Collaborative Encyclopedia of Algorithms - Algpedia

**Thais N. Viana, João C. P. da Silva, Carla A. D. M. Delgado,
Carlos Eduardo da S. Martins, Fernando R. Gouvêa**

[1]Dep. de Ciência da Computação – Universidade Federal do Rio de Janeiro
Ilha do Fundão – CCMN – Rio de Janeiro – RJ – Brasil

{thais, carla}@ppgi.ufrj.br,{jcps,csmartins,fernando.srg}@ufrj.br

Abstract. *The process of building and sharing knowledge gained ground in academic research in the last decade, as more participative ways of learning became feasible. Difficult learning tasks, like learning how to program, can now count with technological support for more democratic ways of learning. In this article we present AlgPedia, a free encyclopedia about algorithms and programs in a wiki format. Algpedia's goal is to serve information not only to humans but also to other programs. Regarding human usage, AlgPedia's intent is to support and encourage the creation of collaborative content, besides helping the software developers' community to find the best algorithmic solutions and the best suited implementations for their problems. Behind the wiki interface, the AlgPedia web platform supports a semantic database that can be accessed by intelligent non-human agents.*

1. Introduction

It is widely recognised by the academic community that learning how to program is an extremely difficult activity [de Castro et al. 2004]. Many resources have been created to help in this task. With technological advancement in the last two decades new forms of learning emerged and others, as self education, gained momentum. Considering the big picture, the process of building and sharing knowledge was democratized and gained ground in academic research. As pointed out by Pierre Lévy in [Lévy 2001], the human relations with formal knowledge are changing, creating what Lévy called distributed intelligence; knowledge lives in the collective, since no one knows everything but everyone knows something.

According to [Wagner and Prasarnphanich 2007], attempts to create online content through the collective action of many individuals with no explicit incentives should be futile, as rational actors should maximize individual benefits and thus consume as much content as possible, with the least possible effort to contribute. Yet Wikipedia came to be the most successful encyclopedia in the world. A likely source of its success is its open architecture based on wiki technology[1], which makes minor contributions feasible and valuable. Also, the simplicity of the wiki gives the participants the feeling that less specialist skills are required to participate.

This dynamics fits very well to the growing tendency of creating opportunities for "hands on" learning experiences [Blumenfeld et al. 1997, Wood 2003,

[1] A wiki is a set of linked web pages, incrementally created by a group of collaborating users, as well as the software used to manage and publish the web pages.

Pedaste et al. 2015]. As Critical thinking, Communication, Collaboration, and Creativity became the so-called four Cs of twenty-first century skills [Association et al. 2012], emphasis has shifted from the way content is transmitted to the role of the individual learner in constructing understanding and the influence of the social environment on that construction.

In this article we present AlgPedia, a free encyclopedia about algorithms and programs in a wiki format. Every person is welcome to use Algpedia, either creating content, rating content, collaborating and consuming content, all of which are valid forms of learning. As a means to push the community knowledge further, AlgPedia has also the intent to help the software developers' community to find the best algorithmic solutions and the best suited implementations for their problems, for what recommendation and reputation systems based on user's content evaluation are used. At last, Algpedia has the goal to serve information to other programs, like classifiers and recommenders, so that other applications could be build and profit from the knowledge stored in Algpedia.

The following sections present in detail the AlgPedia Project and the web platform developed. In Section 2 other works that aim to support the Computer Science community with information about algorithms and their implementations are discussed and compared to AlgPedia. Section 3 presents the Algpedia Project in more detail, and the models that implement its platform. Finally, Section 4 states our main results and conclusions not only in the process of designing and building AlgPedia, but also regarding AlgPedia's usage by the community up to now, after almost three years since Algpedia was launched.

2. Related Work

AlgPedia was motivated by the absence of a portal comprising theoretic and practical information about algorithms and implementations. So AlgPedia was conceived as a project uniting structured information, comments, implementations, evaluation and performance aspects, among other related content.

