let *G* be a category cluster, *r* a set of instructions to identify *C* within *G*, and G_k a copy of *G* where the data item *u* has been specified for the category *C*. The cluster G_k could therefore be referred to as

$$G_k = G \mid r(G, C(u)) \quad (4)$$

i.e., G_k can be interpreted as a partial disambiguation of *G*. In string syntax, this will be expressed as

G | r(G, C(u)) >> G [r u]

For example,

ball = shape(round) ∧ color() ∧ size()

$ball_k$ = shape(round) ∧ color(red) ∧ size(big) >> ball [r_2 red] [r_3 big]

where r_1, r_2, r_3 would represent resp. the sets of instructions to locally identify each of the component categories 'shape', 'color', 'size'. In the general case, the disambiguation of a category cluster *G* will be expressed in string syntax as

$$G\ \Sigma[r_j\ u_j] \quad (5)$$

where Σ denotes a string made up of $[r_j\ u_j]$ pairs, u_j denotes an instance of the component category C_j, and r_j denotes the relation $r_j(G, C_j)$. The possibility to uniquely identify a category cluster even when only some of its component categories have been specified is a feature heavily used by natural language users as a data compression device. Indeed, if there is only one red ball in the room, you would hardly want to refer to it as "the big red expensive air-filled ball on the sofa".

Any set of rules to convert string syntax expressions into different strings will be referred to as a **conventional** syntax. For example,

String syntax	*Conventional syntax*	
ball [r_2 red] [r_3 big]	big red ball	(English)
boule [r_2 rouge]	boule rouge	(French)
bam [r ug]	bugam	(imaginary)
✌ [r_2 👌] [r_3 🖐]	👌✌〰🖐	(non-word)

The notion of category cluster, plus the relations it entails, endow data aggregates with powerful structures that could be used to semantically represent a vast number of concepts, e.g., ontologies, verbs, or semantic representations of space-time concepts, together with a local mechanism to refer to them [8][11]. An example of semantic cluster is described in Section VIII, *A.*

VII. REFERENCE AND UPDATING

The definition of the cluster-category relation implies that categories could also be referred to in terms of the cluster or clusters they are part of. This stems from the definition of the converse relation. Given a relation *r(G, C)*, the expression

$$C \rightarrow C \mid r(G, C) \quad (6)$$

describes the constriction of the general category C to the range of instances allowed in G. In string syntax, (6) will be expressed as

C [r' G]

and r' will be referred to as the **converse relation** of r. Example:

color | r_2(ball, color) >> color [r'$_2$ ball] (7)

If we now incorporate (5) into the definition (6), the general expression will be

$$C\ [r'\ G\ \Sigma[r_j\ u_j]] \quad (8)$$

When G $\Sigma[r_j\ u_j]$ describes a full disambiguation, the expression (8) will point to the unique content of *C* in *G*, i.e., it could be used to indirectly refer to a specific component instance in a category cluster.

The expressions (5) and (8), used to refer to resp. category clusters and component categories, could therefore be used by a sender to update the receiver's presumed representation. For example, if the receiver is believed to ignore that there is a ball on the sofa, then that information could be sent by means of (5), i.e.,

!ball [r_4 sofa]

where the symbol ***!*** denotes a new category cluster to be included in the receiver's representation. Such updating is a commonplace device used in natural language exchanges, to indicate, e.g., that a new character has appeared in a film, or a new guest has arrived at a party. If the receiver were presumed to know that there is a ball on the sofa but not its size, then that fact could be conveyed by means of the expression

ball [r_4 sofa] [r_3 big!]

where ***!*** now denotes an instance intended to be included in a category presumed to be empty at the receiver. In a converse situation, where the sender ignores some information item supposedly known by the receiver, the expressions (5) and (8) could also be used for querying purposes, by pointing to the required item by means of a different symbol, e.g.,

