

The AlgoViz Project: Building an Algorithm Visualization Web Community

Alexander Joel D. Alon

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Clifford A. Shaffer, Chair
Stephen H. Edwards
Edward A. Fox

15 July 2010
Blacksburg, Virginia, USA

Keywords: Educational Communities, E-Learning, Algorithm Visualization
Copyright 2010, Alexander Joel D. Alon

The AlgoViz Project: Building an Algorithm Visualization Web Community

Alexander Joel D. Alon

(ABSTRACT)

Algorithm visualizations (AVs) have become a popular teaching aid in classes on algorithms and data structures. The AlgoViz Project attempts to provide an online venue for educators, students, developers, researchers, and other AV users. The Project is comprised of two websites. The first, the *AlgoViz Portal*, provides two major informational resources: an AV catalog that provides both descriptive and evaluative metadata of indexed visualizations, and an annotated bibliography of research literature. Both resources have over 500 entries and are actively updated by the AV community. The Portal also provides field reports, discussion forums, and other community-building mechanisms. The second website, *OpenAlgoViz*, is a SourceForge site intended to showcase exemplary AVs, as well as provide logistical and hosting support to AV developers.

Acknowledgments

Several individuals deserve acknowledgment.

First and foremost, I thank Dr. Clifford A. Shaffer for his guidance and infinite patience. Thank you for leaving the door open. I thank Drs. Stephen H. Edwards and Edward A. Fox for serving on my thesis committee and providing their valuable feedback.

My colleagues and fellow developers, Monika Akbar and Michael Stewart, deserve recognition and praise. In all things AlgoViz, from the drawing board to the Drupal code base, I am indebted to them. I acknowledge my undergraduate colleagues, Taylor Eagy and Jonathan Park, for their contributions to the AlgoViz Project. Of course, I thank the AlgoViz Project Steering Committee, especially Dr. Tom Naps of the University of Wisconsin-Oshkosh, for their advice and insight in framing the Project at large.

This work was supported by the National Science Foundation under CCLI Grant # DUE-0836940 and NSDL Grant # DUE-0937783. In the course of my work I also received financial assistance from the Faculty Development Institute at Virginia Tech. I express my gratitude to these organizations.

I thank my parents, Helen and Noel Alon, for everything they have done to get me to and get me through this chapter of my life. I thank my sister, Aileen Leigh Alon, for being a close friend, confidant, and source of inspiration. I thank all of the friends and acquaintances I have made in the last three years for making my long stay in Blacksburg an enjoyable one. In particular I mention Leanna Horton, Jon Leidig, Matt Schaefer, Ben Rose, Peter Radics, and Eric Ragan. It's been a pleasure working and not working with all of you.

Last, but certainly not least, I acknowledge Sophie Kim for all her support. 감사합니다.

20 July 2010
Blacksburg, Virginia

Contents

1	Introduction	1
1.1	Issues	1
1.2	The AlgoViz Project	2
1.3	Roadmap and Contributions	2
2	A Review of Online Educational Collections and Communities	4
2.1	AV Collections	4
2.2	CS Education Resource Collections	6
2.3	General Education Collections	7
3	The AlgoViz Portal: A Repository of AV Information	8
3.1	General Technical Information	8
3.2	Algorithm Visualization Catalog	9
3.2.1	Catalog Entry Structure	9
3.2.2	Catalog Structure	11
3.2.3	Data Management and Analysis Tools	11
3.3	Field Reports	15
3.4	Annotated Bibliography	17
4	The AlgoViz Portal: A Community Venue	20
4.1	Forums	20
4.2	User Registration and Permissions	22

4.3	Staying Up-To-Date	24
4.4	Encouraging Community Participation: the AlgoViz.org Awards	27
4.4.1	Nomination	27
4.4.2	Voting	30
4.4.3	Results	30
5	OpenAlgoViz	33
5.1	Reference Implementations	33
5.2	Code Hosting Services	35
5.3	Distribution Services	36
6	Evaluations	38
6.1	AlgoViz Portal	38
6.2	OpenAlgoViz	42
7	Future Work	45
7.1	Increasing Community Participation	45
7.2	User Studies	46
7.3	Concluding Remarks	47
	Bibliography	48
	A Infrastructure View of the AlgoViz Portal	52
	B Contributed Drupal Modules used at the AlgoViz Portal	53
	C AlgoViz.org AV Catalog Keywords	56
	D The Activity Level Field in the AlgoViz.org AV Catalog	60
	E AlgoViz Portal AV Catalog OAI Mapping	62
	F AlgoViz Awards Voting Rubric	64

List of Figures

3.1	An entry in the AV Catalog.	10
3.2	The AV Catalog front page.	12
3.3	The catalog scraper user interface.	13
3.4	A histogram of catalog entries, by source code license, created using the Graph Creator tool.	14
3.5	A portion of the field report index page.	15
3.6	A portion of the field report submission form.	16
3.7	An entry in the Annotated Bibliography.	17
3.8	The Annotated Bibliography index page.	18
4.1	The Forums overview page.	21
4.2	The General Discussion forum page.	22
4.3	The account sign-up / user registration form.	23
4.4	The AlgoViz.org Twitter feed.	25
4.5	An example of an automatic notification, sent via plaintext email.	26
4.6	Node pages contain a collapsible fieldset that links to thread and content type subscriptions.	28
4.7	User profile pages contain not only the subscriptions management UI, but also links to author and content type-author subscriptions.	28
4.8	The nomination form for the AlgoViz.org Awards.	29
4.9	Part of the AlgoViz.org Awards voting slate, showing the nominees for one award category.	31
4.10	The medallion graphic distributed to the winners of the 2010 AlgoViz.org Awards.	32

5.1	The OpenAlgoViz front page.	34
5.2	The Web interface for browsing the OpenAlgoViz repository.	35
5.3	The Web interface for uploading and maintaining file releases at OpenAlgoViz.	36
6.1	Subversion activity at OpenAlgoViz throughout 2009.	42
6.2	Subversion activity at OpenAlgoViz throughout 2010.	43
6.3	Download history for all files publicly available at OpenAlgoViz.	44

List of Tables

4.1	AlgoViz.org Awards Nominees by Category	30
6.1	“Top posters” at the AlgoViz Portal as of 29 July 2010. Users listed in italics are site administrators. Users not listed here have not posted to the site. . .	39
6.2	Top twenty search terms that generated visits to the AlgoViz Portal, from 9 July 2009 to 28 July 2010.	41
6.3	Top ten referring sites for the AlgoViz Portal, from 9 July 2009 to 25 July 2010.	41
A.1	A broad view of the infrastructure of the AlgoViz Portal. The author was primarily responsible for the development of the bolded items, and provided major contributions to the development of the items in italics.	52
E.1	The mapping of AV catalog fields to unqualified Dublin Core metadata specification and a custom specification, as used by the AV Catalog OAI-PMH data provider described in Section 3.2.3. Accurate as of 20 April 2010. . . .	63

Chapter 1

Introduction

Data structures and algorithms are fundamental topics in the study of computer science, yet are commonly held to be difficult topics for CS undergraduates. As a result, educators strive to find better ways to teach them. Employing *algorithm and data structure visualizations* (hereafter referred to as “algorithm visualizations” or “AVs”), which provide graphical representations of the various states of an algorithm or data structure, has become a popular strategy for teaching and reinforcing the respective topics. In its simplest form, an AV may be a digital version of diagrams drawn on the blackboard, printed in a textbook, or shown as slides in a presentation. More complex examples may employ rich animations instead of static images, offer interactive interfaces, or require demonstrations by the user of her understanding of the material.

Many CS instructors believe that algorithm visualizations are useful instructional tools. A survey of CS instructors at SIGCSE 2010 found that the majority of respondents reported teaching with AVs in the last two years [38]. Earlier surveys at ITiCSE 2002 and 2001 showed similar results [27]. Today one can find online hundreds of AVs, which cover dozens of topics. Many of these are indexed in catalogs such as [17, 30, 9]. Meanwhile, algorithm visualization research has proven to be a strong and productive field in the last decade [24]. Much of the literature (see [19, 45] for excellent summaries) focus on evaluating AVs in terms of their usability and, more importantly, their pedagogical effectiveness. Other publications [23, 27, 31, 32] provide “best practices” for developing AVs for educational contexts.

1.1 Issues

Despite the growing body of research and number of implementations, there are still several obstacles inhibiting more widespread use of AVs in the classroom. The SIGCSE 2010 survey [38] found the three greatest impediments to using AVs in the classroom to be:

- “finding suitable AVs”,
- “integrating AV material into a course”, and
- “the difficulty in making AVs”.

Furthermore, Shaffer and Cooper made the following observations while surveying the state of the field [39]:

1. Relatively few AVs are of real pedagogical use;
2. There are “too many AVs for ‘easy’ topics” and, conversely, too few for more difficult or complex concepts;
3. A common infrastructure for storing and disseminating AVs does not exist; and
4. A strong, public venue for collaboration in using and developing AVs does not exist.

Cooper and Shaffer suggested that perhaps AVs were being developed and deployed in a “social vacuum”, in which developers and instructors alike are generally ignorant of the existing body of research.

1.2 The AlgoViz Project

The **AlgoViz Project** offers a potential solution for addressing the above issues. The overarching goal of the Project is to build an online community through which people in the field, from students and educators to developers and researchers, can find and share information on topics in algorithm visualization. Encouraging these interactions would help “focus the ongoing efforts of AV developers” to create useful visualizations, as well as the efforts of instructors to teach with AVs [36]. We hope that the efforts of the AlgoViz Project to promote better AV development practices and better AV deployment will ultimately lead to the more effective use of AVs in the classroom.

Oversight for the AlgoViz Project is provided by the AlgoViz Project Steering Committee. The Committee is comprised of instructors and researchers at academic institutions around the globe. The membership list, as well as additional information about the Committee, can be found at [37].

1.3 Roadmap and Contributions

Chapter 2 compares this work to prior efforts to catalog available algorithm visualizations as well as to related courseware collections. The following chapters discuss the author’s two major contributions to the Project, the Algorithm Visualization Portal website and the

OpenAlgoViz code repository at SourceForge.net. Chapter 3 addresses the creation and implementation of the Algorithm Visualization Portal and describes the available content. Chapter 4 describes the services that the Portal provides for building the AV community. OpenAlgoViz and other resources specifically for AV developers are discussed in Chapter 5. Chapter 6 includes early evaluations of the Portal and of OpenAlgoViz, and Chapter 7 covers a variety of ideas, proposals, and suggestions for the future.

Appendix A provides an overview of the infrastructure for the AlgoViz Portal.

Chapter 2

A Review of Online Educational Collections and Communities

There are several extant collections of algorithm visualizations on the Web, as well as collections of computer science education resources that may include AVs. In this chapter we shall review some of these collections and contrast them with the AlgoViz Project efforts.

Many of these collections are classified in the literature as repositories, referatories, or both, dependent on the nature of the collected items. For reference, a *repository* may be defined as a collection that stores its resources internally, while a *referatory* may be defined as a collection that merely links to the items, located elsewhere [18].

2.1 AV Collections

Many of the AV collections currently available online are simply large repositories of AVs produced by a single development group. Many of these collections, such as POLKA, JAWAA, JHAVÉ, and DSV, are discussed in great detail in the literature [31, 39, 45]). A stand-out example is the collection of AVs powered by the TRAKLA2 system, developed and maintained by a software visualization group at the Aalto University School of Science and Technology (formerly the Helsinki University of Technology) [22]. This repository provides a simple list of over 50 different TRAKLA2 AVs (and their variations) called “exercises”, across eight categories. Notes on using these AVs as well as interpreting the visualizations, are provided. The major issue here is that TRAKLA2 and other “single-shop” collections remain separate entities and thus do little to lessen the time and effort needed by a user comparing different AVs on the same topic.

Only a couple of online AV collections attempt to catalog existing AVs by various authors around the Web. One example is the Complete Collection of Algorithm Animations (CCAA),

is hosted Hope College in Holland, Michigan, USA [9]. Visualizations in the CCAA are categorized under seven general topics: data structures, geometric algorithms, graph algorithms, parallel algorithms, sorting algorithms, tree algorithms, and miscellaneous algorithms. The AVs are presented in a flat, browsable hierarchy; a Google Custom Search bar on the front page affords more intentional access. For each visualization, the CCAA gives the author or editor, a list of visualized algorithms and data structures, software requirements, and a brief overview of the AV. Notably, a webform is provided for users wishing to submit new visualizations. However, the CCAA affords little in the way of subjective analysis. It also has not been updated since 2001. As a result many of the links are now invalid and are not marked as such.

In 2001, participants in the Dagstuhl Seminar on Software Visualization identified the need for a comprehensive AV collection to help users find AVs, particularly ones suitable for teaching [12]. An editorial board was formed to oversee the creation of such a collection, to be called the Algorithm Animation Repository (AAR). AVs indexed in the AAR were to be “refereed and ranked according [to] a certain evaluation scheme”, allowing the better AVs to be handpicked and showcased. Special features of the AAR were to include a built-in search engine to aid exploration. Identifying the possibility to tap into the collective knowledge of the community, the board also suggested that users be able to comment on software published in the repository. Unfortunately, the AAR does not seem to have been implemented, or if it had, it was never made available online.

A similar collection is the Animal Animation Repository, hosted by the Darmstadt University of Technology [30]. The Repository is in fact both a repository and a referatory, in that some of the indexed AVs are part of the Animal Animation project, while many others are not. The AVs cover a wider variety of topics than the CCAA and include visualizations in fields such as cryptography, compression, and verification. The collection is presented as a tabular list which may be sorted by AV title, AV system used, language (since both English and German visualizations are represented), author, topic, and even by access count. This list may also be filtered by animation system, language, or topic. The Repository provides similar metadata about each visualization as the CCAA, but also provides OS requirements, the language used, as well as a BiBTeX entry for citation purposes. Optional metadata includes textual descriptions and screenshots of the AV. Visitors to the site could rate each AV on a scale from 1 to 10. Of the 233 total AVs, 138 are part of the ANIMAL collection, though this number includes English translations of German AVs. The rest come from AV systems like Catai (now defunct), JAWAA, and JSamba, as well as standalone visualizations. Though the Repository has not been updated since 2007, most of the AVs in its index have been added to the AlgoViz Catalog (see Section 3.2).

