Proceedings of the 3rd
Winona Computer Science
Undergraduate Research Symposium

Sponsored by the Departments of Computer Science at Saint Mary's University and Winona State University

April 23-24, 2003
Winona, MN

Table of Contents

Steganography and the Secure Transportation of Sensitive Data

Nathan J. Polencheck
Computer Science Department
Winona State University
Winona, MN 55987
ceo@spytech-web.com

ABSTRACT

Steganography allows for the concealment of files and messages within other files. This allows for the easy transportation of sensitive data without prying eyes being able to tell a message is being transported. By using steganography techniques, observers cannot tell the difference between encoded images and their originals, as shown in the following research. This leads to the suggestion that steganography is an excellent tool for the transport of secure data over the Internet, or by physical media – especially since the original encoding media that the data was encoded into is rarely available for comparison.

KEYWORDS

Steganography, Secure Data Transport, Data Hiding

1. INTRODUCTION

Steganography is the hiding of a message within another media. The purpose of this paper is to show how steganography can be effectively used to transport sensitive data over the Internet in a secure fashion. A prime candidate for steganography is the use of an image to conceal a hidden ciphered message. Steganography improves encryption and security by creating a medium in which sensitive data can be passed through prying eyes via a file, without alerting anyone that the transmitted file actually contains a message. Steganography should be implemented more in practice today as cyber-terrorism and data-theft becomes an everyday occurrence. Steganography is a relatively old technology, but is still very young in regards to modern usage. With the high security concerns of corporations and individual users alike, the secure transportation of data needs to be a viable option. There are numerous techniques and methods for steganography, which have been highly researched and used in practice. This paper shows how well steganography performs in real-world applications in regards to the transportation of sensitive data. The following research uses encoded images and sound files which were presented to test subjects in pairs (one encoded file and its original). The subjects were then asked to analyze them and attempt to pick out the steganographic media from each pair. This study shows that, even when the original non-encoded file is available for comparison, steganographic files are not discernable by standard observation. This paper also shows how modern steganography detection techniques perform when put to the test.

2. BACKGROUND

2.1. What is Steganography?

"The concept of hiding information in other content has existed for centuries; the formal study of information hiding is called steganography." [1] Steganography is the practical science of hiding information inside other media with the intention of giving the impression that no hidden data is present. Steganography is not a new technology, having been practiced for thousands of years dating back to early mapmakers. Steganography allows a sender to embed a hidden file or message inside a cover file. A cover file is simply a file that is used to embed hidden data into. This cover file may be a graphics image, an audio file (such as a WAV or MP3 file), or even a binary executable. Cryptography has the goal of preventing the viewing of sensitive data by obfuscating the message so only the sender and recipient can view it. Steganography is intended to take cryptography to the next level by attempting to prevent the impression of the existence of any sensitive data.

Steganography's main goal is to avoid detection; to deny the existence of sensitive data inside the cover file. In the use of steganography, a cover file and hidden file are used. It is assumed that any eavesdroppers will have no access to the original cover file in question. Steganography techniques try to change the original cover file as little as possible in terms of quality and file size, in order to create the strongest security environment possible. "In steganographic applications there are two levels of security. The first is not allowing an observer to detect the presence of a secret message. The other is not allowing the attacker to read the original plain message after detecting the presence of secret information." [2]

3rd Annual Winona Computer Science Undergraduate Research Symposium, April 2003, Winona, MN.

Common media for steganography includes images (e.g.: JPEG, GIF, BITMAP, PNG), audio files (e.g.: MP3 and WAV), and even executable binaries (e.g.: EXE executables). Just about any file type that has slack or white space in it can be used for steganography – however images are usually the typical medium used for steganography purposes.

2.2. Uses of Steganography

Steganography has a wide array of uses. For example, it can be used for digital watermarking, e-commerce, and the transport of sensitive data. Digital watermarking involves embedding hidden watermarks, or identification tokens, into an image or file to show ownership. This is useful for copyrighting digital files that can be duplicated exactly with today's technologies.

E-commerce allows for an interesting use of steganography. In current e-commerce transactions, most users are protected by a username and password, with no real method of verifying that the user is the actual card holder. Biometric finger print scanning, combined with unique session IDs embedded into the fingerprint images via steganography, allow for a very secure option to open e-commerce transaction verification. [3]

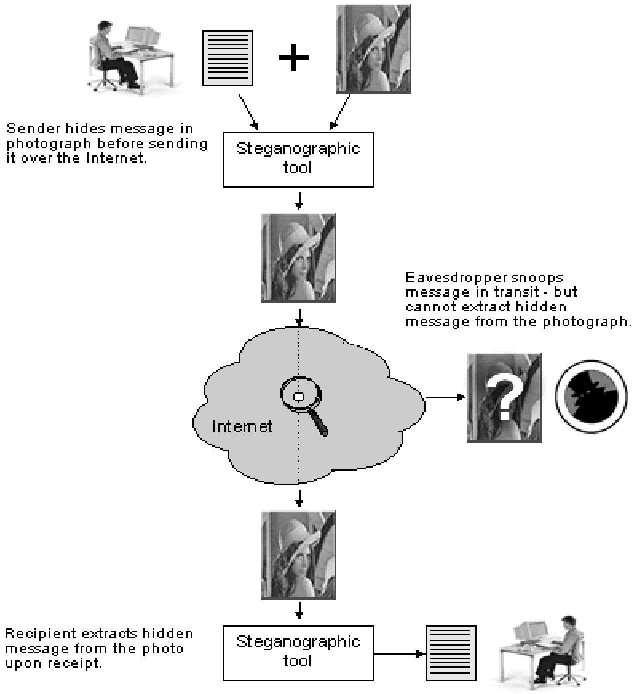

Sender hides message in photograph before sending it over the Internet.
Steganographic tool
Eavesdropper snoops message in transit - but cannot extract hidden message from the photograph.
Internet
Recipient extracts hidden message from the photo upon receipt.
Steganographic tool

Figure 1. Steganography on the Internet

The transportation of sensitive data is another key use of steganography. A potential problem with cryptography is that eavesdroppers know they have an encrypted message when they see one. Steganography allows (or tries to allow) the transport of sensitive data past eavesdroppers (see Figure 1) – without them knowing any sensitive data has passed them. The idea of using steganography in data transportation can be applied to just about any data transportation method, from E-Mail to images on Internet websites. With proper steganography techniques applied, sensitive data can be placed on public systems where only the designated recipient knows where the message is located. For example, an auction on eBay.com could be used to place a hidden, steganographic message for a specified recipient to view. However, no other browsers would have any idea the image contained a hidden message.

2.3. Implementation

Steganography can be implemented in various ways. "There are many different kinds of steganography, but all are based on finding unused space on paper, in sound, or in files in which to hide a message." [5] Many algorithms have been developed to provide robust and secure steganography – each of which uses different embedding techniques. A common technique is Least-Significant-Bit (LSB) embedding. This technique hides data bits in the last two significant bits of an image pixel. For example, a 500x500 pixel image has 250,000 pixels. If a small 249 character null-terminated ASCII message is embedded, we would need approximate 1992 bits (249 * 8 bits per character) for storage of the string. By breaking up the bit pattern for each character into pairs, we would need 4 pixels per character to store the message by storing 2 bits per pixel. This means only 1000 pixels ((249+1)*4) would need changing in their LSB (from out of a total 250,000). The "+1" in the above calculation comes from the null character used to terminate the string. This has very little effect on the overall image. [6]

Figure 2 shows the steganography process of the cover image being passed into the embedding function with the message to encode – resulting in a steganographic image containing the hidden message. A key is often used to protect the hidden message. This key is usually a password used by the decoding software to unlock the hidden message. Most steganography tools offer encryption of the hidden message before the embedding function is executed, so this key is also used to encrypt and decrypt the message before and after the embedding process.