There are other projects with the purpose to support the Computer Science community of developers and students in the process of learning a new programming language, like Rosetta Code[2] and Stack Overflow[3]. Rosetta Code is a programming chrestomathy[4] site, where solutions to the same tasks are presented in many different languages, to demonstrate how languages are similar and different, and to aid a person with a grounding in one approach to a problem in learning another. Rosetta Code currently has 782 tasks, 193 draft tasks, and is aware of 609 languages [Nanz and Furia 2015].

Rosetta Code and AlgPedia are both wiki repositories for computer programs related content, though there are significant differences among them. Rosetta Code main content consists of implementations for tasks while AlgPedia main content are algorithms. In Rosetta Code programs are related to computational tasks as reversing a string, whereas in AlgPedia implementations are related to more general, classic categories of algorithms as for example sorting or machine learning. This is a conceptual feature that tackles the identity of the repository and its perceived usage. Algorithms for a task (in spite of the

[2] rosettacode.org

[3] http://stackoverflow.com

[4] Chrestomathy is a selection of literary passages, specially used as an aid in learning a subject.

	Structured Information	**Implem.**	**Comments and Reviews**	**Recommendation and Ranking**
DBpedia	Yes	No	No	No
Wikipedia	No	Yes	No	No
Rosetta Code	No	Yes	No	No
Stack Overflow	No	No	Yes	Yes
AlgPedia	Yes	Yes	Yes	Yes

Table 1. Comparison among web repositories of algorithms/computer programs related content

language implemented) have a common set of features that can be documented in AlgPedia, but have no place in Rosetta Code. Also, implementations for the same task in Rosetta Code are not ranked according to any criteria, and the community can not comment the implementations. Both features are available in AlgPedia.

Stack Overflow on the other hand has comments and evaluation schema which is used to recommend and rank information [Mamykina et al. 2011, Bacchelli et al. 2012]. But it is not the goal from Stack Overflow to provide implementations for algorithms. Stack Overflow website is a place for users to ask and answer questions and, through membership and active participation, to vote questions and answers up or down.

We can consider that the information available in the web that looks most similar to AlgPedia is the subset of pages from WikiPedia about algorithms. But, as it happens in Rosetta Code, users can not add comments and evaluations (for instance, comments regarding readability or performance) to the available implementations. Also, the content is not organized in a way to prioritize showing the users the best implementations.

Regarding the semantic perspective, just DBpedia[5] provides structured knowledge. DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and Wikidata and make this information available on the web. DBpedia allows one to ask sophisticated queries against Wikipedia, and to link the different data sets on the web to Wikipedia data [Bizer et al. 2009, Morsey et al. 2011]. The aim is to make it easier for the huge amount of information in Wikipedia to be used in some new interesting ways, and also to inspire new mechanisms for navigating, linking, and improving the encyclopedia itself [Lehmann et al. 2015].

A comparison of the aforementioned projects regarding the main features from AlgPedia can be seen in Table 1. It can be seen that no other project alone would cover all features from AlgPedia. Algpedia has the specific goal to store and support the advancement of the community knowledge on algorithms and programs, whereas the other mentioned platforms have different main goals, though not less important ones.

3. The AlgPedia Project

The main goal from Algpedia is the advancement of the community knowledge on algorithms and programs, by means of the collaborative construction and access of knowledge on this subject. Regarding information access, Algpedia should deliver relevant information on computer algorithms both to humans and computers. With this is mind, the

[5] http://wiki.dbpedia.org/

AlgPedia web platform was constructed. This platform is able to serve human users, and also works to support a semantic database that can be accessed by intelligent non-human agents. Figure 1 shows a global perspective of the project. In the following sections the main modules that compose the Algpedia platform are described: *data extraction*, *web portal*, *semantic base*, *evaluation and recommendation*.