2.2 CS Education Resource Collections

There are also a number of extant collections of computer science educational materials with a broader scope.

The SIGCSE Education Links collection is maintained by Scott Grissom for the ACM Special Interest Group on Computer Science Education, or SIGCSE [17]. It contains 109 distinct resources contributed by SIGCSE members, ranging from textbooks and lecture notes to courseware and tutorials, covering all types of computer science course. Twenty of the resources are visualizations, most of which are AVs. Visitors may browse the collection by course, resource type, or author. They may view the most recent submissions, as well as the most popular entries (based on access count). They may also search the collection by keyword. For each resource, the collection provides the first author, language, software platform, and an abstract summarizing the resource. The collection appears to be actively maintained, and users may submit new resources to the collection. However, as with other collections, there is little user activity in the form of peer comments and reviews.

The CITIDEL (Computing and Information Technology Interactive Digital Educational Library) Repository was developed in part by the Department of Computer Sciences at Villanova University and the Department of Computer Science at Virginia Tech [40]. Part of the NSF-funded National Science Digital Library, CITIDEL is composed of various communities, which may contain syllabi, class descriptions, lectures, tutorials, exercises, project descriptions, and other learning objects. Users may request permission to join these communities, at which point they can submit new learning objects to the Repository. By its nature as a digital library, CITIDEL provides permanent handles to all content contained within. CITIDEL is built on the DSPACE open source digital library software. CITIDEL has since been replaced by ENSEMBLE, an NSF-NSDL Pathways project with staff at several major U.S. universities, as well as IEEE-CS, the Computer Science Teachers Association, Google, and Microsoft. Building on lessons learned from CITIDEL, ENSEMBLE aims to “establish a national, distributed digital library for computing education” [41]. Unlike in CITIDEL, learning objects are generally not stored directly in ENSEMBLE. The contents of existing collections, also called communities, are instead indexed and organized into a single, searchable catalog. This allows educators to access all of the ENSEMBLE communities simultaneously, while preserving the unique workflows and processes that community providers use to maintain and build their collections. The AlgoViz Project is in fact an ENSEMBLE partner and contributes metadata to its digital library (see Section 3.2.3).

2.3 General Education Collections

We also examine a number of collections of online educational resources that cover multiple fields and disciplines. Many of these collections are maintained by instructors and staff at a single school or school system. An excellent example is the Maricopa Learning eXchange (MLX) [26], which contains learning objects developed at institutions within the Maricopa Community College District in Arizona. Individual learning objects are represented in the MLX as packages. Each package names the author(s) or point(s) of contact, the overarching discipline, and the college at which the object was generated. A detailed description and evaluation of the object are provided; this metadata often explains not just the object itself but how it was used in a learning context. Supporting documents and media are attached to the package, as well as URLs to associated websites. While only members of the Maricopa Community College District may add new packages or leave feedback on existing ones, any Internet visitor may browse the collection by discipline, topic (e.g. critical thinking, storytelling), or originating college. They may also search by keyword, discipline, and college. As an alternate means of discovery, the MLX provides public RSS feeds for new and recently updated packages.

A self-described “Content Commons of free, open-licensed educational materials”, Connexions is a project spearheaded by Rice University [10]. Educational materials are represented as *modules* in a manner similar to that of MLX packages. However, Connexions allows users to rate modules, as well as add modules to an internal Favorites set, to external bookmarks, or to a publicly-accessible list of content called a *lens*. Users are also permitted to post modifications of existing modules.

The Multimedia Educational Resource for Learning and Online Teaching, or MERLOT, allows users to contribute valuable knowledge and opinions through peer reviews of learning objects [33]. Of all of the collections mentioned in this chapter, MERLOT also comes closest to achieving the goals of the AlgoViz Project, by providing a number of internal communities both for academic disciplines and for learning environments. Unfortunately, MERLOT lacks proper discussion forums, and appears to emphasize peer reviews of existing objects over general discussion.

Chapter 3

The AlgoViz Portal: A Repository of AV Information

The centerpiece of this thesis is the following discussion of the website called the Algorithm Visualization Portal. This website, hereafter referred to as “AlgoViz.org” or “the Portal”, is intended to be an online hub for educators, developers, and users of algorithm visualizations. It succeeds the AlgoViz Wiki, an older project by the Virginia Tech Algorithm Visualization Research Group. In fact, the two main collections at the Portal, the AV Catalog and the Annotated Bibliography, originated at the Wiki site. The Portal is available at <http://algoviz.org>.

3.1 General Technical Information

AlgoViz.org is built on Drupal, an open-source content management system written in PHP. Like most content management systems, Drupal can be used to facilitate the creation and maintenance of websites by multiple users. All major site content is stored in generic *nodes*. Whether a node represents, say, a blog post, a news article, or a forum thread depends on its *content type*. Nodes do not themselves govern what webpages look like, as Drupal separates the concerns of style and content. The Drupal engine assembles webpages from nodes using template-based *themes*.

The functionality built into the basic Drupal installation, known as *Drupal core*, can enable a website with features such as forums, blogs, news feeds, comments, and user registration. Because the source code of Drupal core is distributed under the GNU General Public License (GPL), developers can write *modules* based on the Drupal API that extend Drupal functionality [3]. AlgoViz.org is built on top of a Drupal distribution, Acquia, that packages Drupal core with many popular contributed modules [7]. For a database backend, AlgoViz.org uses a MySQL database driven by the PHP MySQLi (MySQL Improved) Extension [4].

A full listing of the contributed modules used to build and maintain the Portal can be found in Appendix B.

3.2 Algorithm Visualization Catalog

The main informational resource at the AlgoViz Portal is the *Algorithm Visualization Catalog*, a referatory of AVs on the Internet. Organized by topic, the Catalog provides for each AV a basic description, installation and usage notes, and extensive information on the developers as well as the pedagogical value of the visualization. The Catalog is intended to be a comprehensive collection of available AVs and, as such, an invaluable resource for anyone looking for AVs. It has expanded greatly in size and scope since its original incarnation as part of the AlgoViz Wiki. On 22 March 2007 it contained 371 links to AVs [39]; as of 29 July 2010 it had 515. This rapid growth can be mostly attributed to two events: the inclusion of AVs produced by the ALViE visualization environment from the University of Florence, Italy [13], and the mirroring of the Animal Repository.

Indeed, the Portal strongly encourages users to submit new visualizations to the Catalog. At minimum, users need only submit the name of the AV, a URL pointing to the AV, and a topic keyword or keywords; the remaining fields in the entry are optional. Additionally, they may update or add to an entry after it has been published to the site. Beyond adding or updating Catalog entries, users may also leave feedback on existing entries in two ways. First, users may rate an AV on a five-level scale. While the scale categories do not have specific qualitative descriptions, a rating of 1 generally denotes a poor AV, while a rating of 5 denotes an excellent AV. Secondly, users may post comments; these comments are displayed at the bottom of the entry page.

3.2.1 Catalog Entry Structure

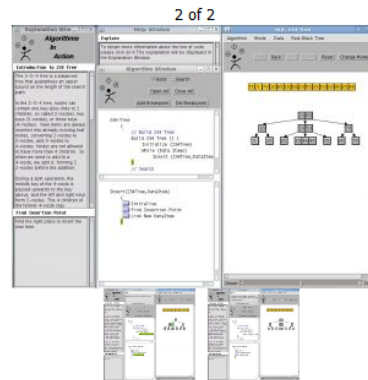
The foundation of the Catalog is its collection of URLs to algorithm visualizations online. It is not intended to be merely a simple link collection, and thus each catalog entry contains not just a link, but also key details about the AV such as its delivery mechanism, information on the developers, an objective description, and a subjective evaluation of the pedagogical value of the AV. Figure 3.2.1 shows a sample entry in the Catalog.

A description of newly-added fields follows. An **Activity Level** field is used to describe exactly how users may interact with a particular visualization (Appendix D lists the possible values.). This dimension is important, given existing literature [31] on what features of AVs make them effective teaching tools. A **References** field is used to list links to academic publications or white papers associated with a particular AV; users are encouraged to provide Digital Object Identifier (DOI) links as references whenever available. Users may also include multimedia in several forms: attached **Screenshots**, embedded **Videos**, or links to external

Figure 3.1: An entry in the AV Catalog.

Algorithms In Action - 2,3,4 Tree

Topic(s)	2-3-4 trees, B-trees, Search Structures
Link(s)	http://ww2.cs.mu.oz.au/aia/demoindex.html Report broken link
Recommendation	Recommended
Delivery Method(s)	Java Applet
Project	Algorithms In Action
Project Relationship	Part of project
Language(s)	English
Author(s)	Linda Stern, Lee Naish, Harald Sondergaard
Institution(s)	University of Melbourne
Activity Level(s)	Animation, Canned data, Random data, Step control
AV is good for	Lecture aid, Self-study
Source code is	Available but unlicensed
Last Modified	



DESCRIPTION

Demonstrates building a particular variant of a 2,3,4 Tree (B-Tree of order 4). Note that this demonstration is rather specific to 2,3,4 Trees, and can't really be viewed as a "typical" instance of how B-trees work in general. In particular, the rules for pre-splitting internal nodes on the way down during an insertion are specific to the 2,3,4 Tree.

EVALUATION

Sophisticated use of pseudocode, that can expand to show more or less detail. Has an explanation window for how the data structure works. Slightly limited in that it doesn't support delete, but does a good job at what it does support. As an extra bonus, you can pop open another window that shows the corresponding Red Black Tree (though with no pseudocode or explanation).

USAGE NOTES

The link above takes you to the AIA demonstration index page. Click on the link to the desired AV, and it should load in your browser as a multi-paned Java applet. Note that the level of detail shown in the visualization is directly tied to the level of detail that you choose to expose in the pseudocode pane.

REFERENCES

RATING

★★★★★ **Your rating:**
 ⌘ ★★★★★

Submitted by AlgoVizAdmin on 19 April 2010 - 8:51am | Last modified on 20 April 2010 - 3:21pm

[Printer-friendly version](#) [Add new comment](#) 86 reads [Add a screenshot of this AV](#)

galleries. Such material would ideally demonstrate how an AV works or even ideal use cases. Finally, a **Usage Notes** field has been added that should contain simple instructions for installing and/or running a particular AV, or a link to an instructional webpage if available. We anticipate that these added fields will be beneficial to end users, particularly instructors looking to quickly and efficiently evaluate multiple AVs for their teaching needs.

3.2.2 Catalog Structure

Catalog entries are organized by keywords representing topics in algorithms and data structures. Originally, the keywords were organized into a strict hierarchy (see [39]); submitted entries were assigned keywords by Wiki administrators. At the Portal, however, regular users can freely tag entries with new or existing keywords. Because of this, we removed the hierarchy entirely. Appendix C contains the full list.

Figure 3.2.2 depicts the Catalog front page. Here users will see a paginated list of all AVs in the collection, sorted alphabetically by AV title. However, they also may conduct simple full text searches of the Catalog. In the left sidebar users can find a shortened list of keywords that span the bulk of the Catalog; clicking on any of these keywords will filter the displayed list to show only those AVs that cover that particular topic.

3.2.3 Data Management and Analysis Tools

Cooper and Shaffer [39] described a number of tools developed for the Wiki implementation of the Catalog for analyzing Catalog data or for maintaining the catalog itself. We have added one new tool and updated existing ones with enhanced functionality. Unless otherwise specified, all the tools described below are written in PHP.

Catalog Scraper

The Catalog Scraper tool parses the entire AV catalog into a comma-separated-values (CSV) file. The Scraper provides the foundation for the rest of the tools described in subsequent sections, as the CSV file makes the catalog entries more accessible to data analysis techniques. These CSVs serve a secondary purpose as timestamped backups of the Catalog should the Wiki go offline.

The current version of the Scraper optionally allows the user to specially process one of the GoodFor, Author, and ActivityLevel fields (see Figure 3.2.3). These fields are unique in that they take values from a predefined list (all other fields are freeform). The Scraper may "expand" the field in two ways. The first processing filter gives each possible value of the selected field its own column in the data set. The second filter will list an entry in multiple

Figure 3.2: The AV Catalog front page.

The Algorithm Visualization Catalog is a comprehensive collection of links to algorithm visualizations, or AVs.