? [r_4 sofa]
? size [r'_3 ball [r_4 sofa]]

where the symbol ***?*** points to resp. an category or instance unknown by the sender. The referencing and updating uses of (5) and (8) could be summed up as follows

reference	G [r u], C [r' G]	(9)
updating	!G [r u], G [r !u]	(10)
querying	? [r u], ? C [r' G]	(11)

VIII. DATABASES AS CATEGORY CLUSTERS

The expressions (10) and (11) provide a means to resp. update and query a communicating party, provided that the latter's representation is consistent with the sender's. Therefore, whatever the spatial configuration of a database and the language used by it to refer to its items, string syntax communication will be possible if the database's content could be interpreted in terms of categories and category clusters. In order to explicitly formalize how that could be done, we shall use two different conceptual frames for the same data, i.e., a database D, configured in the form of tables, and an associated representation R, configured in terms of category clusters.

In a database *D*, an empty table T_θ consisting of the columns C_1, C_2, ... could be interpreted as a combination of the associated categories C_1, C_2, ..., and any instantiation of that combined category would describe a row (or potential row) in T_θ. For example, let the table T_k consist of the columns 'name', 'age', and 'address'. Because each of these can take *any* value within its respective scope, they can also be construed as categories, i.e.,

$$\text{name}() \wedge \text{age}() \wedge \text{address}() \ >>\ G_\theta$$

where G_θ would be a category cluster associated to T_θ. By specifying values for those columns, a number of rows would be obtained, e.g.,

$$\text{name(Oz)} \wedge \text{age(39)} \wedge \text{address(7th Av.)} >> \text{row}_1$$
$$\text{name(John)} \wedge \text{age(54)} \wedge \text{address(97 St.)} >> \text{row}_2$$
$$\text{name}() \wedge \text{age(33)} \wedge \text{address(221B St.)} >> \text{row}_3$$

Now, if we use the connective $\vee$ to link the rows above, then the table T_θ could be interpreted as a category ambiguously referring to any of its rows. If we use the connective $\wedge$ instead, then T_θ could be interpreted as a combination of rows, i.e.,

$$\text{row}_1 \vee \text{row}_2 \vee \text{row}_3 \vee \ldots >> \text{employee}$$
$$\text{row}_1 \wedge \text{row}_2 \wedge \text{row}_3 \wedge \ldots >> \text{employees}$$

In string syntax, both rows and cells within a row could be referred to by means of (5) and (8), e.g.,

employee [r_2 age(33)]
age [r'_2 employee [r_3 221B St.]]

where the relations r_2, r'_2, r_3 would be defined according to (4) and (6). In the general case, communication between a database and a user could be established in either direction as follows:

A. User to database

By reversing the rules used to derive conventional syntaxes, messages sent by users to a database *D* for updating or querying purposes could be expressed in string syntax by means of resp. (10) or (11). Such messages could be processed at *D* insofar as its tables could be interpreted as category clusters and those clusters would be consistent with the sender's. When that is the case, updating and querying could be interpreted in *D* as follows

String syntax	Database operation	
G [r_m !u]	$N_1(u_1) \wedge \ldots \wedge N_m(u \leftarrow)$	(12a)
!G [r_m u]	$N_1(u_1) \wedge \ldots \wedge N_m(u) \leftarrow N_m(u)$	(12b)
?G [r_m u]	$N_m(u) \rightarrow N_1(u_1) \wedge \ldots \wedge N_m(u)$	(12c)
?C_m [r'_m G]	$N_1(u_1) \wedge \ldots \wedge N_m(\rightarrow u)$	(12d)

where the symbols ← and → denote resp. the incorporation of a new item and the identification of an extant item. The updating operation would add resp. a value or a row to D, while the query would prompt D to identify resp. an item or a table, and then send the result to the querying party. Therefore, to be able to process string syntax expressions, a database should be configured so that either (a) the column headers in its tables reflect categories potentially referred to by the user, or (b) a sub-table could be identified in D for each category cluster that might be referred to by a user.