More extensive and robust techniques are available for review, but they are beyond the scope of our intent in this paper. Many tools are available for the steganography of various media, including binary executables and MP3 audio files. In this paper we use the tools *S-Tools* and *JPHIDE*. Each of these tools is further discussed in the Methodology section of this paper. [4,9]

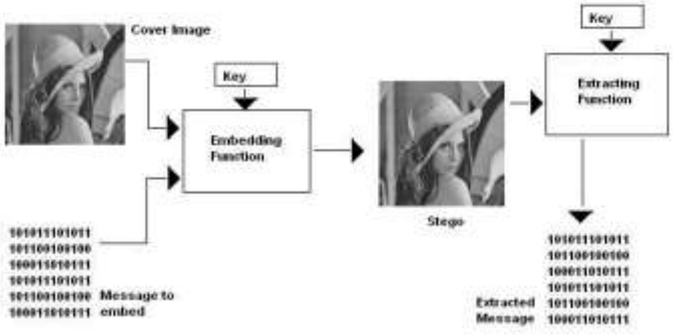


Figure 2. The steganography process [10]

2.4. Detection

The detection of steganography is a rough science, at best. Some tools are present that attempt to detect the presence of a hidden file inside other files. One such tool is *StegDetect*, written by Niels Provos. *StegDetect* attempts to detect steganographic content created by a handful of common tools. These tools include *JPHIDE, OutGuess, Invisible,* and *JSteg.*[9] A common method of detection is to analyze the least-significant bits of an image. Some steganography tools (like *JPHIDE*) leave a noticeable signature in the histogram of an image by using all of the first available LSBs to encode the hidden file (as opposed to a random selection or even distribution). *StegDetect* analyzes the histogram of a selected JPEG image to see if it has a signature created by being altered with steganography tools. This can be a daunting task – especially if the hidden file is very small relative to the overall cover file size (as discussed earlier in the brief LSB technique overview). The smaller the message that is hidden, the less impact there is on the cover file.

In a research paper written by Niels Provos, the author of *StegDetect*, an attempt was made to crawl images on eBay.com and scan them for steganographic content. Out of over two million images crawled and scanned, *StegDetect* found around 17,000 images supposedly altered with steganography – 15,000 of which were supposedly altered by *JPHIDE* (written by Allan Latham). Out of these 15,000 images, no genuine hidden messages were found. Thus leading Provos to conclude that steganography was not being used actively on the Internet - or that other tools (other than the ones detected by *StegDetect*) were being used to perform the steganography. [7]

3. PURPOSE

The purpose of this paper is to show that steganography is a strong solution for the transportation of sensitive data. When encrypted messages are too obvious for transmission, a more subtle approach needs to be taken. Steganography creates such an approach by hiding the existence of any sensitive data in the medium being transferred. Our research shows that, even when presented with the original cover images, computer users are not able to identify the steganographic file from the original. We also show that the steganography detection tool, *StegDetect*, does not identify the hidden files.

4. METHODOLOGY

To test our hypothesis that steganography provides a secure method of transporting sensitive data, we encoded a set of images and WAV files with the steganography tools *S-Tools* and *JPHIDE*. *S-Tools* is written by Andy Brown and allows for the facilitation of image (BITMAP and GIF) and WAV file steganography. "*S-Tools* applies the LSB methods discussed before to both images and audio files."[8] *S-Tools* also allows encryption and decryption of the hidden file with several different encryption algorithms. *JPHIDE* is a steganography tool written by Allan Latham that provides for JPEG steganography. Both tools support password protection of the hidden file.

The image pairs used include GIF, JPEG, and BITMAP images. A pair of WAV files was also used to test steganography in audio media. Within each pair, one file was encoded with a hidden message – the same message was used for all encoding done in this experiment. The test message is a null terminated 1868 character text message. This would require 14944 bits of storage (1868 * 8 bits per character). Using LSB steganography, 7472 pixels of storage would be needed in the cover file (see section Implementation for the arithmetic behind this number). After the files were created, they were presented to test subjects that examined the media files for any variations that may show one of the files was enhanced or modified by steganography. They then marked which image was different, or altered, or they marked a third choice signifying they could not tell any difference between the two files. The survey was given to the test subjects in the format of an HTML page – which we decided suited this research well as images are commonly transferred over the web via web-pages. The test subjects were advised to use only visual and audio feedback to analyze the file pairs – file size should not be taken into account since steganographic files in the wild cannot be compared to their original cover file's file size.

Due to time constraints, we chose to present the steganography files to thirty-four (34) test subjects. Each test consisted of four image pairs. The pairs consisted of JPEG, BITMAP, and GIF image formats of varying sizes. The BITMAP was used to represent a larger size cover file and the GIF file presented the smallest file size in the test. Two WAV file pairs were also used – with one pair representing a small file size, and the other being a much larger file. We predicted beforehand that the files with larger file sizes would be hardest to detect any alterations in – whereas smaller files would be more susceptible to visual changes, as there are less storage bits to work with. A fourth image pair was added to each test in which both images were unchanged. This provided a bias indicator to ensure test subjects were simply not guessing, and being honest if they truly did not know an answer.

After the steganographic images were tested on our test subjects, we used a tool called *StegDetect* to attempt to detect JPEG steganography. *StegDetect* is designed to find hidden messages in

JPEG images encoded with *JPHIDE* (and a few other steganography tools) so naturally a batch of encoded JPEG images (encoded by *JPHIDE*) were used. The steganographic images used in this detection test all used the same cover files (a 34KB JPEG image and a 174KB JPEG). Messages of various file lengths were encoded into the cover file each time to produce different steganographic images.

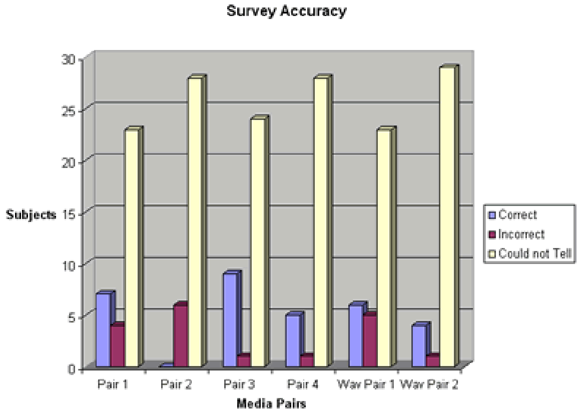
5. RESULTS

The results of this paper are broken up into two aspects: the survey results, and the detection results.

5.1. Survey Results

The results, shown in Table 1, are broken up by pairs, and arranged according to test subjects' answers. Pair 1 indicates the JPEG image pair (34KB image), Pair 2 indicates the JPEG used as a bias indicator, Pair 3 indicates the GIF image pair (7KB image), and Pair 4 indicates the BITMAP image par. Wav Pair 1 indicates the smaller WAV file used (31KB) and Wav Pair 2 indicates the larger WAV file (229KB).

The results are easily read (as seen in Graph 1) – steganographic images are hard to detect, even when the original is given for comparison. Even the low quality, small GIF file used was only selected correctly by nine viewers (out of 34). The much larger BITMAP pair only resulted in five correct responses. We postulated earlier that the larger files would lead to lower detection rates – and that proved valid. The large BITMAP image used (Pair 4) yielded the highest success rate amongst the image pairs. The larger WAV file (Wav Pair 2) yielded the highest success rate as well. The smallest file (Pair 3) was easiest to be detected, given that the message that was hidden was approximately 1/7th the total file size. This also supported our earlier prediction. Our bias indicator proved to work very well – only 6 subjects incorrectly guessed that it was altered (instead of choosing the correct "Do not know" answer).



Graph 1. Graph version of survey results

After subjects answered the survey, they were asked how they evaluated the images. Unfortunately, a few subjects mentioned they use file size as an indicator – which we stated earlier should not be done. This shows that the success rates we have accumulated may have been much higher if these results were filtered out prior to analysis of the survey results. Other subjects who achieved some correct answers without "cheating" shared a common trait – they all worked in the graphics design industry. This leads us to believe that a trained eye could help detect steganographic images – but only when they are able to be compared to the original cover file.

Overall, our findings support our hypothesis that steganography is a strong solution for the transfer of sensitive data. Even when the original cover file was available for comparison, steganographic images could hardly be determined from their original counterparts.

5.2. Detection Results

The detection results provided an interesting picture, which also supports our idea that bigger is better in regards to cover file size. The results of our steganography detection are shown in Table 2. Our steganography detection tool, *StegDetect*, managed to detect some steganographic images, but not many. We embedded various messages ranging from 220 characters to 6KB in the 34KB JPEG image. *StegDetect* was unable to detect any messages in the altered files with its default settings. When *StegDetect's* sensitivity was increased (a feature that allows *StegDetect* to be "more accurate") nearly all steganographic images were detected – minus the first 220 character file. This file could not be detected regardless of the sensitivity settings in *StegDetect*.