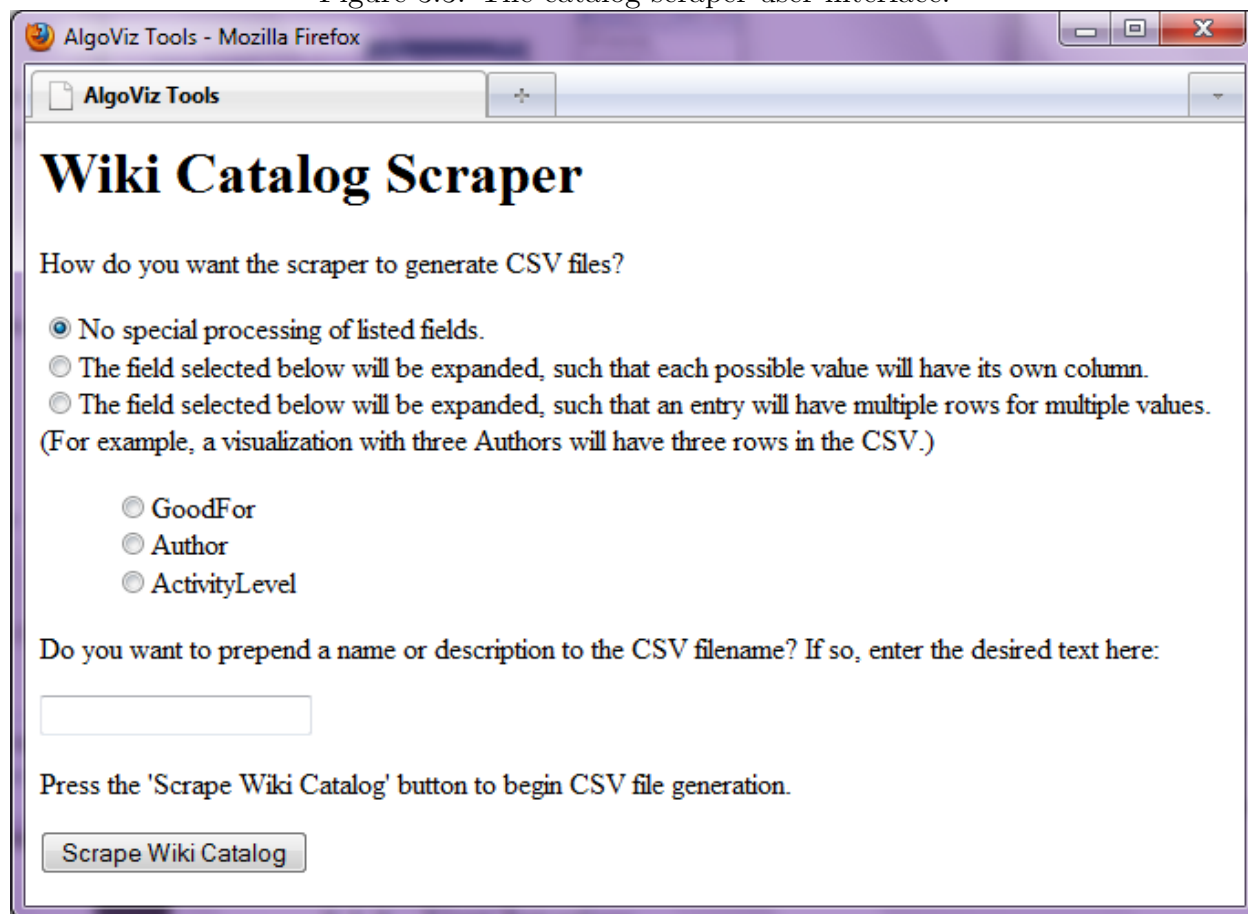
Home

AV Catalog

★★★★★	Algorithms In Action - 2,3,4 Tree	Recommended
Demonstrates building a particular variant of a 2,3,4 Tree (B-Tree of order 4). Note that this demonstration is ...		
Good For: Lecture aid , Self-study Delivery Method: Java Applet Activity Level: Animation , Canned data.. Topic: 2-3-4 trees..		
★★★★★	Algorithms In Action - Heapsort	Recommended
The application works well and shows nice, intuitive visualizations. The application launches four different windows that ...		
Good For: Lecture aid , Self-study Delivery Method: Java Applet Activity Level: Animation , Canned data.. Topic: Heapsort..		
★★★★★	Algorithms In Action - Multiway Radix Trie	Recommended
Demonstrates building a multiway Radix Trie. Given a set of values, the trie structure is built ...		
Good For: Lecture aid , Self-study Delivery Method: Java Applet Activity Level: Animation , Canned data.. Topic: Tries , Search Structures		
★★★★★	Algorithms In Action - Radix Trie	Recommended
Demonstrates building a Radix Trie. Given a set of values, the trie structure is built step ...		
Good For: Lecture aid , Self-study Delivery Method: Java Applet Activity Level: Animation , Canned data.. Topic: Tries , Search Structures		
★★★★★	Algorithms In Action - Skip List	Recommended
Demonstrates building a Skip List. Given a set of values, the trie structure is built step ...		
Good For: Lecture aid , Self-study Delivery Method: Java Applet Activity Level: Animation , Canned data.. Topic: Skip lists..		
★★★★★	ALVIE - Three-Coloring	Unrated
Walkthrough showing an NP-completeness proof for the Three-Coloring problem.		
Good For: Lecture aid , Self-study Delivery Method: Java Application Activity Level: Canned data.. Topic: NP-Completeness		
★★★★★	ALVIE - Breadth- and Depth First Searches	Unrated
Two AVs are available, one for BFS and one for DFS. Each shows a walkthrough for ...		
Good For: Lecture aid , Self-study Delivery Method: Java Application Activity Level: Canned data.. Topic: Graph Traversals..		
★★★★★	ALVIE - Dijkstra and Bellman-Ford Shortest Paths Algorithms	Unrated
Walkthroughs on an example graph for Dijkstra's and Bellman-Ford's algorithms to solve the single-source ...		
Good For: Lecture aid , Self-study Delivery Method: Java Application Activity Level: Canned data.. Topic: Single-source shortest path problem..		
★★★★★	ALVIE - Ordering Matrix Multiplications	Unrated
Walkthrough showing an algorithm for determining the best order for a series of matrix multiplications		
Good For: Lecture aid , Self-study Delivery Method: Java Application Activity Level: Canned data.. Topic: Dynamic programming..		
★★★★★	ALVIE - Closest Pair	Recommended
Demonstrates a divide and conquer algorithm for computing the closest pair of points within a speci&# ...		
Good For: Lecture aid , Self-study Delivery Method: Java Application Activity Level: Canned data.. Topic: Computational geometry		

1 2 3 4 5 6 7 8 9 ... next > last >

Figure 3.3: The catalog scraper user interface.



rows, one for each value. For example, a catalog entry that lists three Authors will have three rows in the data set.

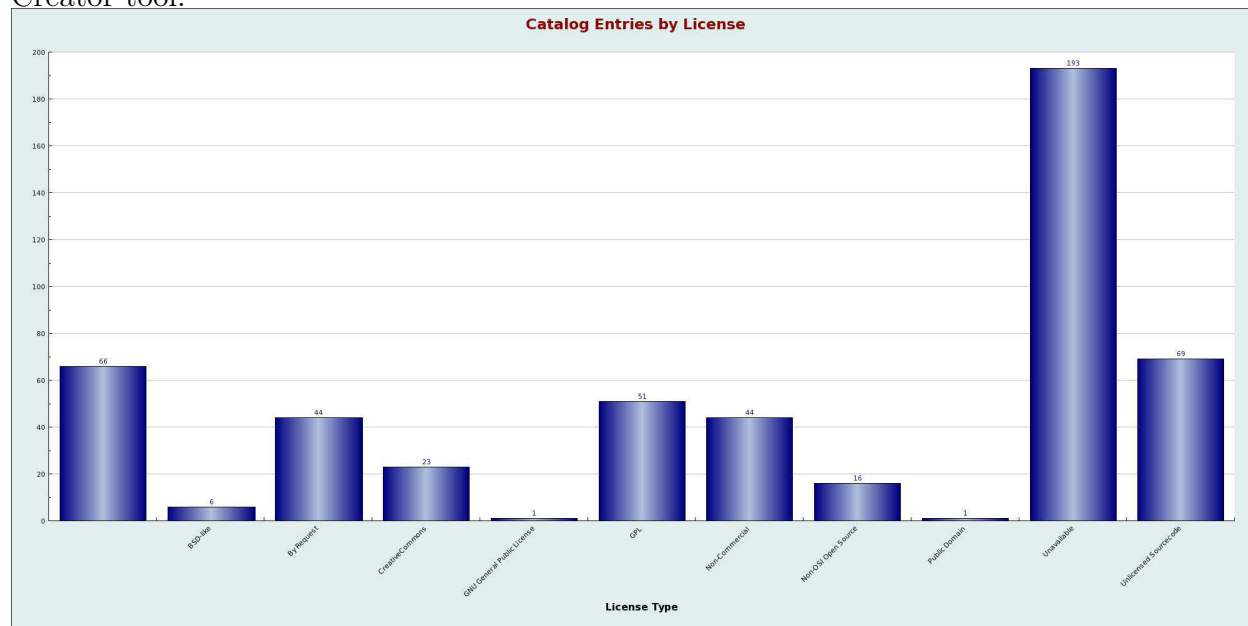
The Scraper makes use of functions available in PEAR, the PHP Extension and Application Repository [43].

Graph Creator

The AlgoViz Graph Creator tool, which generates graphs based on Wiki data, is the primary analysis tool for the catalog. It was originally written by Sean Ponce [11] and was most recently revised by Taylor Eagy [15].

To use the tool, the user must set three parameters. First, the user must choose a data set from a list of saved sets. (Alternately, the user may execute the Scraper tool to generate an up-to-date data set.) The user must then choose a graph type to generate; options include a histogram, a pie chart, and a table. Finally, the user must choose a field from the data set

Figure 3.4: A histogram of catalog entries, by source code license, created using the Graph Creator tool.



to visualize. This field becomes the dependent variable in the graph. Optionally, the user may also set the following: the graph title and axes labels; dependent variable sorting; and empty-entry suppression. The Graph Creator will then generate a colored graph, such as the one shown in Figure 3.2.3, that meets the user specifications. The user may click on any of the regions of the graph to display a list of the AV entries that constitute that region; for example, in a histogram of the “Delivery Method” field, clicking on the bar labeled will display a list of all AVs that use Flash as the method of delivery.

The Graph Creator uses JpGraph, an object-oriented graph library [1], and SQLite, a serverless, zero-configuration SQL database engine [6].

OAI-PMH Data Provider

Since 2008, the AlgoViz Project has contributed its collection of links to the CITIDEL Repository. Thus, when CITIDEL was replaced by the ENSEMBLE Project [41], it was natural that the AlgoViz Project support this new digital library as well. To ensure interoperability between the different repositories in its index, ENSEMBLE employs the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH); a complete specification can be found at [2]. We have implemented an OAI-PMH data provider that maps and exposes AV Catalog entry content to ENSEMBLE and other OAI service providers. (Appendix E describes the mapping to the unqualified Dublin Core specification, as well as a custom specification.) This makes the catalog available to a much larger audience, and allows us to

Figure 3.5: A portion of the field report index page.

Field Reports

Field Reports are meant to give instructors an opportunity to report actual experiences with specific AVs in specific course settings.

[Submit new Field Report](#) | [Field Reports Forum](#)

Author		Visualizations Used				
<input type="text"/>		<input type="text" value="Contains any word"/>		<input type="button" value="Apply"/>		
Enter a comma separated list of user names.						
	Author	AVs used	Teaser	Posted ▲	Last updated	Views
Interactive Hashing Tutorial	shaffer	Interactive Hashing Tutorial	This tutorial was used as part of a quasi-experiment to compare the pedagogical effectiveness of the tutorial as compared to standard textbook/...	2009-05-27 23:13	2010-06-27 11:17	772
Programming nondeterministic pushdown automata	rodger	JFLAP	In this lecture, students first learn about pushdown automata. Then the instructor uses JFLAP...	2009-08-09 17:10	2010-06-27 11:18	540
Interactively experimenting with finite state machines	rodger	JFLAP	In this lesson, we first introduce the formal definition of a deterministic finite automaton (DFA)...	2009-08-09 17:40	2010-06-27 11:18	274
Interactively learning SLR Parsing and how it fits in with automata	rodger	JFLAP	This lesson shows how deterministic finite automata (DFA), pushdown automata (PDA)	2009-08-09	2010-06-27	677

take advantage of various services provided by the ENSEMBLE digital library. The data provider service for the AV Catalog was originally authored by Monika Akbar.

3.3 Field Reports

It is clear that many topics in algorithms and data structures may be taught using the visualizations listed in the AV Catalog. It is also clear that these abstract concepts may be taught in many different ways; the diversity of the AVs within any particular category is proof of this. As mentioned before, a commonly-cited deterrent to the greater use and acceptance of AVs is the time and effort an instructor must spend adapting existing visualizations to her own teaching style, or to existing course content [27, 39]. To address this issue, the Portal provides a mechanism called the *field report*, which provides instructors who have employed algorithm visualizations in their lectures an outlet for sharing their experiences. The reports are designed to supplement the evaluative data in the AV Catalog, which are not necessarily based on use in actual teaching. A field report can thus provide empirical evidence to strengthen or qualify these recommendations. Additionally, field reports can be useful to those instructors preparing ACM-SIGCSE Experience Report conference papers, or simply seeking early feedback from colleagues on their work.

As of 20 July 2010 there are 14 field reports available at the Portal. These reports can be browsed via an index page (See Figure 3.3). By default, the reports are sorted by time of posting; they may also be sorted by author, by time of last update, or by the number of page

Figure 3.6: A portion of the field report submission form.

Create Field report

Field Reports are meant to give instructors an opportunity to report actual experiences with specific AVs in specific course settings.

In the "**Description and Assessment**" section, please describe in detail the lesson itself and how the AV was incorporated into the lesson. Also provide an assessment as to how successful or unsuccessful the lesson was. Feel free to provide a link to the lesson website, if one exists, or attach any documents or files for upload that would help to describe the lesson.

Report Title: *

Instructor(s): *

+

+

Name of the course instructor/lecturer/presenter. If several persons were involved, please list them one name per line item.

Location: *

Name of the institution or conference where the lesson took place.

Forums:

Select the forum you wish to post to.

Course:

+

+

The name of the course (or session, or lecture series) during which the lesson was given. A field report will typically (but not always) describe the experiences with a single course.

views.

The field report itself is a relatively freeform document. Instructors submitting a field report are asked simply to describe the lesson in question and how AVs were incorporated into the lesson; they are free to add as much detail as desired. (The field report creation form can be seen in Figure 3.3. They are asked to provide the following:

- the name of the person, or names of the persons, who presented the lesson (in case the user submitting the field report is not the same person as the actual person, or if there were multiple submitters);
- the name of the institution or conference where the lesson took place;
- the AVs used.

The following fields are optional, but may be filled in to provide more contextual information for the report:

- the name of the course, session, or lecture series;
- the academic term or dates when the lesson took place;

- any materials used during the lesson (as file attachments or URL links).

Unlike AV Catalog entries, published field reports may only be updated and edited by the report author.

3.4 Annotated Bibliography

The third major resource at the Portal is the *Annotated Bibliography*, which lists over 500 publications related to algorithm visualization research. Also originating from the AlgoViz Wiki [11], we intend for the Bibliography to be a starting point for researchers new to this field of study, as well as a repository of important work to which AV developers and users can refer.

An entry in the Annotated Bibliography contains bibliographical information about a publication. Figure 3.4 depicts one such entry. This information is presented to the user in a tabular format, but can also be displayed in RTF, EndNote Tagged, EndNote XML, or

Figure 3.7: An entry in the Annotated Bibliography.

