

## Developing a Set of Requirements for Algorithm Animation Systems

Jean Greyling\*

Nelson Mandela University, South Africa

---

### Abstract

The learning and analysis of algorithms and algorithm concepts are challenging to students due to the abstract and conceptual nature of algorithms. Research conducted at the Nelson Mandela Metropolitan University in Port Elizabeth, South Africa focused on the design of an extensible algorithm animation framework to support the generation of interactive algorithm animations. A comprehensive literature study showed that no unified and common instrument for evaluating the pedagogic effectiveness of algorithm animation systems exists. An output of this research was the compilation of a list of requirements that could serve as such an instrument. The value of this list is proven through the evaluation of extant systems as well as in developing a prototype in order to prove the effectiveness of the animation framework.

Key Words: Algorithm Animation, Design Requirements.

---

### IJCIR Reference Format:

Jean Greyling. Developing a Set of Requirements for Algorithm Animation Systems. *International Journal of Computing and ICT Research*, Special Issue Vol. 3, No. 1, pp. 83-89. <http://www.ijcir.org/Special-Issuevolume3-number1/article9.pdf>.

---

### 1. INTRODUCTION

Algorithm animation is a form of technological support tool which encourages algorithm comprehension by visualizing algorithms in execution. Algorithm animation can potentially be utilised to support students while learning algorithms.

Despite widespread acknowledgement for the usefulness of algorithm animation in algorithm animation courses at tertiary institutions, no recognized framework exist upon which algorithm animation systems can be effectively modelled. Research conducted at the Nelson Mandela Metropolitan University in Port Elizabeth, South Africa focused on the design of an extensible algorithm animation framework to support the generation of interactive algorithm animations [Yeh, 2006]. A comprehensive literature study showed that no unified and common instrument for evaluating the pedagogic effectiveness of algorithm animation systems exists. Before designing a framework, it was thus important to compile a set of requirements for an algorithm animation system. This paper reports on the process that was followed in compiling and applying this set.

A literature study identifies features that increase the instructional value of algorithm animation systems. The identified features are organised into a set of requirements for analysing algorithm animation systems (Section 2). Selected extant algorithm animation systems are then

---

\* Author's Address: Jean Greyling. Nelson Mandela University, South Africa

"Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than IJCIR must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee."

© International Journal of Computing and ICT Research 2009.

International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online), Special Issue Vol.3, No.1 pp. 83-89, October 2009.

International Journal of Computing and ICT Research, Special Issue Vol. 3, No. 1, October 2009.

evaluated making use of these requirements (Section 3). The paper concludes with a discussion on the prototype system that was developed and how it supported the requirements (Section 4).

## 2. DESIRABLE REQUIREMENTS FOR AN ALGORITHM ANIMATION SYSTEM

The focus of algorithm animations has moved beyond merely showing students an algorithm in the hope that they will understand and retain some of the algorithm concepts illustrated. Current emphasis is placed on identifying factors which will increase the instructional value of algorithm animations [Stasko, Badre and Lewis 1993; Saraiya 2002].

This section reports on a literature study on algorithm animation features that would increase the pedagogic effectiveness of an animation system or broaden its usefulness within a teaching environment. The findings are summarized and presented as a list of requirements which can be used to measure the effectiveness of a system.

### 2.1 Requirements based on Levels of Engagement

Studies have suggested that rather than just letting students view an algorithm animation passively, better learning results may be obtained by allowing students to engage interactively with the animation [Hundhausen 2002; Naps, Fleischer, McNally et al. 2003]. Naps et al [2003] defines taxonomy of students' interaction with algorithm visualisations based on six levels of engagement: **No viewing** No algorithm visualisation is utilised.

**Viewing** Algorithm animations can be viewed as a surveillance video that records and displays the execution of an algorithm. When users investigate an algorithm, they may slow the video down to better examine a particular event, speed through events which offer no further contribution to the investigation, or step through key events one at a time.

**Responding** The system should support the activity of letting students make predictive answers by running animations in discrete steps, thus allowing the students to pause before each interesting event in the animation to predict the next algorithm action [Anderson and Naps 2000].

**Changing** Students and instructors should be allowed to input custom data into the algorithm. Instructors will thus be able to demonstrate algorithm specific characteristics to students, such as best-case and worst-case performance scenarios [Naps, Eagan and Norton 2000; Saraiya 2002].

**Constructing** Creating an animation of the algorithm under study would induce students to have a deeper understanding of the algorithm's operations, since students must learn the algorithm with the intent of sharing their understanding of the algorithm concepts to an audience [Hübscher-Younger and Narayanan 2003].