We then tested *StegDetect* on various steganographic images built from the 174KB JPEG cover file. *StegDetect* enjoyed much less success on this test run – only one of the six steganographic images generated was detected. Message sizes used in this run ranged from a 220 character text message to a 16KB image. The file detected was the steganographic image with a very large hidden message – the largest *JPHIDE* could hide within the cover file (the 16KB image). Regardless of sensitivity settings, *StegDetect* could not find the other steganographic images with hidden messages of varying sizes (including a file that was 11KB in size).

This leads us to believe that *StegDetect* works somewhat reliably on smaller images, but not on larger images – especially those with small hidden messages. To bypass *StegDetect*, only a small hidden message in a large cover file is needed. *StegDetect* only supports JPEG images and four steganography encoding programs (all of which only use JPEG cover files). Given that *StegDetect* was the only testable software we could find, we conclude that steganography detection is very primitive in software form – and that it can be bypassed by simply using GIFS, BITMAPS, WAVS, and other media for steganography cover files.

6. CONCLUSION

Steganography is a powerful technique of hiding data in other files. This paper briefly covers the background, implementation, uses, and detection of steganography. There are many tools, like *S-Tools* and *JPHIDE*, that can be used by computer users to easily enhance their security with steganography. With steganography, sensitive data can be transported securely over the Internet and other mediums, without tipping off eavesdroppers and hackers.

Our research, on a small number of test subjects (due to time restrictions) shows that steganography effectively encodes hidden messages in media files without the viewer being able to notice – even if they have the original cover file to compare it to. This research concludes that steganography is effective at hiding hidden messages without altering the cover file noticeably. Our research also shows how unreliable and "young" steganography detection tools are. We were only able to find one worthy steganography detection tool to test (*StegDetect*) – and it did not perform well in our test. *StegDetect* showed marginally accurate detection in small images with large encoded hidden messages, but failed in nearly all tests on larger file sets.

After completing this research, we can choose to further extend our steganography research in a few different ways. Research could be performed to accurately find a plane of reliability with common steganography tools. In other words, we would determine exactly how large hidden messages can be in relation to their cover files without visually altering the cover file during the encoding process. Attention could be given to the improvement of steganography detection, as we have shown the programmatic detection of steganographic files is very weak. Lastly, we would like to enhance security aspects of steganography tools to counter any detection techniques that may be presented in the future. Now that we have a solid idea of how steganography is implemented (and sometimes detected) we could improve steganography algorithms even more.

7. REFERENCES

[1] Acken, J. How watermarking adds value to digital content. *Communications of the ACM,* 41, 7, 75-77, 1998.

[2] Alturki, F. and Mersereau, R. Secure high data embedding rate for steganographic application. In *Proceedings of the 2000 ACM workshops on Multimedia*, Los Angeles, California, November 2000, ACM Press, New York, NY, 131-134.

[3] Bolle, R., Connel, J., and Ratha, N. Secure data hiding in wavelet compressed fingerprint images. In *Proceedings of the 2000 ACM workshops on Multimedia,* Los Angeles, California, November 2000, ACM Press, New York, NY, 127-130.

[4] Brown, A. S-Tools v4. Available from: http://members.tripod.com/steganography/stego/s-tools4.html. Access on 2003 March 30.

[5] Castelluccio, M. Hidden writing and national security.(right to privacy and steganography). *Strategic Finance* 2001,83,5,59.

[6] Chang, L., Longdon, G., and Moskowitz, I. A new paradigm hidden in steganography. In *Proceedings of the 2000 workshop on new security paradigms,* Ballycotton, Country Cork, Ireland, 2000, ACM Press, New York, NY, 41-50.

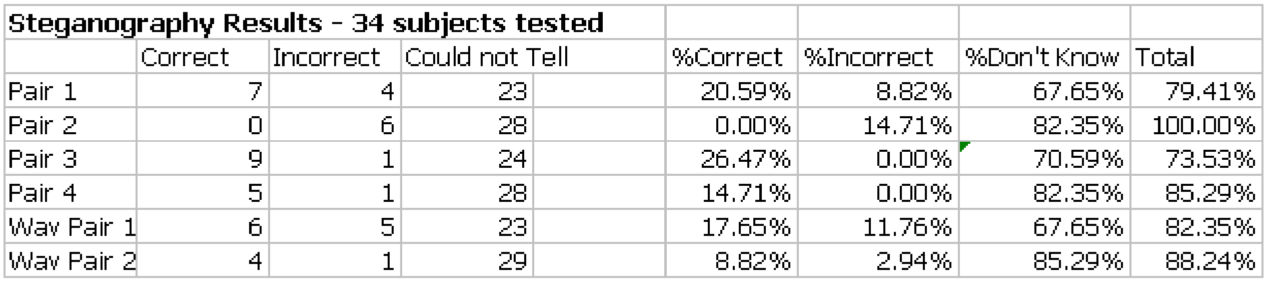
[7] Honeyman,, P., Provos, N. Detecting steganographic content on the Internet. Aug. 2001. Available from: http://www.citi.umich.edu/techreports/reports/citi-tr-01-11.pdf. Access on 2003 March 30.

[8] Johnson, N. Steganography. 1995. Available from: http://www.jjtc.com/stegdoc/steg1995.html. Accessed on 2003 Feb 08.

[9] Provos, N. Steganography detection with stegdetect. Available from: http://www.outguess.org/.

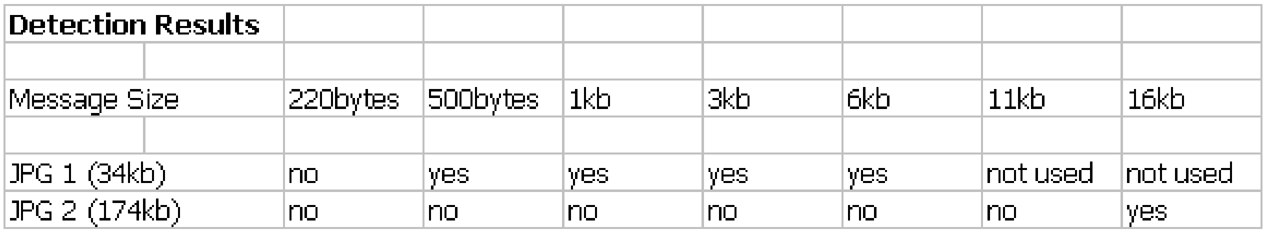
[10] Tran, T., Steganography: the art of hiding data. 2002. Available from: http://www.mills.edu/ACAD_INFO/MCS/CS/S02MCS125/Steganography.htm. Accessed on 2003 Feb 10.

8. APPENDIX



Steganography Results - 34 subjects tested								
	Correct	Incorrect	Could not Tell		%Correct	%Incorrect	%Don't Know	Total
Pair 1	7	4	23		20.59%	8.82%	67.65%	79.41%
Pair 2	0	6	28		0.00%	14.71%	82.35%	100.00%
Pair 3	9	1	24		26.47%	0.00%	70.59%	73.53%
Pair 4	5	1	28		14.71%	0.00%	82.35%	85.29%
Wav Pair 1	6	5	23		17.65%	11.76%	67.65%	82.35%
Wav Pair 2	4	1	29		8.82%	2.94%	85.29%	88.24%

Table 1. The survey results of 34 test subjects



Detection Results							
Message Size	220bytes	500bytes	1kb	3kb	6kb	11kb	16kb
JPG 1 (34kb)	no	yes	yes	yes	yes	not used	not used
JPG 2 (174kb)	no	no	no	no	no	no	yes

Table 2. *StegDetect* Detection Results

Comparison of Ray Tracing and Shadow Mapping Execution Times for Shadow Rendering

Michele O'Brien
Computer Science Department
St. Mary's University of
Minnesota
Winona, MN 55987

ABSTRACT

In computer graphics, many different methods exist for adding shadows to a scene in order to enhance the realism of the scene. Shadow mapping and ray tracing are two of the most widely used methods for rendering shadows. While the ray tracing algorithm can create more realistic shadows than the shadow mapping algorithm, it requires more computation time than the shadow mapping algorithm in rendering shadows. In this paper, the shadow mapping and ray tracing techniques were used to render shadows in scenes containing two to six three-dimensional shapes and one to four light sources. Rendering times were examined to determine to what extent using shadow-mapped shadows versus ray-traced shadows reduces the rendering times for these simple scenes. Results indicated that on average, using shadow-mapped shadows reduced the rendering times by about 31.4%. The results of this experiment also demonstrate how the rendering times were affected with the addition of more objects and light sources to a scene. As more lights and shapes were added to the scenes, the rendering times for both algorithms increased. Varying amounts of shadow rays were also used in the ray tracing algorithm, and it was determined that increased amounts of shadow rays significantly increased the ray tracing algorithm's rendering times.