Effective Features of Algorithm Visualizations

Submitted by parkjo on 14 July 2009 - 10:09pm

Title	Effective Features of Algorithm Visualizations
Publication Type	Thesis
Year of Publication	2002
Authors	Saraiya, P.
University	Virginia Polytechnic Institute & State University
Thesis Type	masters
Keywords	Algorithm Visualization, Education, Pedagogical Effectiveness
URL	http://scholar.lib.vt.edu/theses/available/etd-08202002-132927/

Review:

Master's thesis. This is the complete version of what was later published as Saraiya, et al 2004.

— ▶ Subscriptions (0)

Figure 3.8: The Annotated Bibliography index page.

The Annotated Bibliography is a collection of curated links to research literature on AV topics.

Home

Annotated Bibliography

List Filter

Search the bibliography

Export 519 results: RTF Tagged XML BibTeX

Author | Title | Type | Year ▲

1974

Hopgood, F. R. A., "Computer Animation used as a Tool in Teaching Computer Science", *Proceedings of the IFIP Congress*, pp. 889-892, 1974. RTF Tagged XML BibTeX Google Scholar

1975

Baecker, R. M., "Two systems which produce animated representations of the execution of computer programs", *Proceedings of the fifth SIGCSE technical symposium on Computer science education - SIGCSE '75*: ACM Press, pp. 158 - 167, 1975.

RTF Tagged XML BibTeX Google Scholar

1979

Shrout, P. E., and J. L. Fleiss, "Intraclass correlations: Uses in assessing rater reliability", *Psychological Bulletin*, vol. 86, no. 2, pp. 420-428, 1979. RTF Tagged XML BibTeX Google Scholar

1983

Add a publication to the bibliography by adding a new entry.

Subscribe to know when new publications are added.

Newest entries

- Software visualization as a pedagogical tool
- Animation Using Temporal Constraints: An Overview of the Animus System
- Colour and sound in algorithm animation

More...

BiBTeX format. A link to search for the publication on Google Scholar is also provided. But what sets the Annotated Bibliography apart from other digital libraries such as the ACM Digital Library or IEEE Xplore is that entries may also contain summaries and reviews of the literature, provided by AV experts and other members of the AV community.

Users have several options for submitting new publications for inclusion in the Bibliography. While the site allows for manual input of bibliographic information via an extended submission form, it can also parse that information from BiBTeX entries, as well as conduct Digital Object Identifier (DOI) lookups. Options for importing bibliographies formatted in BiBTeX, EndNote Tagged, EndNote XML, MARC, or RIS are also available.

The primary method for browsing the Bibliography is the index (See Figure 3.4), which by default is sorted alphabetically by publication title. Alternately, it may be sorted by author, year, or publication type (e.g., article, thesis, conference proceedings). The user may also choose to filter the index by author, year, publication type, or keyword. However, as the collection has grown exponentially in the last year -- from over 80 entries in June 2009 to 519 entries in July 2010 -- we are looking to drastically improve the browsing experience. For instance, a topical index would allow users to more easily find papers on particular AV topics. Such an index was created for the Wiki implementation of the Bibliography, and could easily be implemented at the Portal. We also intend on making the Bibliography available as a

content collection in the ENSEMBLE digital library. As with the Catalog, this will allow us to expose the Bibliography to a larger audience.

As the collection has grown exponentially, we are looking to improve targeted searching. Currently, a simple search bar on the index page provides full text searches. We are investigating the possibility of an advanced faceted search that would use the various fields in a Bibliography entry (e.g., authors, year of publication, etc.) as facets.

Chapter 4

The AlgoViz Portal: A Community Venue

The AlgoViz Portal collections represent a crucial resource for AV users of all backgrounds. However, the Portal is intended to be much more than a digital library of AV information -- it is intended to facilitate discussion and interaction between those users. In doing so the collective knowledge of the community can be shared amongst all members. As mentioned throughout Chapter 3, users have ample opportunity to provide commentary on Catalog entries, field reports, and Bibliography entries. They may comment on any of these resources, and in the case of the Catalog can also provide simple ratings. However, AlgoViz.org provides a variety of richer, more complex community-building exercises, described below.

4.1 Forums

The Portal provides several public forums for the discussion of AV-related topics. We anticipate that the forums will be the primary communication channel between both novice and expert users. At the time of writing the available subforums are:

General Discussion A forum for AV-related topics that do not fit in any of the subsequent topics. It is intended for generic questions, announcements, and news updates. For example, developers may announce new releases of AVs and AV software here.

Educators Forum A forum for topics pertaining to AVs as pedagogical aids in the classroom or lecture hall, or to classes about AVs.

Field Reports A forum where instructors who have used visualizations in teaching data structures and algorithms can report on their experiences. Field reports are discussed in detail in Section 3.3.

Developers Forum A forum for the technical discussion of AVs, it is intended primarily for programming and software development questions.

The index page (see Figure 4.1 shows for each forum the number of threads, the number of posts, and the most recent post. Users who have logged in before can see how many new and unread threads and posts are available. Clicking on a particular forum name brings up a listing of all threads within that forum (see Figure 4.1). Besides the title of each thread is the number of replies, number of views, the name of the thread creator, and the name of the person who last responded to the thread. If a thread has been updated since a user’s last visit, a conspicuous notification is displayed.

Figure 4.1: The Forums overview page.

Forums

Mark all forums read

Forum	Topics	Posts	Last post
 General Discussion For AV- and site-related topics that do not fit in the other forums.	4 1 new	18 2 new	How to ... by shaffer 2010-07-13 14:07
 Educators' Forum Discuss using algorithm visualizations as teaching aids in the classroom, as well as teaching about algorithm visualizations.	5	36	question ... by clancy 2010-04-12 22:29
 Field Reports Field Reports are meant to give instructors an opportunity to report actual experiences with specific AVs in specific course settings. You can also browse the field reports by author, post date, and AVs used.	15	23	Interactive ... by shaffer 2010-04-20 00:23
 Developers' Forum Discuss the technical and development aspects of AVs, such as programming languages, tools to use, design patterns, etc.	2	3	Gonna make ... by pauloppenheim 2010-03-31 05:25



Forum Contains New Posts



Forum Contains No New Posts



Forum is Locked




Figure 4.2: The General Discussion forum page.




General Discussion

For AV- and site-related topics that do not fit in the other forums.

[Post new Forum topic](#) [Post new Field report](#) [Mark all topics read](#)

Topic	Replies	Views	Created	Last reply
 Sticky: Introduce yourself!	7	414	by ajalon 2009-05-27 03:28	by mustafab 2010-07-02 13:12
 How to measure the AV tool? updated	1 1 new	60	by SimVis 2010-07-09 00:20	by shaffer 2010-07-13 14:07
 Why don't more instructors use AVs?	5	89	by shaffer 2010-06-17 14:21	by archie 2010-06-23 07:18
 Ideas for an AlgoViz.org Logo	1	399	by ajalon 2009-06-04 15:50	by ajalon 2009-06-04 16:00

 No New Posts  Hot Thread (No New)  Sticky Thread

 New Posts  Hot Thread (New)  Locked Thread



4.2 User Registration and Permissions

Site visitors are free to access the AV Catalog, the Annotated Bibliography, and other resources provided at AlgoViz.org. To contribute to the site, visitors must first create an account and login. While this registration process helps to limit the effects of spammers, phishers, and other malicious Internet users, it also allows visitors to cultivate a more permanent presence on the Portal. We feel that user registration will help with building the community, as users can learn to identify one another by their username or avatar. Such recognition could cultivate the trust needed for users to contribute in this public venue.

Registration comprises two steps:

- Users must fill out an account sign-up form (see Figure 4.2) linked from the front page. At minimum, an account requires only a username and a password. Fields for basic biographical information are available on the form, they are purely optional.
- Users will then receive an email containing instructions to confirm and validate the registration attempt.

Once a user has completed both steps, she will have an account at AlgoViz.org and may begin contributing to the site. The extent of a users influence, however, depends on her *user*

Figure 4.3: The account sign-up / user registration form.

User account

Account information

Username: *

Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

E-mail address: *

A valid e-mail address. The e-mail address is not made public and will not be shared with third parties.

Confirm e-mail address: *

Please re-type your e-mail address to confirm it is accurate.

Password: *

Confirm password: *

Provide a password for the new account in both fields.

Survey Information

How you heard about AlgoViz.org:

The content of this field is kept private and will not be shown publicly.

Privacy statement

This website is hosted on a Virginia Tech server. The developers of this site have created this privacy statement as part of Virginia Tech's commitment to privacy. This page discloses the information gathering and dissemination practices for this web site.

Information is Automatically Logged

This site routinely logs all accesses made by all users. We use these data primarily for two reasons. (1) These data help us to diagnose problems with our server and to administer our site. (2) As part of our research mission, we periodically publish data about Algorithm Visualizations and use of this site in various scientific venues. Such data are

* I have read and understand the 'Privacy Statement' above.

roles, which governs what she is permitted to do at the Portal. By default all registered users are assigned the *authenticated user* role, which grants the ability to submit content to the site, to leave comments on existing content, and to post in the forums. Authenticated users may also edit anything they have submitted. Content submitted by authenticated users must be approved before it will be published to the site. Among other things, this approval requirement is used to limit the amount of posted spam. Approval is waived for users with the *content author* role, such that content submitted by a content author is automatically published. Actual approval of submitted content is the responsibility of users in the *content editor* role, which also grants the freedom to revise or delete any content on the site. The *forum moderator* role allows elevated access on the Portal forums. Moderators may administer the different subforums: they may edit, move, or delete forum posts and forum topics, and they may block users from posting to the forums at all. Finally, users in the *site administrator* role are tasked with site upkeep: they may manage the user list and user permissions table, build site templates, and maintain the codebase and database backend. Currently, only members of the AlgoViz Project team are site administrators.

A common complaint of Internet users is the growing number of user accounts they must create and manage while surfing the Web. User registration forms like the one in Figure 4.2 are commonly considered a major barrier to participation and conversion rate. To address this, we currently provide a partial implementation of the OpenID authentication standard. In this implementation, an authenticated user may link her account to one or more OpenIDs; she may then use a linked OpenID to access the Portal instead of her AlgoViz.org account. We are currently working to allow visitors to completely forgo user registration if they provide an OpenID from the beginning. This solution would not only permit users to have to remember a single digital identity -- the OpenID -- but would also allow us to forego having to store and protect password information. We are also investigating Facebook Connect and Google Friend Connect as alternative authentication solutions.

4.3 Staying Up-To-Date

While we expect users will find AlgoViz.org a valuable resource, we do not expect most to visit frequently. To encourage users to return to the Portal regularly, the site provides a number of services for notifying them of changes, updates, and additions. In designing these notification services we considered two things: firstly, users should be able to receive notifications in ways that fit into their normal information flows; secondly, users should be able to control their notifications so that they may only be informed of content they find "interesting". To meet these goals we send notifications through a variety of communication mediums and allow users to tailor and filter their notification streams. These notification methods are our primary mechanism for disseminating information about AVs throughout the educational community and beyond.

The first service offered is an *email newsletter service*. Users who subscribe to this service will

Figure 4.4: The AlgoViz.org Twitter feed.

The image shows a screenshot of the AlgoViz.org Twitter profile. At the top, the Twitter logo is visible on the left, and navigation links (Home, Profile, Find People, Settings, Help, Sign out) are on the right. The profile header includes the 'algoviz' name and a circular logo with 'Algo' and 'viz' text. Below the header is a 'Follow' button and 'Lists' and 'Settings' dropdowns. The main content area displays a tweet from the 'AlgoViz Project Steering Committee' with a link to 'http://algoviz.org/node/1351'. Below this are several 'AV of the Day' tweets, each with a link to a specific visualization on the website. The right sidebar shows statistics: 1 following, 17 followers, and 3 listed. It also includes sections for 'Tweets' (292), 'Favorites', 'Actions' (block algoviz, report for spam), 'Following' (one user), and an 'RSS feed of algoviz's tweets'.

occasionally receive email newsletters from the AlgoViz.org administrators; these newsletters can serve many purposes, such as summarizing site updates, describing recent happenings in the algorithm visualization community, or eliciting user contributions. Each newsletter issue is archived after sending and may be viewed on the site at any time. All members of the AlgoViz.org community are automatically subscribed to the email newsletter during registration; unregistered users may still subscribe by providing valid email addresses. A user may opt out of the subscription at any time.

Alternately, users may subscribe to one of our *RSS feeds*. These feeds automatically index various sections of the site, from the Bibliography to news stories. We also provide a general RSS feed that covers all generated content on the Portal. Users may also “follow” our *Twitter feed* (see Figure 4.3 which is publicly available at <http://twitter.com/algoviz>). Most of the tweets in the feed link to newly posted content, from catalog entries and field

Figure 4.5: An example of an automatic notification, sent via plaintext email.

```
Dear ajalon,  
  
These are your messages for 30 April 2010 - 1:47pm:  
  
Updates for Catalog entry "UniSorter - MergeSort"  
- The Catalog entry "UniSorter - MergeSort" has been updated  
(http://algoviz.org/node/1298)  
  
Read more http://algoviz.org/node/1298  
Updates for Catalog entry "UniSorter - Radix Sort"  
- The Catalog entry "UniSorter - Radix Sort" has been updated  
(http://algoviz.org/node/1299)  
  
Read more http://algoviz.org/node/1299  
  
--  
This is an automatic message from AlgoViz.org. Visit http://algoviz.org/user/3/notifications to manage your subscriptions or to unsubscribe.
```

reports to news stories and polls. The site also automatically posts a randomly-selected “AV of the Day”.

The key notification service, however, is an *automatic content subscription system* that allows users more fine-grained control over their notification streams (thus fulfilling our second goal for notifications). This system sends out automatic notifications triggered by one of three basic events. A *node creation* event occurs whenever a new node is published to the Portal. A *node update* event occurs whenever an existing node is edited. Finally, a *node comment* event occurs whenever someone comments on an existing node. By default, the notifications are sent via plaintext email to the email address attached to a user’s profile (see Figure 4.3 for an example). Alternately, users may opt to receive their notifications by HTML email. They may also choose to be notified of site updates by Twitter Direct Message (DM) [44], so long as a valid Twitter account is provided. Users may choose to receive a separate notification for each event, delivered immediately, or to receive a notification digest listing all events in a given time period.