**Presenting** Presenting visualisations to other students to stimulate discussions on the given topic. The features are identified and organised based on four levels of engagement, namely viewing, changing, responding and constructing. The first and sixth level of engagement - No viewing and Presenting - are not seen as applicable in this discussion. No viewing is essentially the absence of algorithm animations. Presenting involves the learner demonstrating an algorithm animation. It is thus an activity generally performed by the instructor to aid learners.

### 2.2 Complementary Requirements

A number of additional algorithm animation system requirements are identified which are believed to enhance the pedagogic effectiveness and usefulness of the system. Each of these requirements is discussed below.

**Smooth animation** aids the student intracking changes between discrete steps of an algorithm [Stasko 1998b]. This feature forms a fundamental part of algorithm animation. In certain cases, such as when large datasets are being viewed, students should be able to disable animations and view discrete steps of the algorithm [Rößling and Naps 2002].

**Analysis features** can aid students in better understanding the efficiency of an algorithm and the relative performance differences among various algorithms [Gloor 1998]. Relative performance can be illustrated by running several algorithms simultaneously, thus letting the students compare the differences visually [Naps, Fleischer, McNally et al. 2003].

**Multiple views** of an algorithm may be used in different approaches to aid students. Students may find the use of certain metaphors easier to understand, and thus prefer a certain approach of animation [Gurka and Citrin 1996]. Different views may also be used to illustrate algorithm

executions at different levels of abstraction, or demonstrate different characteristics, such as operational or performance trends [Wilson, Katz, Ingargiola et al. 1995; Naps, Fleischer, McNally et al. 2003].

**Additional materials** accompanying algorithm animations may increase the instructional effectiveness of the animation. The materials may include simple textual explanations, pseudo-code or source code views [Rößling, Schüler and Freisleben 2000]. Alternatives include using multimedia elements, such as audio and video of instructors explaining the algorithm [Stasko, Badre and Lewis 1993].

### 2.3 List of Requirements

From the preceding discussion, a number of features are identified as the requirements for a pedagogically effective system. The identified requirements appear in Table 1. The effectiveness of an algorithm animation system to complement the students' study of algorithms is determined by the system's ability to engage the students in an active learning process (categorised as requirements R1 through R6), and system features which either provide additional information to enhance comprehensibility of the animation, or increase its usefulness in an educational environment (requirements R7 through R10).

Requirements for Algorithm Animations R1: Allow speed control of algorithm animation
R2: Allow rewinding of the animation
R3: Accept user input data for the algorithm
R4: Provide questions to predict algorithm behaviour
R5: Allow stepping control of algorithm animation
R6: Support construction of animation by students
R7: Support for smooth motion
R8: Include capabilities for comparative algorithm analysis
R9: Provide multiple views of an algorithm
R10: Provide additional instructional material

Table 1: List of identified requirements

## 3. EVALUATION OF EXTANT SYSTEMS

The compilation of a set of requirements for Algorithm Animation Systems addresses the concern that there is no unified and common instrument for evaluating the pedagogic effectiveness of algorithm animation systems. In order to highlight the value of the list, the reported research made use of the list to evaluate the following systems:

- Sorting Out Sorting
- Brown University Algorithm Simulator and Animator – BALSAs
- Generalised Algorithm Illustration through Graphical Software – GAIGS
- Java Collaborative Active Textbook – JCAT
- SAMBA/JSAMBA
- Java and Web-based Algorithm Animation – JAWAA
- A New Interactive Modeller for Animations in Lectures – ANIMAL

This paper will report on three of these evaluations.

### 3.1 Sorting Out Sorting

The “Sorting Out Sorting” video [Baecker 1998], although not strictly defined as a system, is nevertheless, worthy of mention. The video employed a number of features which were unprecedented at the time in demonstrating sorting algorithms. These features include the use of various visual metaphors, including animation, colour, audio, and voice-over commentary. The operations of nine sorting algorithms were illustrated, followed by time versus data size performance graphs typically found in textbooks. The nine algorithms were then run simultaneous in a race to compare and contrast their performance characteristics.

International Journal of Computing and ICT Research, Special Issue Vol. 3, No. 1, October 2009.