General Terms

Shadow mapping, Ray tracing

1. INTRODUCTION

One of the goals of computer graphics is to produce life-like images. Adding shadows to a scene makes it look much more realistic because shadows allow the viewer to discern the positions of the objects as well as the positions of the light sources in the scene. While there are many algorithms that can be used for rendering shadows, two of the most commonly used ones are shadow mapping and ray tracing.

Shadow mapping utilizes a depth buffer to store the depths of each pixel in relation to every light source in a scene. The shadow mapping algorithm is not affected by scene complexity but is memory intensive. Ray tracing involves sending out rays from the observer's eye position to each object in the scene, and then to each light source. Ray tracing is computationally intensive but can generate more realistic shadows than the shadow mapping algorithm.

It is well known that the ray tracing algorithm takes longer than the shadow mapping algorithm in rendering shadows due to its computational intensity. Information on how *much* longer the ray tracing algorithm takes in rendering shadows, however, is scarce. This paper determines if the ray tracing algorithm takes a significantly longer amount of time than the shadow mapping algorithm in rendering shadows in scenes that contain only a few shapes and light sources. An insignificant difference in rendering times may greatly influence one to use ray-traced shadows for increased realism, rather than shadow-mapped shadows. This paper also determines how rendering times are affected as more shapes and light sources are added to a scene. If the rendering times for each algorithm increase at different rates, then that information would be useful in determining which shadow rendering algorithm to use in more complex scenes.

1.2 SHADOW MAPPING

The shadow mapping technique was first developed by Lance Williams in 1978 [1]. William's algorithm consists of two steps:

1. Render the scene from the point of view of the light source [2]. 3-D scenes are created in XYZ coordinates, with the Z coordinate representing the depth of a pixel. The Z-value (depth) of each pixel is then stored in a frame buffer (also called a depth buffer), which is "memory that contains a complete digital picture" of the scene [2].
2. Render the scene from the point of view of the observer's eye [2]. If a pixel has a depth that is greater than the corresponding depth stored in the frame buffer, then that pixel is shadowed [3].

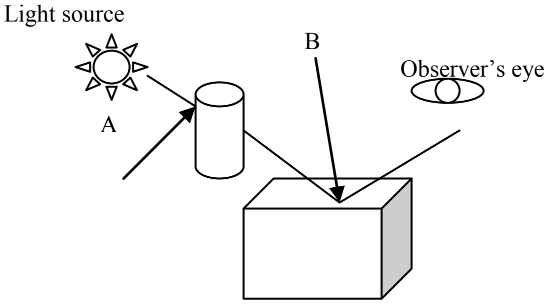


Figure 1.2.1 [3] Shadow mapping example

3rd Annual Winona Computer Science Undergraduate Research Symposium, April 2003, Winona, MN.

For example, in Figure 1.2.1, point B is further from the light source than point A. When the scene is rendered from the observer's eye, the depth value for point B will be greater than the depth value stored in the frame buffer for point A. Therefore, point B will be shadowed.

Some advantages to shadow mapping are that it is easy to understand, it is not affected by the complexity of a scene, and it requires no knowledge of the geometry of the scene in order to create shadows [4]. Also, the depth buffer only has to be recalculated if the objects move in relation to the light sources, not if the observer's eye moves. Shadow mapping is memory intensive, however, and if multiple light sources are used, each light source requires a depth buffer [3]. It is also affected by the screen resolution. A higher the screen resolution means that more pixel depth values must be stored in memory.

1.3 RAY TRACING

Whitted first proposed that ray tracing could be used to shade objects in a scene in 1980, and he also observed that such techniques created more realistic images [5]. Ray tracing is a recursive algorithm in which many rays are extended out from the observer's eye. If the ray intersects with an object, more "feeler" rays are extended from the point of intersection towards the light sources. If the feeler is blocked by a second object, then the original intersection point is shadowed [6].

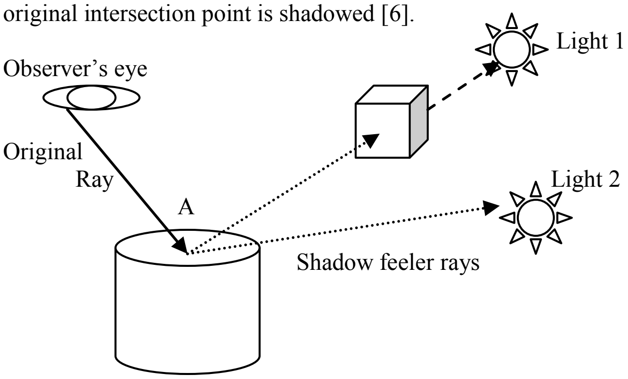


Figure 1.3.1 [1] Ray Tracing Example

Figure 1.3.1 illustrates how the ray tracing algorithm works. The feeler ray from point A to Light 1 is blocked by the cube, therefore point A will be shadowed in relation to Light 1. Point A will not be shadowed in relation to Light 2, however, because the feeler ray from point A to Light 2 is not blocked by any other objects.

The main advantage of using ray tracing to render shadows is that it creates more realistic shadows than is possible with shadow mapping. The shadows will appear less jagged as more shadow feeler rays are sent out from intersection points. Ray tracing also requires less memory than shadow mapping. The major disadvantage is that ray tracing requires more processing time in order to calculate the intersection points of rays with objects [7]. Whitted observed that his ray tracing program spent 75% of its execution time in calculating intersections [5]. Unlike shadow mapping, ray intersections must be recalculated when either the observer's eye moves, or the objects in the scene move, making it impractical for use with animation [8]. Ray tracing also depends on the complexity of a scene because more intersection points must be calculated when there are more objects in the scene.

1.4 BACKGROUND RESEARCH

Many efforts have been made to improve both shadow mapping rendering times and ray tracing rendering times. Rendering times and shadow quality can be decreased primarily through improved hardware. "Hardware accelerated shadow mapping is available on GeForce3 GPUs," (Graphics Processing Unit) and is "also exposed in OpenGL and Direct3D 8," [4].

Many variations of the ray tracing algorithm have also been created in order to increase its computational speed. Glassner notes that there are three strategies for improving ray tracing times: reducing intersection times, producing fewer rays, and "replacing individual rays with a more general entity" [6]. To reduce intersection times, either intersections must be calculated in less time or there must be fewer intersections to calculate [6].

Weghorst, Hooper, and Greenberg [8], for example, reduced the number of intersections by using bounding volumes. Bounding volumes are simple, three-dimensional shapes such as a sphere or cube that enclose a more complex object. If a ray does not intersect with the bounding volume, then an intersection test does not have to be performed for the object inside the volume. Computing an intersection with the simple bounding volume requires less time than computing an intersection with a more complex object.

Since ray tracing produces more realistic shadows, efforts continue to be made in order to make its rendering times comparable to those of the shadow mapping algorithm. Whether or not ray tracing will ever completely replace shadow mapping is yet to be determined. "Increased CPU performance and dedicated ray-tracing hardware" also help to decrease ray tracing times [10]. According to several members of a panel of computer graphics experts, ray tracing will probably not replace shadow mapping [10]. Instead, new techniques will be developed that contain the strengths of both algorithms [10].

1.5 HYPOTHESES

The time it takes to render a scene using the shadow mapping algorithm to create shadows will be significantly less than the time it takes to render the same scene using the ray tracing algorithm to create shadows. Also, the rendering times of both algorithms will increase as more light sources are added to the scene, but only the rendering times of the ray tracing algorithm will increase as more objects are added to the scene. Finally, increasing the amount of shadow feeler rays sent out from an intersection will also increase the rendering times of the ray tracing algorithm. Rendering times with a difference of less than .01 seconds are considered to be equal.