Users may choose between four types of subscriptions. The first type is the *thread subscription*, which permits per-node notifications. Users who have subscribed to a node will receive update and comment notifications for that node. Users may subscribe to any node that they have permission to view; node authors are automatically subscribed to the nodes that they create. The second type is the *content type subscription*. Subscribing to a particular content type lets a user receive update and comment notifications for any node of that type; she will also receive creation notifications for new nodes of that type. Users may subscribe to any of five content types:

- catalog_entry
- forum (forum topics)
- field_report
- biblio (Annotated Bibliography entries)

- story (news stories posted to the front page)

The third type is the *author subscription*. This subscription allows users to be notified of any node-centric activity the target user performs. The fourth type is the *content type-author subscription*, which is a hybrid of the previous two subscription types. For example, user A may create a subscription on the ‘forum’ content type and on user B; user A will then receive notifications of all of user B’s forum activity. The author subscription is functionally equivalent to content type-author subscriptions on all content types.

Users may toggle these subscriptions in various parts of the Portal. A collapsible fieldset is provided on node pages for thread and content type subscriptions, as in Figure 4.3. Links for author and content-type author subscriptions are provided on user pages, as in Figure 4.3. Users may also manage all of their subscriptions via their own user profile pages.

4.4 Encouraging Community Participation: the AlgoViz.org Awards

Services like those described above provide users myriad opportunities to contribute and to participate. However, an initial impetus for users to use those services is usually required. This section describes not a community-building service of the Portal, but a means of promoting the site and driving traffic. We created the *AlgoViz.org Awards* to highlight outstanding algorithm visualizations. We hoped that conferring the responsibilities of nomination and voting, described in detail below, on the general AlgoViz.org community, would encourage users to participate.

4.4.1 Nomination

The nomination process began with the public announcement of the AlgoViz.org Awards on 16 June 2009 and closed on 1 October 2009. Nominating visualizations was a privilege granted to registered, authenticated users, though several AVs were selected by the AlgoViz Project Steering Committee to initially seed the competition. To nominate an AV, users simply had to fill out a form (see Figure 4.4.1). Users were asked to provide the name of the visualization, a brief description of the visualization, a link to its entry in the AlgoViz Catalog, and their reasons for nomination. We only allowed nominations of AVs already indexed in the Catalog, suggesting that users who wanted to nominate AVs not in the catalog should submit catalog entries as well.

Recognizing that algorithm visualizations may take many different forms and that simple comparisons of the nominees would be inherently unfair, we also asked users to assign their AV nomination to one of three award categories. The first category, **Standalone AVs in**

Figure 4.6: Node pages contain a collapsible fieldset that links to thread and content type subscriptions.

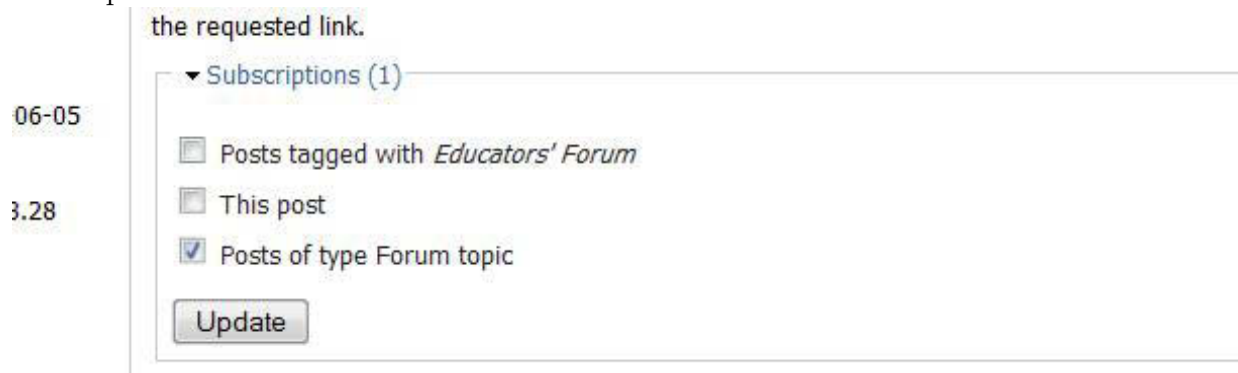


Figure 4.7: User profile pages contain not only the subscriptions management UI, but also links to author and content type-author subscriptions.

My profile



Current status:

- You have 6 active subscriptions.
- Your default sending method for new subscriptions is Mail
- Your default sending interval for new subscriptions is Every hour

You can:

- [Administer your subscriptions](#)
- [Edit your notifications settings](#)
- [Temporarily disable all your subscriptions](#)
- [Cancel all your subscriptions](#)

Figure 4.8: The nomination form for the AlgoViz.org Awards.

Create AV.org Award nominee

To make a nomination for the AlgoViz.com Awards, just fill out this form. The awards committee will review your nomination before adding it to the voting slate.

Nominated Artifact: *

AlgoViz.org Award Categories: *

- Please choose -

Please choose a category for this nomination.

URL:

Link to the corresponding [Wiki catalog entry](#). **We will not accept nominations of AVs that are not already in the catalog!** You can create a new catalog entry [here](#).

Description:

Brief description of the artifact. You might get some ideas from the "Description" field in the corresponding AV Wiki catalog entry.

Switch to rich text editor

— ▶ Select an input format

Reasons for Nomination: *

Why does this AV deserve an award? You might get some ideas from the "Evaluation" field in the corresponding AV Wiki catalog entry.

[+] Feed

Table 4.1: AlgoViz.org Awards Nominees by Category

Award Category	# Nominees
Standalone AVs, CS1/CS2 Topics	3
Standalone AVs, Advanced Topics	6
Online Tutorials/Hypertextbooks	5
Total	14

Lower Division Topics, represents AVs that cover topics generally taught in CS1, CS2, and sophomore-level courses. The second, **Standalone AVs in Advanced Topics**, represents AVs that cover topics taught in upper division and graduate-level computer science courses, as well as topics in specialized fields. The third category, **Online Tutorials and Hypertextbooks**, represents AVs that are integrated into a larger learning object, instead of presented alone. The aforementioned award categories were chosen not only to “level the playing field” for different AV types, but to draw attention to the various issues of AV development (see Section 1.1). It is our hope that by categorizing nominated AVs into these three topics we highlighted the continuing need for better AVs in the future.

4.4.2 Voting

Authenticated users were also given the opportunity to vote on the actual award winners. The voting slate listed all of the nominees, sorted by award category (see Figure 4.4.2). Users could read the description and reasons for nomination, as well as visit the actual AV (by way of the corresponding catalog entry) before voting.

The actual "voting" process involved rating each AV on a five-point scale, where one point represents a poor AV and five represent an excellent AV. Users were provided with a list of rating criteria to guide their decisions; these criteria, which are reproduced in Appendix F, emphasized classroom usefulness over pure aesthetics. Users were permitted to rate as many or as few AVs as desired, and were allowed to change their votes as many times as desired until voting closed. Voting began on 16 June 2009, the same day as nomination. We chose to open nominations and voting at the same time to allow users the opportunity to vote on the pre-selected nominees, to nominate other AVs, and to vote on these newer nominees. Voting ended on 31 January 2010.

4.4.3 Results

In total, fourteen AVs were nominated for an AlgoViz.org Award. All fourteen were submitted before 31 August 2009 by members of the Steering Committee.

The two highest-rated AVs in each category were deemed award winners, for a total of six

Figure 4.9: Part of the AlgoViz.org Awards voting slate, showing the nominees for one award category.

Category: Standalone AV, CS1/CS2 Topics

An algorithm visualization that covers topics in data structures and algorithms that often appear in CS1 or CS2 courses.

Algorithms In Action - Quicksort

<http://wiki.algoviz.org/AlgovizWiki/Viz/AlgInActionQuicksort>

Description: Presentation of the Quicksort algorithm. Expandable pseudocode display controls the level of detail for the visualization.

Reasons for Nomination: The application works well and shows nice, intuitive visualizations.

Average:
★★★★★
Average: 2.5 (13 votes)

Your rating:
☒ ★★★★★

Binary Treesome

<http://wiki.algoviz.org/AlgovizWiki/Viz/BinaryTreesome>

Description: Allows the user to see a number of binary tree variants, including the BST and AVL trees.

Reasons for Nomination: Clear presentation of the content. Students are not just passive observers, but can show their knowledge by selecting where the next action should take place.

Average:
★★★★★
Average: 2.2 (13 votes)

Your rating:
☒ ★★★★★

VT - Radix Sort

<http://wiki.algoviz.org/AlgovizWiki/Viz/VTRadixSort>

Description: Radix sort visualization.

Reasons for Nomination: Clear, step-by-step explanation for everything that is taking place.

Average:
★★★★★
Average: 3.5 (13 votes)

Your rating:
☒ ★★★★★

Figure 4.10: The medallion graphic distributed to the winners of the 2010 AlgoViz.org Awards.



AVs. These winners were publicly declared in a video presentation on 13 March 2010 at the 41st Annual SIGCSE Technical Symposium on Computer Science Education, during a Special Session titled “Building an Online Educational Community for Algorithm Visualization” [34]. This video gives a quick demonstration showcasing the strengths of each winning AV. We recently distributed to the developers of the award-winning AVs a simple medallion graphic (depicted in Figure 4.4.3 to be published on their respective sites. We hope that in time the graphic will be considered a mark of quality and excellence, encouraging the use of these winning AVs as well as promoting the AlgoViz.org Awards and the AlgoViz Portal.

Chapter 5

OpenAlgoViz

Developers of algorithm visualizations, like other AV users, will certainly find the AlgoViz Portal to be useful. However, the Portal alone does not meet some special needs of this class of users. The difference between AV developers and other users is simple: while the latter are concerned only with how a visualization looks and performs, the former must also, obviously, consider how to implement the visualization. Developers may search the AV Catalog (Section 3.2) for exemplary AVs to reference or emulate, and they may ply the research literature for best practices in AV development (Section 3.4) to apply. However, they will need to architect and program the source code that builds their own AVs, and in this regard will need two things:

1. Access to reference implementations, particularly the source code, of quality AVs and AV-related tools.
2. Access to services that can aid in the logistics of software development.

With these needs in mind, we have created a SourceForge.net project called *OpenAlgoViz*. This sister site to the AlgoViz Portal (see Figure 5) contains the source code for a number of quality open-source implementations of AVs and AV systems. For developers of AVs, OpenAlgoViz also offers various code hosting, development, and distribution services.

5.1 Reference Implementations

As the Portal hosts an AV Catalog that links to various AVs, OpenAlgoViz hosts a Subversion repository containing the source code of various AVs and AV-related tools of high quality and/or usefulness. As all of the source code in the repository is released under various Open Source Initiative (OSI)-approved licenses, such as the GNU General Public License or the MIT License, developers are free to download any or all of it for their use (although how they

Figure 5.1: The OpenAlgoViz front page.

OpenAlgoViz
Open Source Algorithm Visualizations

Navigation

- [SourceForge Project Page](#)
- [Algorithm Visualization Portal](#)

Downloads

- [Union-Find](#)
- [Hashing](#)
- [JHAVE](#)
- [Internationalization](#)

Summary

We seek to provide a repository of example implementations and utility code for developers of Algorithm Visualizations (AVs). We provide a central point of distribution for many AV developers and projects.

Requirements

If you would like to include your distribution within the OpenAlgoViz repository, it must be directly relevant to AV in some way (be an AV, an AV-building toolkit, or a classlibrary useful for implementing AVs). All sourcecode must meet the SourceForge requirement of being licensed under a standard Open Source license such as an Open Source Initiative Approved License (also a requirement for all software hosted at SourceForge).

Developer Information

Repositories

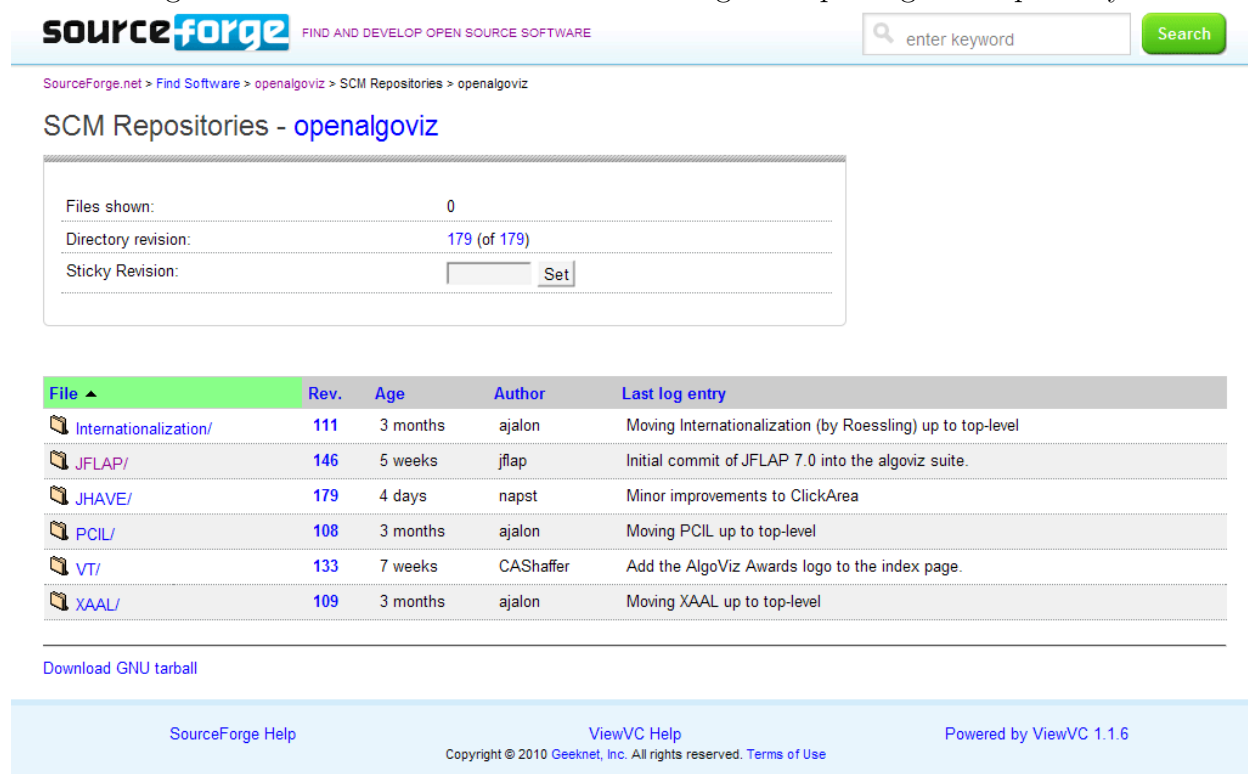
SVN browse code, statistics, most recent commit

```
svn co https://openalgoviz.svn.sourceforge.net/svnroot/openalgoviz openalgoviz
```

GIT browse code, statistics

```
git://openalgoviz.git.sourceforge.net/gitroot/openalgoviz/openalgoviz (read-only)
```

Figure 5.2: The Web interface for browsing the OpenAlgoViz repository.



sourceforge FIND AND DEVELOP OPEN SOURCE SOFTWARE

enter keyword Search

SourceForge.net > Find Software > openalgoviz > SCM Repositories > openalgoviz

SCM Repositories - openalgoviz

Files shown: 0

Directory revision: 179 (of 179)

Sticky Revision: Set

File	Rev.	Age	Author	Last log entry
Internationalization/	111	3 months	ajalon	Moving Internationalization (by Roessling) up to top-level
JFLAP/	146	5 weeks	jflap	Initial commit of JFLAP 7.0 into the algoviz suite.
JHAVE/	179	4 days	napst	Minor improvements to ClickArea
PCIL/	108	3 months	ajalon	Moving PCIL up to top-level
VT/	133	7 weeks	CASHaffer	Add the AlgoViz Awards logo to the index page.
XAAL/	109	3 months	ajalon	Moving XAAL up to top-level

[Download GNU tarball](#)

SourceForge Help ViewVC Help Powered by ViewVC 1.1.6

Copyright © 2010 Geeknet, Inc. All rights reserved. [Terms of Use](#)

may use that code varies depending on the corresponding license). Accessing the repository itself is simple: users may browse the repository via a web interface like the one provided by SourceForge (See Figure 5.1) or via a dedicated Subversion client.