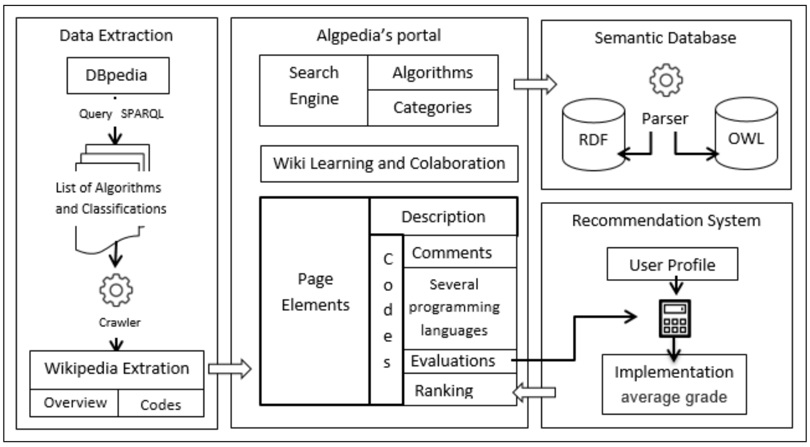


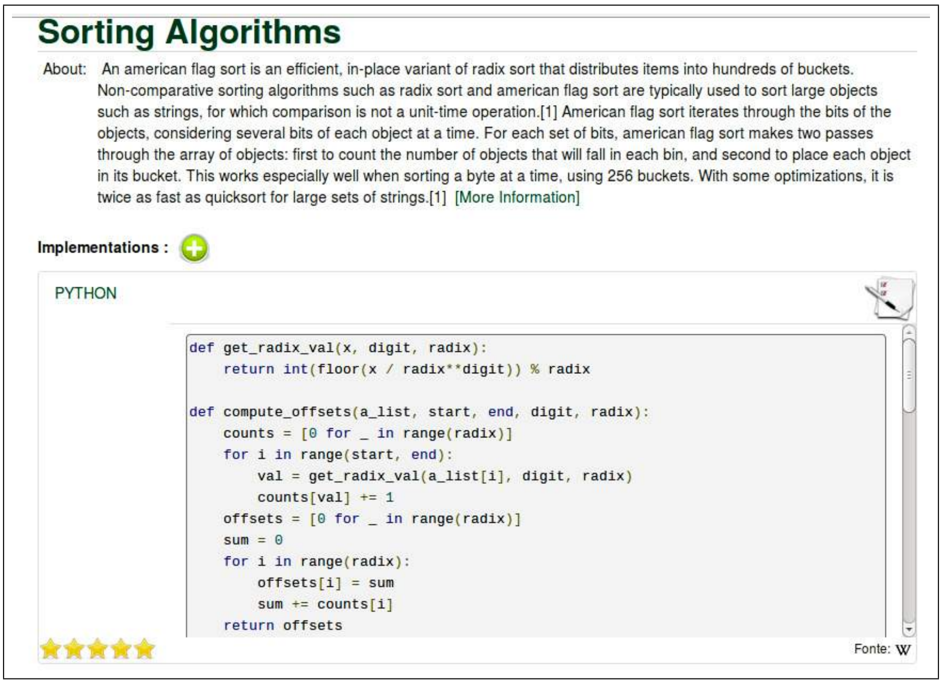
Figure 1. An overview of AlgPedia platform implementation structure.

3.1. Data extraction module

The engagement of users in a collaborative system is key for its success, but it might take time until the community comes to actively and continuously contribute to it [Xiao et al. 2007]. The purpose of the data extraction module is to create the initial content (seed data) for the wiki platform [Aigrain 2003], so that users have some content to consume even before the collaborations start to happen. A crawler [Manning et al. 2008] was created to extract information from DBpedia and Wikipedia and use this information to populate Algpedia.