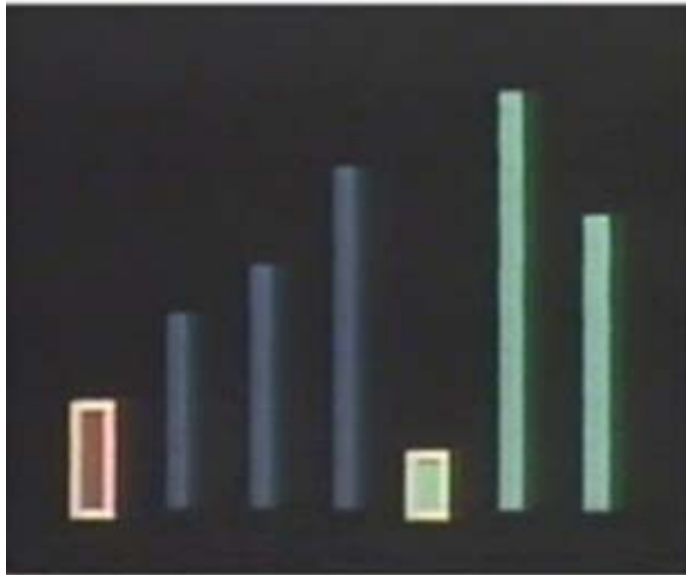


Figure 2: “Sorting Out Sorting”

The video is identified as supporting smooth motion (R7). Comparative algorithm analysis and multiple views of an algorithm are included (R8, R9). Addition material is provided through audio narratives (R10). Because the animations are produced as a video, all requirements are essentially of historical nature and thus regarded as partially supported.

### 3.2 Brown University Algorithm Simulator and Animator II (BALSA)

The BALSA animation system [Bazik, Tamassia, Reiss and van Dam 1998] can be regarded as a concept prototype for all current systems due to the novel design concepts utilised by the system. The system was designed and implemented to integrate into Brown University’s electronic classroom concept.

The system supports dynamic input to generate animation (R3), however, literature did not specify if the feature is directly accessible by students. No details were available on the level of animated motion support by BALSA. The system allows for speed control of algorithms (R1) and stepping through animations (R5). As shown in Figure 1, the system also supports capabilities to compare algorithms (R8), and show alternative animation views (R9).

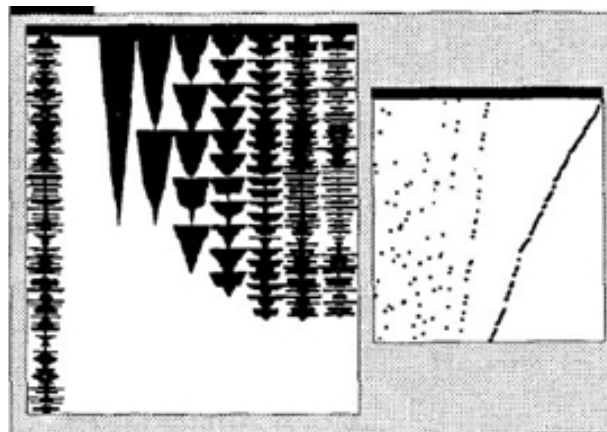


Figure 1: BALSA-II - Illustrating a mergesort using a clouds view, and a bar chart to

show consecutive states of the data.

### 3.3 Java And Web-based Algorithm Animation (JAWAA)

JAWAA [Pierson and Rodger 1998; Akingbade, Finley, Jackson et al. 2003] is an algorithm animation system which employs a scripting language. The visual objects and associated commands available in JAWAA are designed for the animation of algorithm operations, with specific support for data structure objects like arrays, stacks, queues, pointers and linked lists.

Students have speed control of animations (R1). The scripting language graphic system gives more accessibility for constructing various animations (R6). The system is implemented in a Java<sup>™</sup> applet, the embedding of static material into the client webpage is thus possible (R10). Smooth motion is supported (R7).

Evaluation of the seven systems listed showed that all the systems only satisfied a subset of the requirements reported on in Section 2. This emphasised the need for a framework that would support all these requirements.

## 4. FUNCTIONAL PROTOTYPE SYSTEM

In designing a framework for Algorithm Animation, the reported research aimed at addressing the limitation that no extant system satisfied all the requirements identified from literature. In order to demonstrate the effectiveness of the framework, a functional prototype system was developed. The prototype system was developed to animate the quadratic and  $O(N \log N)$  sorting algorithms commonly taught in introductory algorithm curricula.

User interaction with the algorithms and data are provided through a data layer interface which supports the requirement of allowing the user to input data for the algorithm (R3).