Current contributions to the OpenAlgoViz repository include the Interactive Hashing Tutorial and the Union-Find Algorithm Visualization, both by the Virginia Tech Algorithm Visualization Research Group [35, 20]; the JHAVÉ visualization system [42]; the JFLAP system from Duke University [29]; the PseudoCode Interpreted Language project [25]; the XAAL (XML Algorithm Animation Language) project [21, 22]; the GAIGS scripting language project; and internationalization libraries by Guido Roessling. All contributions were solicited by the AlgoViz Project. In time, we hope that OpenAlgoViz will exhibit an exemplary cross-cut of open-source AV software.

5.2 Code Hosting Services

A major concern of software developers is finding a safe and secure place to store their code. OpenAlgoViz provides a solution for this by offering code hosting services for AV-related software projects [5]. The primary advantage of hosting on OpenAlgoViz is having full

Figure 5.3: The Web interface for uploading and maintaining file releases at OpenAlgoViz.

OpenAlgoViz

Summary | Files | Support | Develop | Hosted Apps | Tracker | Mailing Lists | Code | Project Admin

Home

+ Add File + Add Folder

Name ▾	Date ↕	Size ↕	Downloads ↕
AlgoViz-VT	2009-08-26	529.1 KB	0 <input type="text"/>
Internationalization	2009-09-30	194.9 KB	0 <input type="text"/>
JFLAP	2010-07-27	22.5 MB	9
JHAVE	2009-09-30	3.1 MB	0 <input type="text"/>
OldFiles	2009-02-25	26.4 KB	0 <input type="text"/>
PCIL		0 Bytes	0 <input type="text"/>
XAAL		0 Bytes	0 <input type="text"/>
Totals: 7 Items		26.3 MB	9

Tip: Upload files using rsync, SCP and SFTP.

access to the repository. Indeed, the developers of JHAVÉ, JFLAP, and the Virginia Tech visualizations mentioned above currently use this service and are actively developing their projects on OpenAlgoViz.

To make use of this and other services, a developer needs only to register a SourceForge.net account and ask an OpenAlgoViz administrator to link that account to the project. Project members may then commit their work to the repository, provided it is distributed under an OSI-approved license. They may then checkout the code at any time, from anywhere on the Internet, and commit changes as needed. Revision control is another advantage of using OpenAlgoViz, for the Subversion system maintains differential copies of each revision of each file in the repository. Thus, developers may use all of the features built into Subversion to help manage their source code.

5.3 Distribution Services

OpenAlgoViz also provides several SourceForge-powered distribution services [5]. One service is the File Release System (FRS), which gives developers a simple way to make pre-packaged software available for public download. If a developer is working on an AV that runs on multiple platforms, she can upload different versions of the AV to SourceForge (see Figure 5.3;

the File Release System will automatically offer the appropriate version for download based on the end user's platform. Files uploaded through the are stored on a worldwide network of download mirrors. This guarantees download performance, since a user can download parts of a file from multiple mirrors simultaneously. It also guarantees file availability; if one server in the SourceForge network goes down, the user can simply resume or restart a download from another mirror.

Finally, OpenAlgoViz tracks download and repository activity; this information is made available to all project members (see Section 6.2 for current statistics).

This is made possible by the fact that the OpenAlgoViz codebase is in fact a Subversion repository. This is convenient for the developers of those systems, since instead of maintaining their own code on local servers that may or may not be accessible anywhere on the Internet, they may rely on the robust and globally-accessible SourceForge.net platform. This also means that users will always have access to the latest builds of these particular AVs.

Chapter 6

Evaluations

At this early stage in the efforts of the AlgoViz Project, it is difficult to say whether it will ultimately be successful in fostering an online community of AV users, or whether it can increase AV use in classroom. Much of the work could be classified as building the infrastructure needed to be able to address the issues presented in Section 1.1. Nonetheless, in this chapter we will attempt to quantify the success of both the Portal and of OpenAlgoViz, by looking at a number of measurable web metrics.

6.1 AlgoViz Portal

As the Portal provides both a clearinghouse of AV information as well as a variety of community services, we will need to evaluate how well it does both. We will use a number of metrics designed by Reed [28], presented below, to measure the “health” of an online community, by measuring the quality and quantity of community members’ interactions and contributions. Reed actually defines several additional metrics that will be ignored in this evaluation as they are either subjective and thus not easily quantifiable, or because they does not apply to the Portal.

1. The number of posts within each discussion topic or thread
2. The overall number of new contributions
3. The amount of time members spend on your community
4. The number of searches taking place in your community
5. The number of search terms being used to find your community
6. The number of links your community is picking up from external websites

From these metrics we find that the Portal has had limited success while it has been active. For instance, the Portal has seen little activity over its lifetime in terms of the first metric,

“the number of posts within each discussion topic or thread”. Looking strictly at the public discussion forums, including the Field Reports forum, there are (at the time of this writing) a mere 80 forum posts in 33 threads, for a mean of 2.4 posts per thread. Furthermore, only 13 comments have been posted in other areas of the site.

The Portal does moderately better at the second metric, “the overall number of new contributions”. Table 6.1 shows a list of those users who have contributed content to AlgoViz. As can be seen, site administrators contributed a high percentage (approximately 70%) of the published content; this can be attributed to the fact that most of the content at AlgoViz.org was programmatically migrated by the administrators from the AlgoViz Wiki. The overwhelming majority of the remaining content lies in contributions to the Annotated

Table 6.1: “Top posters” at the AlgoViz Portal as of 29 July 2010. Users listed in italics are site administrators. Users not listed here have not posted to the site.

User	Post Count
<i>AlgoVizAdmin</i>	538
<i>shaffer</i>	162
guido	139
ville	97
<i>ajalon</i>	75
<i>parkjo</i>	71
hundhaus	49
rodger	35
piluc	26
naps	5
affandy	5
<i>Monika</i>	4
clancy	3
mmcnally	2
furcyd	2
chelwig	2
eurostat	2
archie	2
scottgrissom	2
<i>tgm</i>	1
mustafab	1
thanasis	1
SimVis	1
pauloppenheim	1
cabrera	1
Total	1227

Bibliography. As mentioned in Section 3.4, the Bibliography has six times as much content as it did just over a year ago. It is important to note, however that these contributions have come mostly from various AV researchers offering their personal research bibliographies to the site. In truth, most of the entries were added to the Portal and attributed to the researchers by site administrators. Though one could consider these non-contributions, since they were not added to the Portal directly, we prefer to think of them otherwise, as they were still offered for inclusion in the Bibliography regardless of the mechanism. Regardless, it is hard to argue that the Bibliography can be considered the most successful aspect of the Portal thus far.

The third metric can be measured using data provided by the Google Analytics web analytics software, which has tracked usage data and site traffic at the Portal since 9 July 2009. On average, users visiting the Portal have spent 4 minutes and 51 seconds at the site, with an overall bounce rate of 50.15%. (The *bounce rate* refers to the percentage of single-page site visits, or visits in which the user left the site from her “landing” page.) It is hard to interpret this relatively high bounce rate: on the one hand, we would like users to stay at the site and contribute. Conversely, this rate may mean that users are finding the AVs or publications at the site. In the future we will need to begin tracking outgoing links in order to determine how users are leaving the Portal.

The fourth metric addresses internal site search. Though the Portal provides site search functionality, there is little data available on the actual searches. As the search capabilities of the Portal have been a low priority until recently, we expect more useful data once we have enriched the search experience. Perhaps more useful is the fifth metric, on referring search terms. Table 6.1 lists the top twenty search terms that produced visits to the Portal. Two of the terms, including the top search term, are titles of publications listed in the Annotated Bibliography. Six terms are variants on the keyword “algorithm visualization”, which bodes well considering that the Portal is intended to be a definitive AV resource. Eight terms reference particular topics in algorithms and data structures. (Further exploration of such search terms may be invaluable for AV developers.) Finally, the fifth-most-relevant term, “sigcse survey”, references the survey conducted at SIGCSE 2010 [38], which advertised the Portal to respondents. Clearly, the wide variety in search terms being used to reach AlgoViz.org is encouraging.

Finally, we find that, from 9 July 2009 to the present, users went to the Portal directly 4,047 times. In contrast, users followed links from 155 different websites to AlgoViz.org 2,270 times. Table 6.1 lists the top ten referring sites. The top site is the AlgoViz Wiki, which should drop in ranking as more content is migrated to the Portal.

In evaluating whether the Portal is succeeding as a community venue, it may be tempting to look at the number of user registrations or user accounts. Indeed, there have been over 2700 unique registrations since the site went online. However, as of 14 July 2010 only 160 of those registrations represent actual users who have completed the registration procedures outlined in Section 4.2. Reed [28] notes that member count and similar measures are not

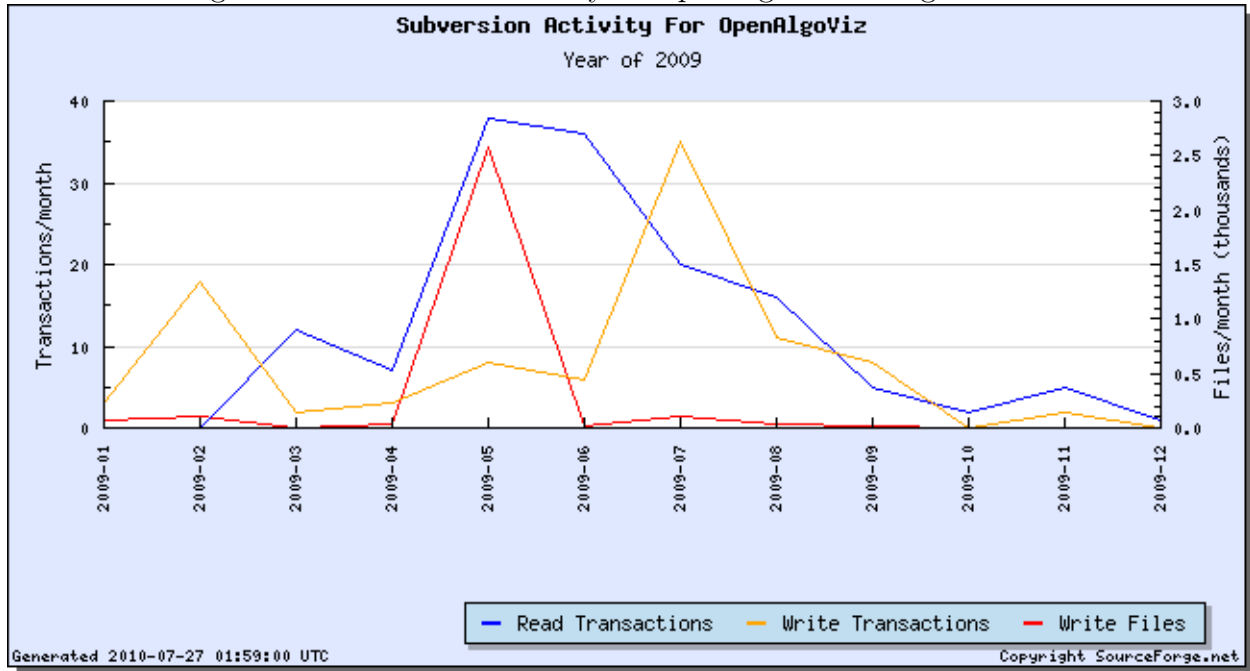
Table 6.2: Top twenty search terms that generated visits to the AlgoViz Portal, from 9 July 2009 to 28 July 2010.

Keyword	Visits
empirically evaluating the use of animations to teach algorithms	352
algoviz	333
algorithm visualization	177
hashing tutorial	65
sigcse survey	41
slr parsing	38
algovis	33
http://research.cs.vt.edu/algoviz/clip/clipping_applet.html	32
algoviz.org	30
binary heap tutorial	23
heap tutorial	19
algoviz forum	17
slr parsing example	17
villekaravirta	17
slr parsing table	16
algorithms visualization	15
binary heap animation	15
javenga	14
visualization algorithms	14
algo viz	13

Table 6.3: Top ten referring sites for the AlgoViz Portal, from 9 July 2009 to 25 July 2010.