At first, DBpedia Knowledge Base was queried using a SPARQL [Pérez et al. 2009] endpoint, looking for algorithms resources like *algorithm's name*, *author*, *category* and its *DBpedia and Wikipedia pages*. Then we looked for pseudo-codes and implementations (identified as code boxes) that appeared in the selected Wikipedia pages (in English and Portuguese). In case such code boxes were not found, the page was discarded. For pages where we identified such code boxes, we extracted the implementation code (or pseudo-code) and the name of the programming language used, as well as the first paragraph of the Wikipedia page to be used as a description of the algorithm in AlgPedia. We could not identify in DBpedia any specific relation that could be associated to author of an algorithm. We noted that in very few cases the information about the author could be extracted from DBpedia property *rdfs:comment* if we had in that the text *"provided by ..."*. Links to DBpedia and Wikipedia are also available in the AlgPedia page for each extracted algorithm (Figure 2).

The first extraction was done in 2014. 117 algorithms, 291 implementations and only 7 authors were successfully extracted from Wikipedia and inserted in the Algpedia's relational database. A lot of data was extracted but could not be automatically classified so we were not able to use it. Another extraction was done in late 2016, but at this time



Sorting Algorithms

About: An american flag sort is an efficient, in-place variant of radix sort that distributes items into hundreds of buckets. Non-comparative sorting algorithms such as radix sort and american flag sort are typically used to sort large objects such as strings, for which comparison is not a unit-time operation.[1] American flag sort iterates through the bits of the objects, considering several bits of each object at a time. For each set of bits, american flag sort makes two passes through the array of objects: first to count the number of objects that will fall in each bin, and second to place each object in its bucket. This works especially well when sorting a byte at a time, using 256 buckets. With some optimizations, it is twice as fast as quicksort for large sets of strings.[1] [More Information]

Implementations :

PYTHON

```
def get_radix_val(x, digit, radix):
    return int(floor(x / radix**digit)) % radix

def compute_offsets(a_list, start, end, digit, radix):
    counts = [0 for _ in range(radix)]
    for i in range(start, end):
        val = get_radix_val(a_list[i], digit, radix)
        counts[val] += 1
    offsets = [0 for _ in range(radix)]
    sum = 0
    for i in range(radix):
        offsets[i] = sum
        sum += counts[i]
    return offsets
```

★★★★★

Fonte: W

Figure 2. Sort Algorithm example

we did not consider author names and implementations. This time 2144 algorithms were extracted. There since our extractor is being improved so that we periodically check if there are new algorithms from Wikipedia to be automatically inserted in AlgPedia.

3.2. Web portal module

This module adds all features a user has available in the domain `http://algpedia.dcc.ufrj.br/`. These features can be classified in three main groups:

WikiLearning and Colaboration comprises all collaboration opportunities a user has: *add* or *edit algorithms and implementations*, *make comments*, *ask to become a moderator user*. In order to collaborate, the user needs to be authenticated so a profile creation step is require.

Search engine In general, the users of wiki projects have an access profile similar to the *known-item-search*. *Known-item-search* is a search for an item for which the author or title is known [Ogilvie and Callan 2003]. The ordinary AlgPedia user knows the name of the algorithm he is searching for, or at least the category that the algorithm he wants, before he starts to search or navigate in AlgPedia. The search system implemented in AlgPedia is not constructed on top of a query in a relational database. Our search goes through the items in classes "Algorithm" and "Classification" in the AlgPedia semantic database. More details about the Algpedia semantic database can be found in section 3.3.

Page elements Refers to the information a user will see while looking at the webpage of an algorithm in AlgPedia: a theoretical description of the algorithm, and codes. If no contribution from the community has yet been given, the description is the one extracted initially from Wikipedia. For the algorithms added via the "Add Algorithm" feature, this description is provided by the community of AlgPedia users.

Regarding the codes, in AlgPedia an algorithm can have different implementations using the same programming language and can also have implementations in distinct programming languages. Algpedia has a ranking system on the algorithms and implementations which prioritizes to show the best ranked ones first. Users can also make comments to improve the implementations, ask for help to understand the codes and get answers from the community for their questions.

3.3. Semantic base module

The Semantic base module is responsible for the construction of a RDF and OWL [Welty and McGuinness 2004] dataset from the Algpedia repository. The semantic base is made available in the initial page, where there is a link to download the Algpedia's dataset, which comprises all the data on the categories, algorithms, implementations, programming languages and relations among these entities.