This is not uncommon. Meteorological and geolocation databases usually record data expected to be of interest for the general user, and databases containing spatial/temporal data most often lend themselves to semantic interpretation. As an example, let us define the category cluster G as follows:

...	h_1	h_1	H	h_2	...
...	T	t_1	T	t_2	...

which could be interpreted as describing a stay at the location h_1 for an indefinite time T until the time t_1, then some movement along some distance H during an indefinite time span T, and then the presence on a fixed location h_2 at the time t_2. From that cluster, the subclusters

h_1	H
t_1	T

H	h_2
T	t_2

could be denoted resp. as G_{depart}, G_{arrive}, implying the relations

$\text{from}(G_{depart}, \text{loc})$
$\text{at}(G_{arrive}, \text{time})$

The above relations could be used to construct a number of useful string syntax expressions, e.g.,

G_{depart} [from Rome]
G_{arrive} [at 09:23]

and therefore also updating and querying expressions, e.g.,

?time [r'$_2$ G_{arrive} [to Rome]]

For a database to be able to interpret such expressions, the adjacency relations in the sub-table

origin	destination	departure	arrival
Bonn	Rome	20:15	22:30

should be reconfigured so as to reflect the semantic relations in G, e.g.,

[origin]	H	[destination]
[departure]	T	[arrival]

so that, e.g., the sub-table

H	h_2
T	t_2

could be associated to the category cluster

$G_{arrive}(\text{loc}, \text{time})$

The reconfigured table in D is actually a three-dimensional table, where the original columns are now arranged differently, i.e., only the topology of the table has been changed.

B. Database to user

The correspondences (12a-b) could reciprocally be used by D to derive reports expressed in string syntax, i.e.,

Database operation	String syntax
$N_1(u_1) \wedge ... \wedge N_m(\rightarrow u)$	G [r_m !u]
$N_m(u) \rightarrow N_1(u_1) \wedge ... \wedge N_m(u)$	!G [r_m u]

that would prompt the receiver to update her representation in response to the query previously sent, or by, e.g., a geolocation algorithm intended to keep a user updated about his surroundings. An example would hopefully illustrate the reporting process. In a meteorological database M, the column headers 'temp', 'humidity', 'loc', and 'time' could be associated a combined category that a user would interpret as a number of variables describing different weather states, i.e.,

Column headers	Combined category
temp humidity loc time	temp $\wedge$ humidity $\wedge$ loc $\wedge$ time

A query intended to find out, e.g., the temperature in Paris at 22:05 would be expressed in string syntax as

?temp [r_1 Paris] [r_2 22:05]

In response to that query, the database would locate the row R having 'Paris' under the header 'loc' and '22:05' under the header 'time'. It would then retrieve from that row the cell under the header 'temp', and express the resulting value in string syntax as

R [r_3 !33ºC] [r_1 Paris] [r_2 22:05] (13)

If we use English words for the subindices, then we can write

r_1	r_{in}
R	$R_{a\_row\_in\_this\_database}$
r_2	r_{at}

A few translation rules, together with (8), would convert (13) into the conventional syntax expression

> the temperature from a row in this database in Paris at 22:05 is 33ºC

However, the receivers need not even know that the data has been retrieved from some table in the source database. They have chosen to ask the source because they trust it to output reliable data. Therefore, the source might safely decide to just translate

> the temperature in Paris at 22:05 is 33ºC

This omission might seem like a trick shrewdly devised to get the desired result. On the contrary, it is an information compression device routinely used by natural language speakers. Consider just a few examples.

- the kitchen [of our house] is in the ground floor
- I can see the airport [of Beijing] now
- the book [you expressed an interest to buy three minutes ago] is *Finnegan's Wake*

IX. Communication with Deafblind Users

By applying or reversing the rules that define a conventional syntax (cf. Section IV), communications with a database could be established in any conventional syntax, including haptic languages such as the ones used by DB people [10]. As to the possible implementations, a portable device, that could physically change hands to send and receive messages by other human parties, would arguably provide a higher degree of autonomy than garments or other wearable devices. At the same time, it could be programmed to cope with the wide variety of languages and dialects used by DB people, due to local learning environments and different degrees of sensory impairment. But it could also be a means, or at least provide a stimulus, for the users to simply learn the rules of string syntax as a universal language. Its three basic elements, i.e., categories, instances and relations, could be readily expressed by means of tactile icons, and its syntax rules are simplest and intuitive, and might help DB users to enhance their knowledge of the world [4][15]. The author has devised an interface that demonstrates this. However, such an interface is sufficiently specific and detailed to be reported in a separate paper.