Referring Sites	Visits Generated
wiki.algoviz.org	455
geosim.cs.vt.edu	192
webmail.sf-csiaf.unifi.it	130
sciweavers.org	108
people.cs.vt.edu	107
objectmix.com	85
moodle.fiit.stuba.sk	84
research.cs.vt.edu	57
cse.hut.fi	53
computingportal.org	51

Figure 6.1: Subversion activity at OpenAlgoViz throughout 2009.



appropriate metrics of community success; more important is the level of contribution by and the amount of interaction between community members. Of the 160 authenticated users at the Portal, only 25 have posted to the site, with six site administrators contributing the majority of the content (see Figure 6.1).

6.2 OpenAlgoViz

In contrast to the Portal, OpenAlgoViz in its current state is simply a repository of AV and AV-related program code. Fortunately, we have access to a variety of statistics, gathered by SourceForge, to measure whether it is successful.

Currently, there are ten SourceForge users with repository commit access at OpenAlgoViz. Subversion automatically tracks all activity on the Subversion repository, exposing the number of daily read and write transactions as well as how many files have been written or updated each day. Figures 6.2 and 6.2 show aggregate Subversion activity for the years 2009 and 2010, respectively. For reference, the peaks in file writes around May 2009 and June 2010 correspond to the inclusion of the JHAVÉ and JFLAP systems in the repository.

SourceForge also tracks the downloads of the packaged files available at the site as seen in Figure 6.2. Since its inception OpenAlgoViz has served 635.3 MB of data across 448 downloads. We expect SourceForge will be more than capable of handling the additional

Figure 6.2: Subversion activity at OpenAlgoViz throughout 2010.

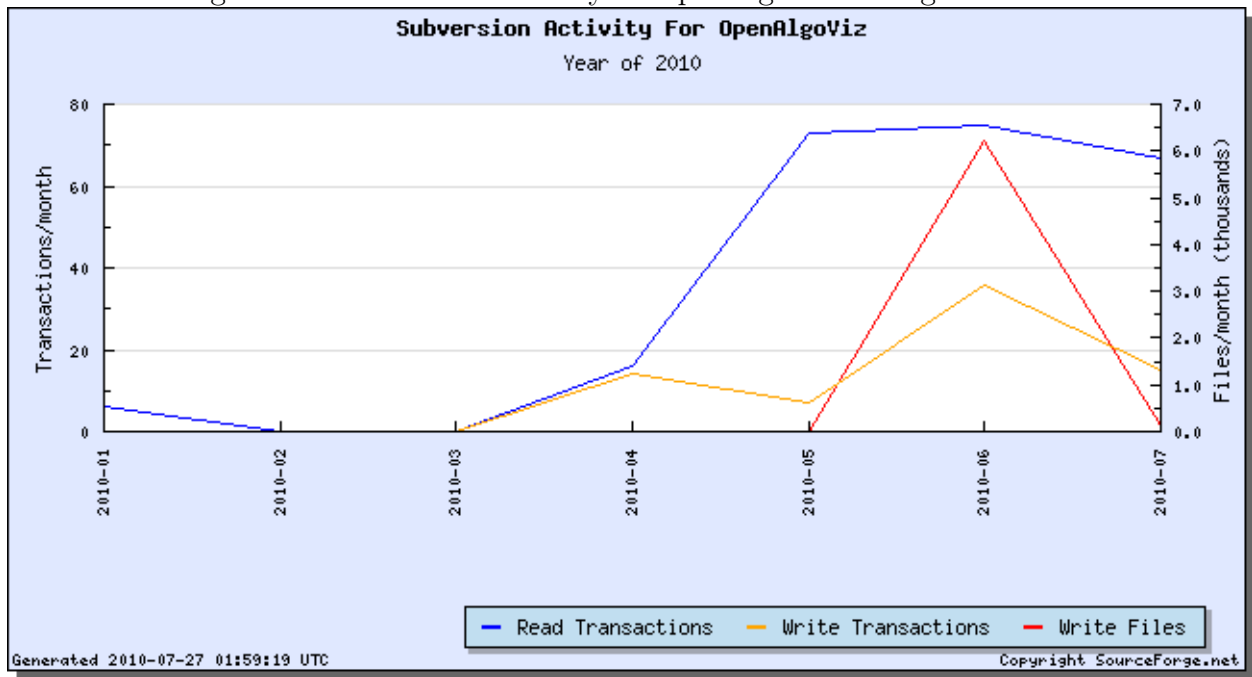
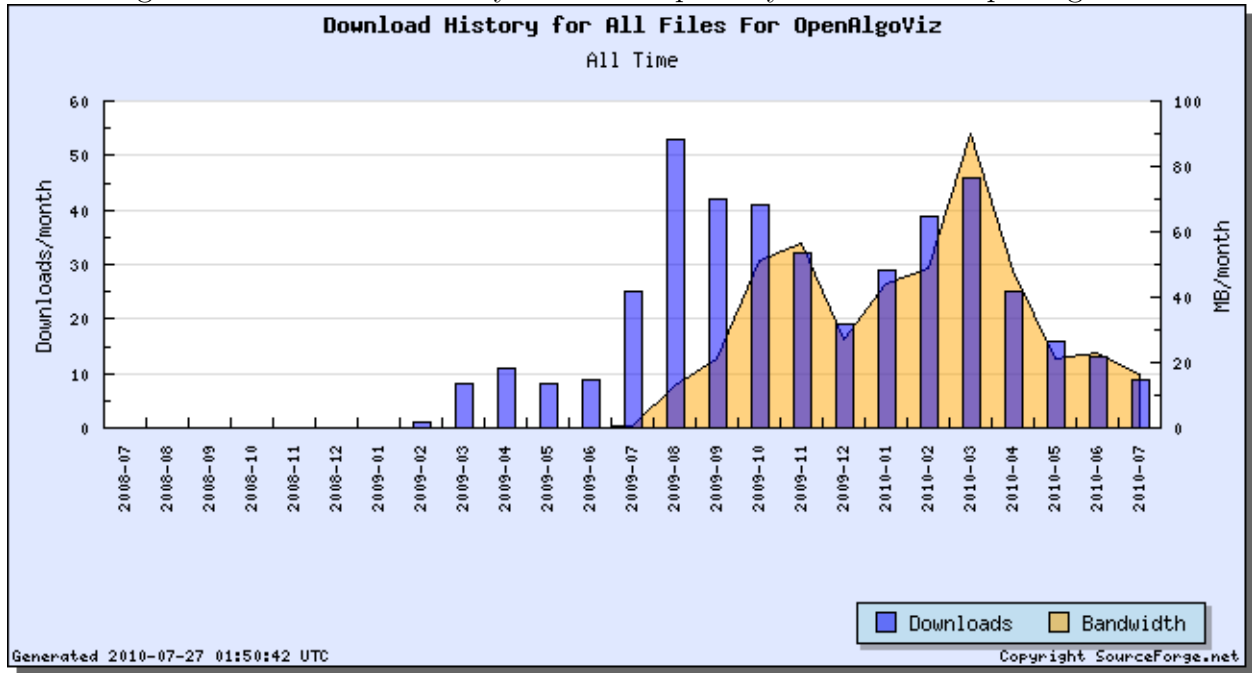


Figure 6.3: Download history for all files publicly available at OpenAlgoViz.



load of more developers contributing code and more users downloading it.

Chapter 7

Future Work

In review, we have succeeded in building the initial infrastructure of the AlgoViz Project sites. We have finished populating the Portal with existing content -- namely, the AV Catalog and Annotated Bibliography -- and established various community services, and we have begun building the OpenAlgoViz Repository. However, there is still much to be done with the AlgoViz Project. This chapter describes various avenues of future development.

Though we intend for the AlgoViz Portal to succeed the AlgoViz Wiki, some content still remains at the latter. For instance, the Wiki lists other visualization repositories, several of which are described in Chapter 2. It also provides additional information on various AV research groups and research projects. For developers there is a list of toolkits and libraries that may be used to help develop AVs. All of this content will need to be moved over before the Wiki is shut down and we move on to other tasks.

7.1 Increasing Community Participation

It is possible for the AlgoViz Portal team to continue adding new content to and maintaining existing content on the AlgoViz Project sites. However, as the AlgoViz Project is intended to build the algorithm visualization community, we must rely on the community to become involved and to contribute to the site. The community features described in Chapter 4 are only the beginning. We intend to continue our seminal community-building effort, the AlgoViz.org Awards, but we must always look for new and improved ways to solicit user contributions.

One way to incentivize user participation would be to create a system through which users are awarded points for various activities on the site. To emphasize content creation, we can assign different point values to different actions: for instance, listing new AVs in the Catalog could be worth more than posting a new topic in the forums, and writing a text review of

an AV could be worth more than merely leaving a fivestar rating. We can then assign user ranks based dependent on how many points each user has accumulated. Users could use these ranks to gauge how active a user is. This "user points" system would provide a simple way of building up social capital within the AlgoViz community.

One area in which the Portal currently falls short is the lack of means for private communication between community members. Many online communities provide facilities for Private/Direct Messaging, which allows users to discreetly contact one another. Though this may seem counter-intuitive to the goals of the AlgoViz Project, it allows users to increase slowly their interaction from one member to the whole community. Additionally, we could grant users the option to enable personal email contact forms. We expect however, that some users will be hesitant to publicize their email addresses in such a manner. Email communication may not be worthwhile for other members, who might have registered with email addresses used specifically for site registrations and not for regular communication.

One possibility related to OpenAlgoViz would be to expand it from a simple code repository into a fully-fledged open-source development community, along the lines of drupal.org [14]. In such a community, developers would be encouraged not only to contribute their own code, but to also contribute to the code of others, perhaps by fixing bugs or implementing new features. Already, OpenAlgoViz project members already have full access to the Subversion repository. An open-source AV development community could also benefit from services such as project and issue trackers as well as patch systems. It could also benefit from its own forum; perhaps the Developer's Forum at the Portal (see Section 4.1) could be moved to OpenAlgoViz. SourceForge.net provides all of these features [5]; we merely need to enable and customize them for OpenAlgoViz.

7.2 User Studies

To date we have heavily relied on the results of a variety of polls and user surveys [27, 38] to inform the creation of the Portal and of OpenAlgoViz. While development has been an iterative process, we need to push forward with formal usability testing to ensure two things: that the site provides what users want and that users can navigate the site easily and effortlessly.

These studies, which could be run both on-site as well as remotely, would likely involve having participants perform a variety of tasks whilst providing feedback both during and after the study. A list of such tasks would need to be developed that includes common and uncommon as well as simple and complex tasks. It may be handy to generate a set of user profiles, or personas, to aid in developing the tasks as well as identifying candidates for study participation.

During the studies we will need to make sure that we collect as much feedback as possible. An on-site study would undoubtedly rely on verbal feedback given via the think-aloud

protocol; such feedback could also be obtained remotely through the use of Voice-over-IP. We could utilize embedded mouse-tracking scripts (and perhaps even gaze-tracking methods) to determine where users spend their time on site pages and provide valuable data for refining the user experience. Finally, we can rely on online forms for obtaining written feedback from users. Currently, the AlgoViz Portal provides both a traditional email contact form as well as an AJAX-enabled feedback popup dialog; these interface elements have already proven useful in normal site usage.

We may be able to use Google Website Optimizer in conjunction with Google Analytics during these studies. Google Website Optimizer is a tool that tests whether changes to site content are more effective in getting conversions. It does this by dynamically changing certain pages or portions of a page and collecting the resultant usage data, compiling reports to help developers gauge the effectiveness of their websites [16].

7.3 Concluding Remarks

In review, the AlgoViz Project has produced several useful resources for developers, educators, and researchers. For it to ultimately succeed, however, the algorithm visualization community needs to come together and contribute its collective knowledge. We intend to continue developing and refining the Project artifacts, for the benefit of the community and of computer science practitioners worldwide.

Bibliography

- [1] JpGraph - PHP Graph Creating Library. <http://www.aditus.nu/jpgraph/>, 2000-2009.
- [2] Open Archives Initiative - Protocol for Metadata Harvesting - v2.0. <http://www.openarchives.org/OAI/openarchivesprotocol.html>, Jun 2002.
- [3] General Concepts | drupal.org. <http://drupal.org/node/19828>, Oct 2009.
- [4] PHP: Mysqli - Manual. <http://php.net/manual/en/book.mysqli.php>, Jan 2010.
- [5] SourceForge.net: Features. <http://sourceforge.net/register-project/features.php>, 2010.
- [6] SQLite Home Page. <http://www.sqlite.org/>, Mar 2010.
- [7] ACQUIA, INC. Drupal - Content Management System | Acquia. <http://acquia.com/>.
- [8] BAECKER, R. Sorting out sorting: A case study of software visualization for teaching computer science. In *Software Visualization: Programming as a Multimedia Experience*, J. T. Stasko, J. B. Dominique, M. H. Brown, and B. A. Price, Eds. MIT Press, 1998, ch. 6, pp. 369--381.
- [9] BRUMMUND, P., AND UTI, N. V. The Complete Collection of Algorithm Animations. <http://www.cs.hope.edu/~dershem/ccaa/ccaa/>, jun 2001.
- [10] CONNEXIONS. Connexions - Sharing Knowledge and Building Communities. <http://cnx.org/>, 2010.
- [11] COOPER, M. L. Algorithm visualization: The state of the field. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA, Apr 2007.
- [12] CRESCENZI, P., FALTIN, N., FLEISCHER, R., HUNDHAUSEN, C. D., NÄHER, S., RÖSSLING, G., STASKO, J. T., AND SUTINEN, E. The algorithm animation repository. In *Proceedings of the Second International Program Visualization Workshop, Hornstrup-Centret, Denmark* (2002), pp. 14--16.