2. METHODS

The Maya Personal Learning Edition is a 3-D animation software package that also contains 3-D modeling, lighting, and rendering tools. Maya supports shadow mapping by means of its depth map shadow rendering option, as well as the basic ray tracing method used for shadow rendering [11]. This software was used in rendering scenes containing one through four light sources and two through six shapes. The scenes were first rendered without shadows, then rendered with depth map shadows, and finally, were rendered with ray-traced shadows. The ray traced shadows were also rendered using different amounts of shadow rays.

Rendering times were then compared in order to test the hypotheses.

2.1 SETTING UP THE SCENES

Each scene consisted of 3-D shapes, specifically a sphere, a cube, a torus, a cone, a cylinder, and wine glass, as well as 3 planes to serve as the walls and floor. Figure 2.1.1 illustrates the scene containing all six shapes. The writing across the picture is simply a watermark used in preventing commercial use of the software.

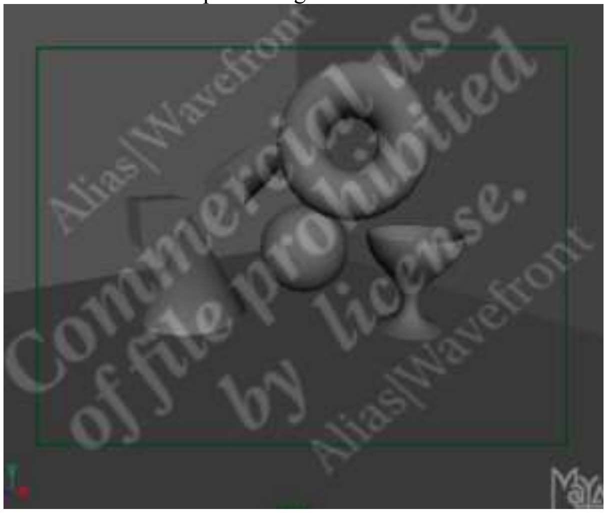


Figure 2.1.1 Scene containing all six of the 3-D shapes

Spot lights were used for the light sources, and the scenes contained from 1 to 4 light sources. Therefore, 12 scenes were be tested. They consisted of:

- 2 of the 3-D shapes listed above with 1, 2, 3, and then 4 light sources
- 4 of the 3-D shapes listed above with 1, 2, 3, and then 4 light sources
- 6 of the 3-D shapes listed above with 1, 2, 3, and then 4 light sources

2.2 RENDERING THE SCENES

All scenes were rendered in a 640 X 480 (pixels) window. Preliminary rendering tests were performed, and rendering times for the shadow mapping algorithm were all within .1 seconds of each other. The same was true for the ray tracing rendering times. Therefore each scene was only rendered 10 times using the shadow mapping algorithm to render the shadows. This was accomplished by turning on Maya's depth map shadow option and by turning off the ray tracing shadow option. Also, only one depth map will be used per light source. Figure 2.2.1 provides an example of a scene containing four shapes and three light sources that was rendered with depth map shadows.

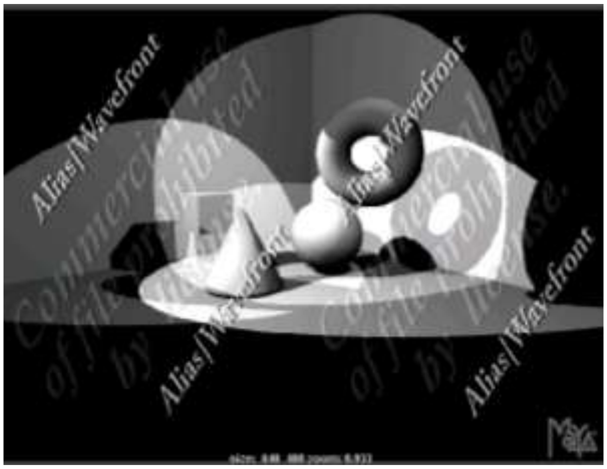


Figure 2.2.1 Example of a rendered scene

Next, each scene was rendered 10 times using the ray tracing algorithm with the amount of shadow rays set to 1, then 10 times with the amount of shadow rays set to 15, and then 10 more times with the amount of shadow rays set to 30 (Maya permits ranges from 1-40 shadow rays). For this, the depth map shadow option was turned off, and the ray tracing option was turned on.

The first light source in the scene had an intensity of two in order to make the objects in the scene more visible, whereas any additional light sources had an intensity of one. The cone angle for each light was set to 60 degrees. Default values were used for all other settings. The standard render option was used versus the IPR render since the standard render has the ability to render ray traced shadows. The rendering times were recorded in seconds for all trials. The rendering times were determined using the Maya Embeded Language timerX command, which is "used to calculate elapsed time" and "returns sub-second accurate time values" [11].

3. RESULTS AND ANALYSIS

Scenes containing depth map shadows took longer to render than scenes containing no shadows for all trials. The scenes containing the depth map shadows also took less time to render than scenes containing ray-traced shadows for all trials. Figure 3.1.1 illustrates the average rendering times for the scene containing six shapes with no shadows, depth map shadows, and ray-traced shadows using one shadow ray.

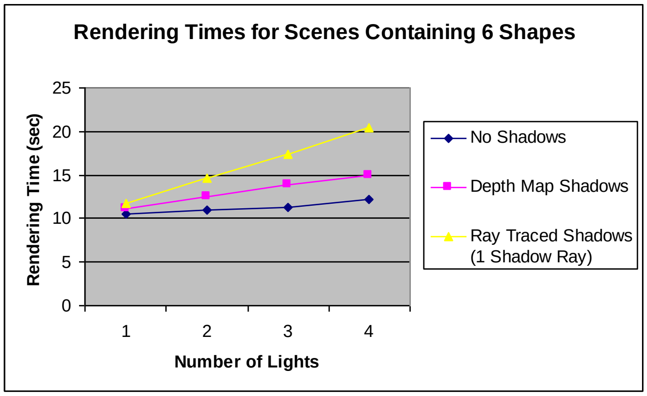


Figure 3.1.1 Rendering Times

In the scene containing six shapes and one light source, the ray tracing rendering times, on average, were only 6% higher than the shadow mapping rendering times. As more light sources were added to the scene, the ray tracing rendering times grew to be 45% higher than the shadow mapping rendering times when the fourth light was added. The difference between the shadow map and ray tracing rendering times also increased as more light sources were added to the scene with two shapes, but not when more light sources were added to the scene with four shapes.

As illustrated by Figure 3.1.1, in the scene containing six shapes, both the shadow mapping and ray tracing rendering times increased as more light sources were added to the scene. The ray tracing rendering times increased at a higher rate than the depth map rendering times, however. This was also the trend for the scene containing two shapes. In the scene containing four shapes, the average ray tracing rendering time decreased by 7% when the fourth light was added to the scene.

The average rendering times for the scenes containing depth map shadows increased as more shapes were added to the scene. The rendering times for the scenes containing ray-traced shadows did not always increase as more shapes were added to the scene, however, as illustrated in figure 3.1.2. This was true for all of the scenes containing ray-traced shadows.

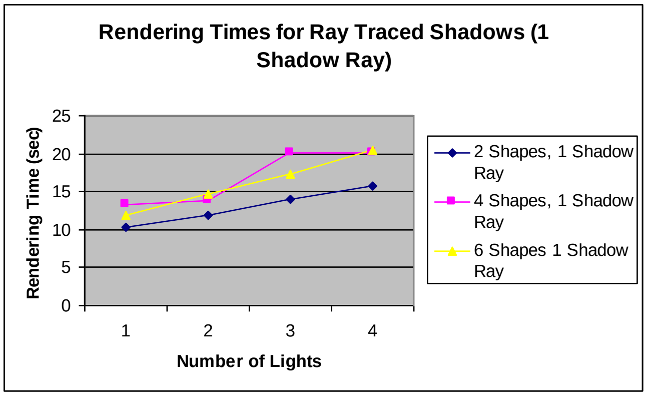


Figure 3.1.2 Rendering times of ray traced shadows (1 shadow ray)

Increasing the amount of shadow rays used in rendering ray-traced shadows also greatly increased rendering times. Figure 3.1.3 illustrates the average rendering times for scenes containing six shapes and ray-traced shadows with 1, 15, and 30 shadow rays. The results were similar for the scenes containing two and four shapes. In the scene containing six shapes and one light source, for example, the ray-traced shadows using 30 rays took about one minute longer to render than the ray-traced shadows using only one shadow ray. As the number of light sources increased, the difference between the rendering times for scenes with ray-traced shadows using one and 30 shadow rays increased by as much as 3 times when the fourth light was added.