The design of the prototype centres on the concept of using a unified algorithm animation desktop, which allows the user to control the generation and display of animations. The desktop acts as a centralised placement area for animation panels, each of which presents a particular scenario. This feature of the prototype supports the requirements of speed (R1) and stepping (R5) control of algorithm animation, smooth motion (R7) as well as the construction of animation by students (R6).

The algorithm animation desktop further provides an integrated platform for comparative analyses, allowing any number of potential combinations of data lists and algorithms to be created and examined. These include comparisons of a data list using different algorithms, or different data lists using a single algorithm. When multiple scenarios are to be compared, an animation panel is created and setup for each scenario (Figure 2). The animation panel, once created, is placed within the unified desktop. Scenarios can then be individually selected for parallel display. This feature supports the requirements of comparative algorithm analysis (R8) and multiple views (R9)



Figure 2: The play control managing an animation race

Since this prototype was developed as an instrument for an existing Data Structures module within the department there was no need to develop additional instructional material (R10).

Based on the scope of the initial project, it was decided not to support rewinding of animation (R2) as well as the provision of questions to predict algorithm behaviour (R4). In a follow up project which investigated the extensibility of the designed framework, these two features were successfully added [Moruge, 2006].

## 5. CONCLUSION

Algorithm animation can potentially be utilised to support students while learning and analysing algorithms. Literature studies showed that there was no unified and commonly accepted requirements framework for evaluating the effectiveness of algorithm animation systems in an algorithm course environment. The derived list of requirements (Table) is a theoretical contribution towards identifying instructionally effective features of algorithm animation systems.

The list of requirements can be used as a foundation for the creation of a unified algorithm animation system evaluation instrument. Tertiary educational institutions can also utilise the proposed requirements as a preliminary method for evaluating the suitability of algorithm animation systems in particular course environments, with the aim of integrating the systems to complement the institution's existing teaching strategies.

## 6. REFERENCES

- AKINGBADE, A., FINLEY, T., JACKSON, D., PATEL, P. and RODGER, S.H. (2003): JAWAA: Easy Web-Based Animation from CS 0 to Advanced CS Courses. Proc. Technical Symposium on Computer Science Education, Reno, Nevada, USA. 35:162 - 166, ACM Press
- ANDERSON, J.M. and NAPS, T.L. (2000): A Context For The Assessment Of Algorithm Visualization Systems As Pedagogical Tools. Proc. First Program Visualization Workshop, Helsinki, Finland.
- BAECKER, R. and PRICE, B. (1998): The Early History of Software Visualization. In Software Visualization: Programming as a Multimedia Experience. 29 - 34. STASKO, J., DOMINGUE, J., BROWN, M.H. and PRICE, B.A. (eds). The MIT Press.
- BAZIK, J., TAMASSIA, R., REISS, S.P. and VAN DAM, A. (1998): Software Visualization in Teaching at Brown University. In Software Visualization: Programming as a Multimedia Experience. 383 - 398.
- STASKO, J., DOMINGUE, J., BROWN, M.H. and PRICE, B.A. (eds). The MIT Press.
- GLOOR, P.A. (1998): User Interface Issues For Algorithm Animation. In Software Visualization: Programming as a Multimedia Experience. 145 - 152.
- STASKO, J., DOMINGUE, J., BROWN, M.H. and PRICE, B.A. (eds). The MIT Press.

- GURKA, J.S. and CITRIN, W. (1996): Testing Effectiveness of Algorithm Animation. Proc. 1996 IEEE Symposium on Visual Languages. 182, IEEE Computer Society
- HÜBSCHER-YOUNGER, T. and NARAYANAN, N.H. (2002): Influence of Authority on Convergence in Collaborative Learning. Proc. Computer Support for Collaborative Learning Conference, Boulder, Colorado USA.
- HUNDHAUSEN, C.D. (2002): Integrating algorithm visualization technology into an undergraduate algorithms course: ethnographic studies of a social constructivist approach. Computers & Education 39(3):237 - 260. November 2002.
- MURUGE, K (2007) An Interactive Algorithm Animation Prototype. BCom Honours Dissertation. Nelson Mandela Metropolitan University, South Africa.
- NAPS, T.L., et al. (2003): Evaluating the Educational Impact of Visualization. Proc. ITiCSE 2003, Thessaloniki, Greece. ACM Press
- NAPS, T.L., EAGAN, J.R. and NORTON, L.L.(2000): JHAVE - An environment to Actively Engage Students in Web-based Algorithm Visualizations. Proc. Technical Symposium on Computer Science Education, Austin, TX, USA. 109 - 113, ACM Press