The RDF dataset is created by translating the relational database of AlgPedia using DBpedia ontology resources and URI. A simple ontology was defined with the basic elements *Classes* and *Object Properties*. These elements are described in Tables 2 and 3.

class name	**description**
AUTHOR	individuals who had developed an algorithm
CLASSIFICATION	the algorithm categories extracted from DBPedia. This class has the attributes *wikipediaLink* and *dbpediaLink*
ALGORITHM	the names of algorithms extracted from Wikipedia. This class has the attributes *description*, *wikipediaLink* and *dbpediaLink*
PROGRAMMING LANGUAGES	the names of the programming language in which the extracted algorithms are implemented

Table 2. Algpedia Ontology Classes

object property	**description**
belongs	describes the relationship between one or more instances of *ALGORITHM* with one or more instances of *CLASSIFICATION*
implementation	describes the relationship between one or more instances of *ALGORITHM* with one or more instances of *PROGRAMMING LANGUAGES*. This property has the attribute *evaluation* used to classify the quality of each implementation in AlgPedia;
developer	describes the relationship between one or more instances of *AUTHOR* with one or more instances of *ALGORITHM*
developed by	is inverse property of *developer*.

Table 3. Algpedia Object Properties

3.4. Evaluation and recommendation module

As AlgPedia is a collaborative knowledge base, most of its data is provided by users. In order to provide a better user experience, we soon noticed that it was necessary to classify the provided implementations according to some quality criteria, and use this criteria to organize the information to be presented to the end user.

A ranking system was implemented in AlgPedia to prioritize the most relevant implementations. The main idea is to rank the implementations from the perspective of a set of users [Andersen et al. 2008]. The ranking schema uses (1) the expertise of the users combined with (2) evaluation scores users can give to an implementation generating (3) a reputation for each one of the different implementations of an algorithm. This reputation is then used to rank the implementations for the same algorithm.

To understand the ranking system, lets first explain how the *user expertise* is estimated. When a user joins AlgPedia, or when a person browsing an AlgPedia page decides to contribute, she is prompted to complete her user profile. The user has than to answer the set of questions that can be seen in Table 4.

	Question	**Weight**	**Answer**	**Value**
1	What is your profile? (*Prof*)	2	Professor	1
			IT-Professional	0.8
			IT-Student	0.6
			non-IT	0
2	Which university do you study or teach? (*Univ*)	1	List of Universities	$[0.5, 1]$
3	Which of these areas below do you have knowledge? (*Knowl*)	5	List of Know. Areas	$\{0, 0.5, 1\}$
4	What programming languages do you know? (*ProgLang*)	3	List of Prog. Lang.	$\{0, 0.5, 1\}$

Table 4. User profile questions

The provided information is used to estimate the users' expertise. Question 1 relates to the user background[6]. Question 2 considers the university where the user studies, works or graduated. We use the classification of the Best Global Universities List for U.S. News[7] and normalize this ranking to the interval $[0.5, 1]$ where 1 is associated to the best university of the ranking and 0.5 is associated to the universities that do not appear in the list. Note that both questions 1 and 2 have their associated values fixed for each user.

Question 3 (*Knowl*) and Question 4 (*ProgLang*) ask the user for lists of knowledge areas and of programming languages known by her. The values to be associated with this answer in the evaluation system will vary depending on the implementation the user decides to evaluate. The value to be associated depends whether the implementation category overlaps the list of the user knowledge areas (no overlap for value 0, all items overlap for value 1.0, anything in between for value 0.5) for Question 3. For Question 4, it depends if the user does not know any programming langue at all (value 0), or more languages but not the one of the implementation (value 0.5) or exactly the language used in the implementation (value 1.0).