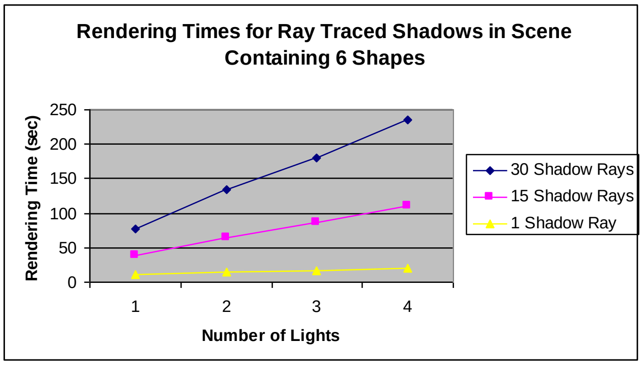


Figure 3.1.3 Rendering times of ray traced shadows

Table 3.1.1 shows the average increase in rendering time, in seconds, for each light and shape that is added to a scene for scenes rendered with no shadows, depth map shadows, and ray-traced shadows.

	Avg. Time gained by adding 1 Light Source (sec)	**Avg. Time gained by adding 1 Shape (sec)**
No Shadows	.4	.133
Depth Map Shadows	.864	.838
Ray-Traced Shadows (1 Shadow Ray)	1.739	.779
Ray-Traced Shadows (15 Shadow Rays)	15.239	6.544
Ray-Traced Shadows (30 Shadow Rays)	34.617	13.788

Table 3.1.1 Time gained by adding shapes and lights

As shown in Table 3.1.1, the average time gained by adding a light source to a scene containing ray-traced shadows (one shadow ray) is not quite one second greater than the average time gained by adding a light source to a scene containing depth map shadows. Also, adding a shape to a scene containing ray-traced shadows (one shadow ray) requires less rendering time, on average, than adding a shape to a scene containing depth map shadows.

4. CONCLUSION

The ray tracing algorithm (one shadow ray) for rendering shadows takes on average 31.4% longer to render shadows than does the shadow mapping algorithm in simple scenes containing two through six shapes and one through four light sources. This is because much time is spent in calculating ray-object intersections for the ray tracing algorithm, whereas in the shadow mapping algorithm, only pixel depths need to be compared. Figure 4.1.1 provides a summary of average rendering times for scenes containing six shapes and no shadows, depth map shadows, and ray-traced shadows.

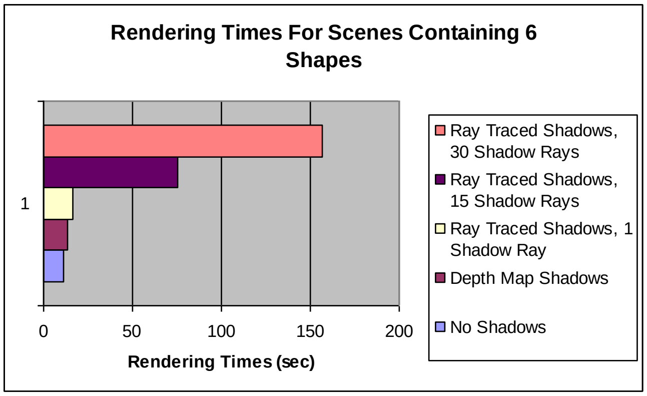


Figure 4.1.1 Average Rendering Times

In the scenes containing only two shapes and one light source, the rendering times for the ray traced shadows (one shadow ray) and shadow mapped shadows were almost identical. As more shapes and lights were added to the scenes, however, the difference in the rendering times of the two algorithms became more significant. Therefore, in very simple scenes containing only a few shapes and one light source, using ray traced shadows may provide added realism without a high increase in rendering time.

As more lights and shapes are added to the scenes, the rendering times for both the ray tracing algorithms and the shadow mapping algorithm generally increase. As the amount of shadow rays sent towards the light sources decreases, the rendering time of the ray tracing algorithm also decreases. This provides one simple method of optimizing the ray tracing algorithm.

More complex scenes containing more shapes and light sources should also be tested to see if these trends continue. If rendering times are of no concern, other factors such as memory usage or the realism of the shadows may be considered in choosing one algorithm over the other. Therefore, the amount of memory used by both the ray tracing and shadow mapping algorithms as well as the realism of the shadows produced by each algorithm could also be studied.

5. REFERENCES

[1] Foley, Van Dam, Feiner, Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Boston, MA, 1996.

[2] Williams, Lance. Casting Curved Shadows on Curved Surfaces. *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques.* ACM Press, New York, NY, 1978. Available from: http://portal.acm.org Accessed 2003 Mar 6.

[3] Hill, F.S. Jr. *Computer Graphics Using OpenGL*. Prentice Hall, Upper Saddle River, NJ, 2001.

[4] Everitt, C., Rage, A., Cebenoyan, C. Hardware Shadow Mapping. Available from: http://developer.nvidia.com/docs/IO/1830/ATT/shadow_mapping.doc. Accessed 2003 Feb 12.

[5] Whitted, Turner. An Improved Illumination Model for Shaded Display. *Communications of the ACM* 23, No 6, 1980. Available from: http://portal.acm.org Accessed 2003 Feb 12.

[6] Glassner, Andrew. *An Introduction to Ray Tracing*. Academic Press, San Diego, CA, 1989.

[7] Wald, I., Slusallek, P., Benthin, C., Wagner, M. Interactive Rendering with Coherent Ray Tracing. *Eurographics* 20, No 3, 2001. Available from http://graphics.cs.uni-sb.de/~wald/Publications/EG2001_IRCRT/InteractiveRenderingWithCoherentRayTracing.pdf Accessed 2003 Feb 11.

[8] Weghorst, H., Hooper, G., Greenberg, D.P. Improved Computational Methods for Ray Tracing. *ACM Transactions on Graphics* 3, No 1, 52-69, 1984. Available from: http://portal.acm.org Accessed 2003 Feb 12.

[9] Meaney, D., O'Sullivan, C. Heuristical Real-time Shadows. Available http://www.cs.tcd.ie/publications/tech-reports/reports.99/TCD-CS-1999-19.pdf Accessed 2003 March 7.

[10] Grantham, Brad. When Will Ray-Tracing Replace Rasterization. SIGGRAPH Panel Discussion. San Antonio, TX, 2002 July 21.

[11] Maya Personal Learning Edition Software. Available http://www.aliaswavefront.com. Accessed 2003 March 30.

Design of a Voice-Aware Firewall Architecture

Justin S. Cook
Department of Computer Science @
Winona State University
jscook2345@webmail.winona.edu

Abstract

Voice Over IP (VoIP) is the concept of multimedia communications occurring between hosts on an Internet Protocol (IP) based network. VoIP has grown more and more as companies have been building on their IP infrastructure. VoIP strives to imitate the services given by the common telephone infrastructure we use everyday. Communication done by telephone falls into a peer-to-peer topological communications model and it is this peer-to-peer nature that is the source of many of the problems we are facing. These problems are experienced when VoIP is communicating across networks segmented by a Network Address Translation component, commonly found in today's firewall architectures. We believe that by designing an infrastructure of compatibility around that of a common firewall architecture, we can bring a network wide solution to problems experienced with VoIP.

General Terms

Design

Keywords

Voice over IP, VoIP, H.323, Gatekeeper, Firewall, Network Address Translation, NAT, Application Level Gateway, ALG, Connection Tracking.

1. Introduction

1.1. Voice over IP

Voice Over IP (VoIP) is the concept of multimedia communications occurring between hosts on an Internet Protocol (IP) based network. IP networks dominate today's corporate desktops and include many of today's network topology technologies [1]. Using VoIP in place of the common telephone infrastructure has number of distinct advantages [2,3,4]:

- **Internetworking**: VoIP can bridge multimedia communications between packet-based and switched-circuited networks. Standards are in place to provide VoIP gateways to phones, fax machines, and a number of different devices.
- **Simplification**: A single infrastructure that supports all forms of communication allows more standardization and less equipment management.
- **Long Distance Cost Reduction:** Placing calls using the Internet can reduce long distance calling costs.
- **Make Use of Infrastructure:** Companies with a large IP-based network structure can make use of it and avoid costs involved with their telephone structure.
- **Internet Telephony Service Providers:** A simple Internet Service Provider (ISP) can offer Internet Telephony Service with minimal changes to their infrastructure.
- **Competition:** Telecommunications companies currently in place will have to offer better service to compete with emerging VoIP technologies.