X. Conclusion

The categorizing feature of natural languages provides a means to refer to items in a data aggregate that is consistent with both conventional languages and databases. This could be the basis for a communication interface connecting DB people to (a) databases, either through actively querying or updating the database or by passively receiving reports from it, or (b) other human partners, by providing a portable means to send and receive messages without the help of an assistant. Additionally, a portable interface could also provide a starting point for both DB and non-DB people to use string syntax as a universal language.

References

[1] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, "Natural language interfaces to databases - an introduction," Natural Language Engineering, 1(1), pp. 29–81, 1995.

[2] N. Caporusso, et al., "Enabling touch-based communication in wearable devices for people with sensory and multisensory impairments," in: 1st International Conference on Human Factors and Wearable Technologies, pp. 149-159, 2017.

[3] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks," IBM Research Laboratory, San Jose, California, 1970.

[4] W. S.Curtis, E. T. Donlon, and D. Tweedie, "Learning behavior of deaf-blind children. Education of the Visually Handicapped," 7(2), American Psychological Association, pp. 40-48, 1975.

[5] T. Hachisu, M. Sato, S. Fukushima, and H. Kajimoto, "HaCHIStick: simulating haptic sensation on tablet PC for musical instruments application," Adjunct Proceedings of the 24th annual ACM symposium on User interface software and technology, pp. 73-74, 2011.

[6] E. Horowitz and S. Sahni, "Fundamentals of Data Structures," Computer Science Press, 1983.

[7] J. Kramer and L. Leifer, "The talking glove," SIGCAPH Comput. Phys. Handicap 39, pp. 12–16, 1988.

[8] M. Petruck, "Frame semantics," Handbook of pragmatics, pp. 1-13, John Benjamins Publishing Company, 1996.

[9] P. Z. Revesz and R-R. K. Veera, "A Sign-To-Speech Translation System Using Matcher Neural Networks," https://cse.unl.edu/~revesz/papers/ANNIE93.pdf, 1993.

[10] L. O. Russo et al., "PARLOMA – A Novel Human-Robot Interaction System for Deaf-Blind Remote Communication," https://journals.sagepub.com/doi/10.5772/60416, 2015.

[11] C. Seror, "Human language and the information process," https://zenodo.org/record/3263753#.XRmuceszaUk, 2019.07.10.

[12] C. E. Shannon, "The lattice theory of information," in Report of Proceedings, Symposium on Information Theory, London, Sept., 1950," Institute of Radio Engineers, Transactions on Information Theory, No. 1, February, 1953, pp. 105-107..

[13] R. Vigo, "The GIST of concepts," Cognition, Vol. 129, Issue 1, Elsevier, pp. 138-162, 2013.

[14] R. Wille, "Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies," Formal Concept Analysis: Foundations and Applications. Ganter, B., Stumme, G., Wille, R. Eds. Springer, pp. 1-33, 2005.

[15] K. Wolff Heller, P.A. Alberto, and J. Bowdin, "Interactions of communication partners and students who are deaf-blind: A model," Journal of Visual Impairment & Blindness, 89(5), pp. 391-401, 1995.

[16] M. Zerkouk, A. Mhamed, and B. Messabih, "A user profile based access control model and architecture," http://www.airccse.org/journal/cnc/0113cnc12.pdf, 2013.

[17] B. Munat, "The Lowercase Semantic Web: Using Semantics on the Existing World Wide Web," http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.3445&rep=rep1&type=pdfMay 2004.

 ISBN: 978-1-61208-738-2