As can be see in Table 4, each question has an associated weight, that with the values defined above, is used to calculated the *expertise* of an user u when she evaluates the implementation i on programming language l of an algorithm a. This is defined as

[6] the set of classes should be extended in the near future

[7] www.usnews.com/education/best-global-universities/rankings

follows:

$$exp(u, a, i_l) = \frac{2 * Prof + 1 * Univ + 5 * Knowl(u, a) + 3 * ProgLang(u, i_l)}{11} \quad (1)$$

Implementations in AlgPedia are shown mainly in the AlgPedia page of the algorithm they implement. When several implementations of an algorithm are available, they are ranked according to their reputation. The reputation of an implementation is obtained through evaluations from the AlgPedia's user community, according to the questions described in table 5. The user has to inform how good she evaluates the speed performance and code readability, and if she was able to compile the code.

Question	**Weight**	**Value**
Speed performance ? (*Speed*)	5	1 (slow) to 5 (fast) or 0 (not compile)
Code readability ? (*Read*)	4	1 (bad) to 5 (very good) or 0 (not know lang.)
This code compiles? (*Compile*)	3	0 if does not compile and 1 if compiles

Table 5. Evaluation score questions

The evaluation from user u for implementation i on a programming language l is given by:

$$eval(u, i_l) = \frac{5 * \frac{Speed(u,i_l)}{5} + 4 * \frac{Read(u,i_l)}{5} + 3 * Compile(u, i_l)}{12} \quad (2)$$

A reputation of an implementation is a grade in range 0 to 1. This grade is computed taking into consideration the information from users' profile and also from implementations evaluation score. Given a set of users $U = \{u_1, u_2, \cdots, u_n\}$ that evaluated an implementation i on language l of an algorithm a, the reputation of implementation i wrt users U is defined as:

$$rep(i, U) = \frac{\sum_{j=1}^{n} expertise(u_j, a, i_l) * eval(u_j, i_l)}{\sum_{j=1}^{n} expertise(u_j, a, i_l)} \quad (3)$$

The reputation of an implementation is the final value used to rank implementations. This ranking is used to show the best implementations first in AlgPedia.

4. Results and Conclusions

In this article we presented the AlgPedia Project, and the web platform that implementes it. The first AlgPedia's version released to the public access took place on July 10, 2014 at 4:00pm. In the first two days, AlgPedia got 1397 views of users 229 pages spread over 9 countries [8]. An exact week after the site was released, we got 3,685 sessions of 3,273 users from 81 different countries. Today AlgPedia counts 19,509 sessions of 15,908 users from 133 countries, with an average access of 300 sessions per month in 2016..

[8] We used the data flow analysis tool developed by Google to monitor the number of hits on our website.

AlgPedia's access growth was followed by a good number of positive feedback. Python Weekly[9] mentioned our platform as an interest project, and InfoWorld Magazine[10] ranked us as one of the top 5 websites to learn a second programming language. AlgPedia was also mentioned other times on Twitter and Facebook by many software developers.

Algpedia is still an ongoing project (and we expect it to be the case for some years), but we consider that it has fulfilled its role. It is a live, web accessible repository of algorithms and implementations. Today, Algpedia counts with a community of users who actively add comments and reviews to the implementations, and contribute with evaluations. Algpedia has thus implemented all the vital features to become a collaborative learning tool, and has reached this status. Besides, Algpedia is in a technically robust state: the semantic database handles the structured information, the wiki interface handles user contributions and navigation.

Regarding individual knowledge, we have many users actively using Algpedia: contributing, interacting, and consuming its content. Regarding advancing the knowledge of the community, the Algpedia Project also has made some contributions: (1) it led to the discovery that no satisfactory ontology relating algorithms, problems and implementations exist, and also (2) that testbeds and benchmarks to evaluate implementations would benefit to have a central repository. We are currently working on extending the Algpedia ontology to other program ontologies like COPS (Core Ontology of Programs and Software) [Lando et al. 2007] and COS (Core Ontology of Software) [Oberle et al. 2009]. Also, we are considering including testbeds and benchmarks in the the Algpedia repository (and thus in the ontology). These are the challenges we are facing now.