Because VoIP imitates the functionality offered by the common telephone infrastructure, it naturally fits into a peer-to-peer network topology, where anyone can call anyone else. *Peer-to-Peer systems* are distributed systems without any centralized control or hierarchical organization, where the software running at each host is equivalent in functionality [14].

Following the rapid growth of the Internet during the early 1990s and accompanying investment in IP networking infrastructure by business, vendors, and carriers, VoIP has finally become a viable alternative to using the common telephone infrastructure. There have been a number of projects developed to support this alternative, and a number of projects to help solve the problems that VoIP is facing [4].

The Session Initiation Protocol (SIP) works in concert with the numerous protocols that have been designed to carry various forms of real-time multimedia session data such as voice, video, or text messages. SIP works by enabling Internet endpoints (called user agents) to discover one another and to agree on a characterization of a session they would like to share [5].

The Middlebox Communications (MIDCOM) framework uses the idea of Middleboxes. These Middleboxes, implementing firewall and network address translation services, typically embed application intelligence within the device for their operation. MIDCOM specifies an architecture and framework in which trusted third parties can be delegated to assist the middleboxes to perform their operation, without resorting to embedding application intelligence. Doing this will allow a middlebox to continue to provide the services, while keeping the middlebox application agnostic [6].

3rd Annual Winona Computer Science Undergraduate Research Symposium. April 2003, Winona, MN.

There have also been a number of gatekeepers (an entity discussed in the H.323 standard) developed to aid in the problems VoIP has when interacting with common networking entities. Cisco provides their 'Gatekeeper/Multimedia Conference Manager', which handles more then just H.323 [7].

There is also a number of open source projects dealing with H.323, including the GNU Gatekeeper [8] and the OpenH323 Project (which is made use of in this work) [9].

We know that VoIP is trivial when a public host is contacted by a private host, but what happens when a public host contacts a private host? A series of problems arise here when network address translation is involved between their communications. It is our goal to solve this very general problem by breaking it up into a series of subproblems and working from there.

Our approach to solving this problem was to develop a design that was a part of a common firewall architecture, so that security and routing concerns involved with VoIP could be a part of how the firewall works as well.

In this paper, we discuss the research bringing us to the conclusion of our final voice-aware firewall architecture we dubbed VAFW. Our goal is to show that this design will provide any network with a proper starting point for allowing H.323 communications.

2. Background

For VoIP to be more then a concept, it needs to be integrated into a network. The H.323 Standard provides a way of building this VoIP infrastructure.

2.1. H.323

H.323 is an umbrella recommendation from the International Telecommunications Union (ITU) that specifies components, protocols, and procedures for real-time, peer-to-peer communication over packet-based networks. [1,3]

2.1.1. Architecture

In a general H.323 implementation, four logical entities are required, but there are also a number of terms defined [1,3,10]:

- A *terminal* is a host involved in the VoIP communications.
- A *gateway* provides a translation function between Terminals and other Terminal types (such as IP Phones, Faxes, etc.).
- An *endpoint* is a collective term for a Terminal or Gateway
- A *gatekeeper* provides a number of functions including address translation, admission and access control, bandwidth management, call routing, and Terminal registration. It can be thought of as a Gateway to its controlled Zone.
- A *zone* is a collection of Terminals, Gateways, and the Gatekeeper they are registered with.

Even though an H.323-enabled network can be established with only Terminals, the other components are essential to provide other services from H.323.

H.323 services can be broken up into two main categories, Call Signaling and Media Communications.

2.1.2. H.225 Call Signaling

H.225 call signaling is used to establish a connection between two H.323 endpoints. This is achieved by exchanging H.225 protocol messages on the call-signaling channel. The call-signaling channel is opened between two H.323 endpoints or between an endpoint and the gatekeeper on a well-known port [11].

2.1.3. Media Control and Communications

H.245 Media Control is how the Terminals determine what settings to use before their audio, video, and/or data communications can be established. H.245 has the following main set of functions: [3]

- **Master/Slave Communications**: After the initial signalling, the Media Control services takes over, choosing a master and slave of communications. The master will initiate any openings of logical channels used for media exchange. The slave will simply open up a matching logical channel in the opposite direction of the master. Channels are used either for sending or receiving, so when a master endpoint opens up a channel for sending media, a channel will be opened by the slave to receive that media and vice versa.
- **Capability Exchange**: Each Terminal records their sending and receiving capabilities in a message and sends it to the other Terminals involved in the conference.
- **Opening and Closing of Logical Channels**: Audio and video data are sent through logical channels in the H.323 protocol scheme. These channels are uni-directional and a separate channel is required for audio and video conferencing.

Real-time transport protocol (RTP) provides end-to-end delivery services of real-time audio and video. RTP is typically used to transport data via the user datagram protocol (UDP). RTP, together with UDP, provides transport-protocol functionality [11].

Real-time transport control protocol (RTCP) is the counterpart of RTP that provides control services. The primary function of RTCP is to provide feedback on the quality of the data distribution [11].

2.1.5. A Simple H.323 Call

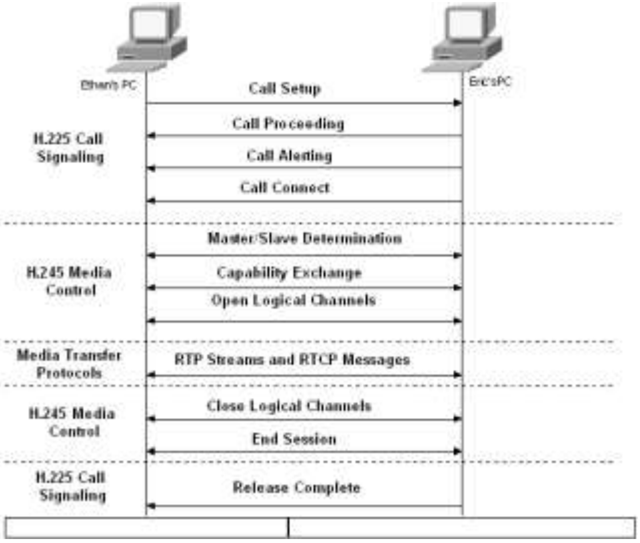


Figure 1: A Simple H.323 Call [11, 12]

Figure 1 shows a sample H.323 call between two endpoints that know each other's addresses. The call initially goes through the call signalling mechanisms to set up a session for communications. After which, the communications session is set up, the logical channels are opened, and media streams are started. There can be any number of logical channels opened and any number of streams occurring, depending on the type of call (video, voice, etc.). After the communications have finished, the channels and session are closed and the release is given, marking the end of the call.

2. Hybrid Firewall Architecture

Figure 2 depicts our view of a hybrid firewall architecture that is in common use in many networks today. It consists of two parts: the packet filtering component and the network address translation component, and it acts as a first line of defense for the network it is a part of [12].

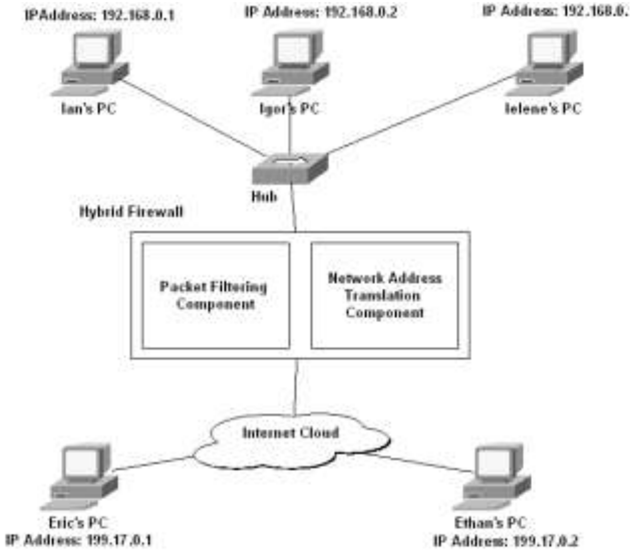


Figure 2: Hybrid Firewall Architecture

2.2.1. Basic Firewall – A Packet Filtering System

A *firewall* is an agent which screens or filters traffic in some way, blocking traffic it believes to be inappropriate, dangerous, or both [13]. The decision on what traffic is considered for filtering is determined by the firewall's security policy, which is usually established through a configurable rules system [14]. The following is a set of basic default rules that govern how a firewall works: [14]

1. Packets from a private network are allowed to flow to a private network.
2. Packets from a public network that are associated with a host that originated communications from a private network are allowed to flow through to the private network.
3. Packets from a public network that originated communications from a public network are not allowed to flow through to the private network.