- [13] CRESCENZI, P., AND NOCENTINI, C. Fully integrating algorithm visualization into a cs2 course.: a two-year experience. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE 2007)* (Dundee, Scotland, 2007), ACM, ACM, pp. 296--300.
- [14] DRUPAL ASSOCIATION. drupal.org | Community plumbing. <http://drupal.org/>, 2010.
- [15] EAGY, T. Algoviz Graph Creator. <http://research.cs.vt.edu/AVresearch/WikiTools/Grapher/AlgovizGraphCreator.pdf>, December 2008.
- [16] GOOGLE. Getting Started - Website Optimizer Help. <http://www.google.com/support/websiteoptimizer/bin/topic.py?topic=14307>, 2010.
- [17] GRISSOM, S. Education Links | SIGCSE. <http://www.sigcse.org/resources/links>, 2000-2010.
- [18] HART, J., AND ALBRECHT, R. Instructional repositories and referatories. *Research Bulletin, EDUCAUSE Center for Applied Research 2004*, 5 (Mar 2004).
- [19] HUNDHAUSEN, C., DOUGLAS, S. A., AND STASKO, J. T. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing 13* (Jun 2002), 259--290.
- [20] KANIA, C., AND SHAFFER, C. Union/Find Algorithm Visualization. <http://research.cs.vt.edu/AVresearch/UF/>, 2004-2009.
- [21] KARAVIRTA, V. Xaal - extensible algorithm animation language. Master's thesis, Helsinki University of Technology, Helsinki, Finland, Dec 2005.
- [22] KARAVIRTA, V. *Facilitating Algorithm Visualization Creation and Adoption in Education*. Licentiate's thesis, Helsinki University of Technology, Helsinki, Finland, Dec 2009.
- [23] KHURI, S. A user-centred approach for designing algorithm visualizations. *Informatik / Informatique, Special Issue on Visualization of Software* (Apr 2001), 12--16.
- [24] LISTER, R. The naughties in csed research: a retrospective. *ACM Inroads 1*, 1 (Mar 2010), 22--24.
- [25] MALONE, B., ATKISON, T., KOSA, M., AND HADLOCK, F. Pedagogically effective effortless algorithm visualization with a pcil. In *Proceedings of the 39th IEEE international conference on Frontiers in education conference* (Piscataway, NJ, Oct 2009), IEEE : Institute of Electrical and Electronics Engineers, IEEE Press, pp. 1501--1506.
- [26] MARICOPA CENTER FOR LEARNING AND INSTRUCTION. Maricopa Learning Exchange. <http://www.mcli.dist.maricopa.edu/mlx/>, Jan 2010.

- [27] NAPS, T. L., RÖSSLING, G., ALMSTRUM, V., DANN, W., FLEISCHER, R., HUNDHAUSEN, C., KORHONEN, A., MALMI, L., MCNALLY, M., RODGER, S., AND VELÁZQUEZ-ITURBIDE, J. A. Exploring the role of visualization and engagement in computer science education. *SIGCSE Bull.* 35, 2 (2003), 131--152.
- [28] REED, M. Member count: Not a measure of community building success | Community Building | Community Spark. <http://www.communityspark.com/member-count-not-a-measure-of-community-building-success/>, Mar 2009.
- [29] RODGER, S. H., AND FINLEY, T. W. *JFLAP: An Interactive Formal Languages and Automata Package*, 1 ed. Jones & Bartlett Publishers, Sudbury, MA, Feb 2006.
- [30] RÖSSLING, G. The Animal Animation Repository. <http://www.animal.ahrgr.de/animations.php3>, Feb 2007.
- [31] SARAIYA, P. Effective features of algorithm visualizations. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA, Jul 2002.
- [32] SARAIYA, P., SHAFFER, C. A., MCCRICKARD, D. S., AND NORTH, C. Effective features of algorithm visualizations. *ACM SIGCSE Bull.* 36, 1 (Mar 2004), 382--386.
- [33] SCHELL, G. P., AND BURNS, M. Merlot: A repository of e-learning objects for higher education. *e-Service Journal* 1, 2 (Winter 2002), 53--64.
- [34] SHAFFER, C., ALON, A. J. D., AKBAR, M., AND STEWART, M. Issues in Algorithm Visualization. http://www.youtube.com/watch?v=4nU_g-Bj-PM, Nov 2009.
- [35] SHAFFER, C. A. Interactive Hashing Tutorial. <http://research.cs.vt.edu/AVresearch/hashing/>, 2007-2009.
- [36] SHAFFER, C. A. The AlgoViz Portal: Lowering the Barriers for Entry into an Online Educational Community. <http://research.cs.vt.edu/AVresearch/Documents/NSDL.pdf>, Apr 2009.
- [37] SHAFFER, C. A., AND ALON, A. J. D. AlgoViz Project Steering Committee. <http://algoviz.org/algoviz-project-steering-committee>, Jul 2010.
- [38] SHAFFER, C. A., AND ALON, A. J. D. SIGCSE 2010 AlgoViz Survey Results. <http://algoviz.org/news/2010-03-25-sigcse-2010-algoviz-survey-results>, Mar 2010.
- [39] SHAFFER, C. A., COOPER, M., AND EDWARDS, S. H. Algorithm visualization: a report on the state of the field. *ACM SIGCSE Bulletin* 39, 1 (Mar 2007), 150--154.
- [40] THE COMPUTING AND INFORMATION TECHNOLOGY INTERACTIVE DIGITAL EDUCATIONAL LIBRARY PROJECT. CITIDEL: CITIDEL Home. <http://www.citidel.org/>, 2002--2006.

- [41] THE ENSEMBLE PROJECT. About Ensemble. <http://www.computingportal.org/site/about>.
- [42] THE JHAVE PROJECT. JHAVE: Home Page. <http://jhave.org/>, 2008.
- [43] THE PHP GROUP. PEAR - PHP Extension and Application Repository. <http://pear.php.net/>, 2001-2010.
- [44] TWITTER, INC. What Is a Direct Message? (DM). <http://help.twitter.com/entries/14606-what-is-a-direct-message-dm>, Nov 2008.
- [45] URQUIZA-FUENTES, J., AND VELÁZQUEZ-ITURBIDE, J. A. A survey of successful evaluations of program visualization and algorithm animation systems. *Trans. Comput. Educ.* 9, 2 (2009), 1--21.

Appendix A

Infrastructure View of the AlgoViz Portal

The thesis presented above explains in detail the various features and services of the AlgoViz Portal. This appendix provides a tabular summary in Table A of those features and services. It also describes in more detail the author’s actual contributions to this artifact of the AlgoViz Project, as it was truly a joint effort of several developers and contributors.

In contrast, the author was almost entirely responsible for the development of OpenAlgoViz, save the front page.

Table A.1: A broad view of the infrastructure of the AlgoViz Portal. The author was primarily responsible for the development of the bolded items, and provided major contributions to the development of the items in italics.

Collections	Community Services	Other Facets
<i>AlgoViz Catalog</i>	Forums	Site Theme
<i>Catalog Scraper</i>	Polls	Site Feedback & Contact Forms
Field Reports	Twitter Feed	
<i>Annotated Bibliography</i>	RSS Feeds	
	Email Newsletters	
	Email Notifications	
	User Registration	
	AlgoViz.org Awards	

Appendix B

Contributed Drupal Modules used at the AlgoViz Portal

The AlgoViz Portal is powered by Acquia Drupal version 1.2.26, a Drupal distribution that packages Drupal 6.17 core with several popular third-party modules. Installed modules include all those available in Acquia 1.2.26 (see <http://acquia.com/products-services/acquia-drupal-modules>), as well as the following:

- ACL
- Advanced Forum
- Author Pane
- Backup and Migrate
- Better Formats
- Biblio
- Book Access
- Charts
- CKEditor
- Corresponding Node References
- Comment Subject
- Devel
- Devel Themer
- Diff
- Emfield
- Feedback
- Flag
- Flatcomments
- Forum Access
- Graphstat

- Hide This Block
- Hierarchical Select
- HTML Comment
- JQuery Update
- Lightbox2
- Links
- LoginToboggan
- Mail Edit
- Messaging
- Migrate
- Node Reference Create
- Node Reference URL
- Notifications
- Notifications Extra
- Popup
- Popups
- Quote
- Quotes
- RPX
- Schema
- Signatures for Forums
- SimpleNews
- SimpleNews Statistics
- Site Documentation
- Statistics Advanced
- Statspro
- Tagadelic
- Tagging
- Taxonomy Redirect
- Term Node Count
- Table Wizard
- Twitter
- Typogrify
- User Stats
- User Titles
- Vertical Tabs
- Viewreference
- Views Attach
- Views Bulk Operations (VBO)

- Views Cloud
- Views Custom Field
- Views Group By
- Views Slideshow
- Visitors

Appendix C

AlgoViz.org AV Catalog Keywords

This appendix contains a list of all available keywords in the AV Catalog. The keywords were originally derived from a list of keywords used to organize the AV Catalog when it was a part of the AlgoViz Wiki (see [39]).

- 2-3-4 trees
- AA trees
- Algorithm analysis
- All-pairs shortest path problem
- Automata
- AVL trees
- B-trees
- Backtracking
- Basic Programming
- Binary search trees
- Binary search
- Binary trees
- Binomial heaps
- Clipping
- Computer graphics
- Connected components
- Dijkstra's algorithm
- Dynamic programming
- Eulerian paths
- ExpressionTrees
- finite automata
- Graph Representations

- Graph Traversals
- Greedy Algorithms
- Hashing
- Heaps
- Heapsort
- Huffman coding
- Image Processing
- insertion sort
- InterpolationSearch
- Kruskal's algorithm
- Lempel-Ziv compression
- LinearSearch
- LineReps
- Lists
- Math Algorithms
- Mealy machine
- Memory Management and Buffer Pools
- MergeSort
- Minimum spanning trees
- Moore machine
- Network Flow
- Networking
- Number Representations
- Operating Systems
- Other Compression Algorithms
- Other Sorts
- OtherSpatial
- Parallel and Distributed Processing
- Parsing
- Patricia tries
- Petri Nets
- PointRepresentations
- pushdown automata
- Prim's algorithm
- Process Management
- Program Verification
- Quadtrees
- Queues
- Quicksort

- Radix sort
- Radix trees
- RectangleReps
- Red-black trees
- RegionReps
- Run-Length Encoding
- Scapegoat trees
- Self-Organizing Lists
- Sequences
- Shell Sort
- Single-source shortest path problem
- Skip lists
- Sort Algorithm
- Sorting Networks
- Sorting Overviews
- Splay trees
- Stacks
- String Matching
- Topological Sort
- Treaps
- Tries
- Turing machine
- Union-find
- Weiler--Atherton clipping algorithm
- Algorithmic Techniques
- Compression Algorithms
- Computational geometry
- Graph Algorithms
- Linear Structures
- Miscellaneous Data Structures
- Miscellaneous Sorts
- Miscellaneous Topics
- $N \log N$ sorts
- NP-Completeness
- Numerical
- Quadratic Sorts
- Search Algorithms
- Search Structures
- Spatial Search Structures

- Systems and Languages

Appendix D

The Activity Level Field in the AlgoViz.org AV Catalog

This appendix lists the possible values for the Activity Level field, which is not described in [39].

Slide Show These AVs should not necessarily be considered algorithm visualizations, in that they do not animate the algorithm or data structure; instead, they are static presentations of content, much like static lecture slides.

Animation Only These AVs provide an animation of the content, but do not give the user the ability to control the pacing, aside from possibly a framerate control. A classic example of this would be Ronald Baecker's 1980 "Sorting Out Sorting" video [8].

Animation These AVs explicitly give the user the option to run the presentation as an animation, typically with a speed control.

Step Control These AVs split the presentation into a series of animated frames through which the user may step forward/backward, usually by pressing a next/previous button.

Canned Data These AVs visualize the algorithm or data structure using one or more input data sets that are pre-selected by the developers. Oftentimes the sets will showcase edge or exceptional behavior.

Random Data These AVs may visualize the algorithm or data structure using randomly-generated input data sets.

User Data These AVs may visualize the algorithm or data structure using input data sets supplied by the users.

Questions These AVs integrate questions for the user about the visualized algorithm or data structure, through intermittent pop-ups, post-test, associated guide questions, or other means.

Predictions These AVs ask (or require) the user to predict the behavior of the visualized algorithm or data structure. For example, the user may have to indicate where an input will be stored in a data structure before the AV reveals the actual storage location.

Exploration These AVs allow the users to explore in some way the behavior and/or performance of the visualized algorithm or data structure.

Appendix E

AlgoViz Portal AV Catalog OAI Mapping

Table E.1: The mapping of AV catalog fields to unqualified Dublin Core metadata specification and a custom specification, as used by the AV Catalog OAI-PMH data provider described in Section 3.2.3. Accurate as of 20 April 2010.

AV Catalog Field	Dublin Core Mapping	Custom Mapping
Recommendation	dc:type	wpters:recommendation
Link	dc:source	wpters:link
Delivery Method	dc:format	wpters:deliveryMethod
License	dc:rights	
Language	dc:language	
Author	dc:creator	
Institution	dc:publisher	
Project	dc:publisher	wpters:project
RelationshipToProject	dc:type	wpters:relationshipToProject
Works	dc:type	wpters:Works
Description	dc:description	
Evaluation	dc:description	wpters:evaluation
ActivityLevel	dc:type	wpters:activityLevel
GoodFor	dc:type	wpters:goodFor
Screenshots	dc:relation	wpters:screenshots
Videos	dc:relation	wpters:videos
References	dc:relation	wpters:references
First Visited	dc:date	wpters:firstVisited
Last Visited	dc:date	wpters:lastVisited
Last Updated	dc:date	wpters:lastUpdated
Topic	dc:subject	wpters:topic
Name	dc:title	
Location	dc:identifier	

Appendix F

AlgoViz Awards Voting Rubric

In the time that voting on the AlgoViz Awards was open, visitors to the AlgoViz Portal could view a voting slate listing all of the nominees. They were also presented with a set of guidelines for informing their votes; these guidelines are reproduced below:

Does the AV...

- depict the workings of an algorithm relevant to the study of CS? Consider the importance of the algorithm, the degree of difficulty students typically encounter in learning about the algorithm, and the technical correctness of the depiction.
- adapt to the knowledge level of the user? Examples might be some level of control for the level of detailed information presented.
- provide multiple views?
- include performance information about the algorithm or data structure (not just mechanics)?
- include execution history through the states of the visualization? Examples include a text window with messages at each step, and the ability to back up through prior states.
- support flexible execution control? Or is it limited to a simple animation?
- support user-input data sets?
- support interaction such as questions or user-guided progress through the visualization?
- complement visualizations with explanations?

Is the AV...

- easy to use and aesthetically pleasing? Consider clarity of controls, clarity of information content, layout, color, timing, etc.
- pedagogically sound?
- easy to interpret graphically?