References

Aigrain, P. (2003). The individual and the collective in open information communities. In *16th BLED Electronic Commerce Conf.*, pages 9–11.

Andersen, R., Borgs, C., Chayes, J., Feige, U., Flaxman, A., Kalai, A., Mirrokni, V., and Tennenholtz, M. (2008). Trust-based recommendation systems: an axiomatic approach. In *Proc. of the 17th Int. Conf. on World Wide Web*, pages 199–208. ACM.

Association, N. E. et al. (2012). Preparing 21st century students for a global society: An educators guide to the four cs.. *Retrieved February 21, 2015, from the National Education Association.*

Bacchelli, A., Ponzanelli, L., and Lanza, M. (2012). Harnessing stack overflow for the ide. In *Proc. of the Third Int. Workshop on Recommendation Systems for Soft. Eng.*, RSSE '12, pages 26–30, Piscataway, NJ, USA. IEEE Press.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165.

Blumenfeld, P. C., Marx, R. W., Patrick, H., Krajcik, J., and Soloway, E. (1997). *Teaching for Understanding*, pages 819–878. Springer Netherlands, Dordrecht.

[9] Python Weekly is a free weekly newsletter featuring curated news, articles, new releases, jobs etc related to Python.

[10] `http://www.infoworld.com/article/2825055/application-development/5-sites-to-help-developers-make-a-cross-language-leap.html`

de Castro, T. H. C., de Castro Júnior, A. N., and de Menezes, C. S. (2004). Aprende–um ambiente cooperativo de apoio à aprendizagem de programação. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 1, pages 71–79.

Lando, P., Lapujade, A., Kassel, G., and Furst, F. (2007). Towards a general ontology of computer programs. In *ICSOFT*.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Lévy, P. (2001). *Cyberculture*, volume 4. U of Minnesota Press.

Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., and Hartmann, B. (2011). Design lessons from the fastest q&a site in the west. In *Proc. of the SIGCHI Conf. on Human Factors in Comp. Sys.*, CHI '11, pages 2857–2866.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge Univ. Press, New York, NY, USA.

Morsey, M., Lehmann, J., Auer, S., and Ngomo, A.-C. N. (2011). Dbpedia sparql benchmark–performance assessment with real queries on real data. In *The Semantic Web–ISWC 2011*, pages 454–469. Springer.

Nanz, S. and Furia, C. A. (2015). A comparative study of programming languages in rosetta code. In *Proceedings of the 37th Int. Conf. on Soft. Eng. - Vol. 1*, ICSE '15, pages 778–788, Piscataway, NJ, USA. IEEE Press.

Oberle, D., Grimm, S., and Staab, S. (2009). An ontology for software. In *Handbook on Ontologies*, pages 383–402.

Ogilvie, P. and Callan, J. (2003). Combining document representations for known-item search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 143–150, New York, NY, USA. ACM.

Pedaste, M., Mäeots, M., Siiman, L. A., De Jong, T., Van Riesen, S. A., Kamp, E. T., Manoli, C. C., Zacharia, Z. C., and Tsourlidaki, E. (2015). Phases of inquiry-based learning: Definitions and the inquiry cycle. *Educational research review*, 14:47–61.

Pérez, J., Arenas, M., and Gutierrez, C. (2009). Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34(3):16:1–16:45.

Wagner, C. and Prasarnphanich, P. (2007). Innovating Collaborative Content Creation: The Role of Altruism and Wiki Technology. page 18. IEEE Comp. Soc.

Welty, C. and McGuinness, D. L. (2004). Owl web ontology language guide. *W3C recommendation, W3C (February 2004) http://www.w3.org/TR/2004/REC-owl-guide-20040210*.

Wood, D. F. (2003). Problem based learning. *BMJ: British Medical Journal*, 326(7384):328.

Xiao, W., Chi, C., and Yang, M. (2007). On-line collaborative software development via wiki. In *Proceedings of the 2007 International Symposium on Wikis*, WikiSym '07, pages 177–183, New York, NY, USA. ACM.