Very often rule number three is changed to read 'Packets from a public network that originated communications from a public network are allowed to flow through to the private network if they are connecting to a well-known port that the firewall knows about'.

Firewall functions are disjoint from network address translation functions – neither implies the other, although sometimes both services are provided by the same devices [13].

2.2.2. Network Address Translation

Network Address Translation (NAT) is a common component in firewall architectures and provides a transparent routing solution to hosts trying to communicate from disparate address spaces. This is achieved by modifying host addresses en-route and maintaining state for these updates so that datagrams pertaining to a session are routed to the right host in either address space. This allows hosts in a private network to transparently communicate with destinations on a public network and vice versa [15]. The two address spaces we can use are: [15,16]

- A *public network* is a network with an address space that has unique network addresses assigned by the Internet Assigned Numbers Authority (IANA) or an equivalent address registry.
- A *private network* is a network with an address space independent of public network address spaces. The IANA has three blocks of addresses, namely 10/8, 172.16/12, and 192.168.0/16, set aside for private network addresses. Anybody can make use of these address spaces to address their private network infrastructure. These private addresses, however, will not be routed properly in a public network.

NAT is most often used to share a number of public addresses with a larger number of private addresses on a LAN. This helps conserve the number of public IP addresses required for public network communications.

2.2.2.1. Address Translation

Figure 3 depicts how NAT chooses what to translate in the packet header. It will examine the addressing information in the packet header and determine the direction the packet is going. If the packet is going from a private to a public network (outgoing), the NAT component will translate the destination address to be that of the NAT device and the port to be an open TCP port the NAT device can use. This is similar when a packet is going from a public network to a private network, only the destination address and port will be changed.

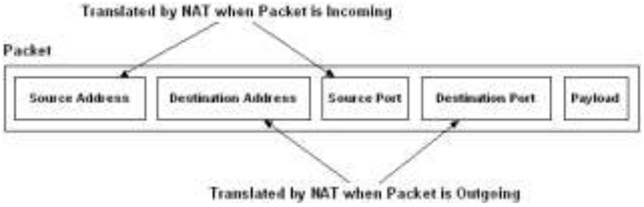

Translated by NAT when Packet is Incoming
Packet
Source Address
Destination Address
Source Port
Destination Port
Payload
Translated by NAT when Packet is Outgoing

Figure 3: Packet View of Address Translation

2.2.2.2. Address Mapping

NAT uses address mapping to determine where a packet goes and whom it should return to. In Figure 4 we see a simple example of how address mapping works. Ian is trying to reach the web service available on Ethan's computer. To do this, he sends a packet to the NAT component. NAT then keeps track of Ian's addressing information when it sends a translated packet to Ethan, also keeping track of the addressing information it uses. When the packet is returned, the NAT component can simply look up what addressing information supplied by NAT maps to what address information supplied by a private host. This address mapping is retained for a certain amount of time, or can be specified by a proprietary setup.

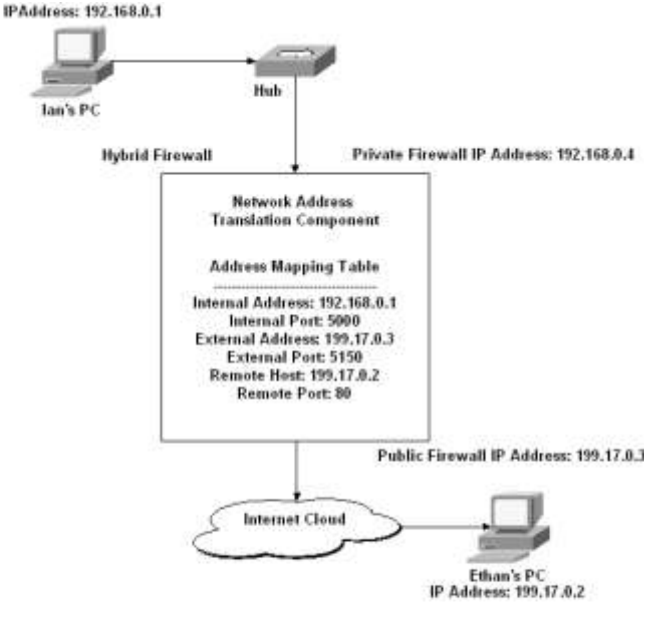


Figure 4: NAT Address Mapping

2.3. Problems

When H.323 communications interact with the hybrid firewall architecture we have discussed, a number of problems arise that need to be solved in order for the firewalled network to be H.323 enabled.

2.3.1. Knowledge of Addresses

The separation of address spaces is at the heart of this problem. Remember that VoIP is a peer-to-peer oriented concept. In order for the concept to be implemented, a host that is a part of a public network, such as the Internet, should be able to contact any host that is a part of a private network, such as in any company's office building. However, this cannot be accomplished because a private network is just that, private. Its basic idea is to have a hidden address layout with addresses that cannot be reached over a public network.

2.3.2. Retention of Address Mapping

H.323 applications require address mapping to be retained across contiguous sessions. These applications require the private-to-public address mapping to be retained between sessions so the same public address may be reused for subsequent session interactions. In the case where the NAT component shares more then one public address with the private network it is segmenting, this can cause a problem. NAT cannot know that the address mapping needs to be retained and may reassign public addresses to different hosts between sessions [17].

2.3.3. Addressing in the Payload

H.323 applications exchange address and port parameters within a control session to establish data sessions and session orientation. NAT cannot know the inter-dependency of the bundled sessions and would treat each session to be unrelated to one another. The most likely reasons for failures will be that addressing information in control payload is address space-specific and is not valid once packet crosses the originating address space [12, 17].

If we examine Figure 3, we see a definite separation of the addressing information. The source and destination addressing information is what NAT uses for address translation, but the information we need is the embedded addressing information inside the payload. Therefore, we need to find a way to have NAT translate the embedded addressing to enable H.323 communications.

3. Our Voice-Aware Firewall Architecture Design

The problems of H.323 combined with NAT lead us to design a series of components that could solve the individual problems covered in Section 2.

3.1. NAT Application Level Gateway

When a packet crosses between different address spaces and there is application specific information contained in the payload (as was discussed in Section 2.3.3), an Application Level Gateway (ALG) is suggested to provide the work around necessary to extract that information and process the packets transparently [6]. In the case of H.323, addressing information will need to be extracted from the payload of packets involved in the call setup so that they can be routed to endpoints correctly. Figure 3 shows how NAT normally translates addresses, but when the ALG module is added to the NAT infrastructure, it will extract the embedded addressing information from H.323 call signaling packets for NAT to use for translation instead. This is a common practice and many ALGs have been developed for other applications, such as DNS [17].

The only time the ALG will need to do the application specific processing is when an original call signal message is received from a public endpoint. This is the first message in any H.323 call. After the initial call signaling, there will be an address mapping setup for the two endpoints to communicate, leaving NAT to handle the routing. If a call signal originated from a private endpoint, the public endpoint's address is already known, and therefore addressing information will not need to be extracted from the payload and an address mapping will be correctly setup.

3.2. Connection Tracking

Because H.323 applications require address mapping to be retained across contiguous sessions (as was discussed in Section 2.3.2) we needed to make sure that when the endpoints exchange addressing information, that the subsequent messages after the call signal will be handled correctly by the NAT. To do this, we need to develop a module for NAT that will maintain state for the address mappings until there is a significant enough of a timeout or the H.323 communications terminate. Otherwise, address mappings may not be correct within the address mapping table, thus leading to packets being routed to incorrect addresses [18].

3.3. H.323 Gatekeeper

A *gatekeeper* is the most important component of an H.323 enabled network. It acts as the central point for all calls within its zone and provides call control services to registered Terminals [1]. Think of the gatekeeper as the go between, finding out information on endpoints that want to communicate and giving them information they need to do so.

Gatekeepers communicate with endpoints using the H.225 RAS (Registration, Admission, Status) protocol. The RAS is used to perform registration, admission control, bandwidth changes, status, and disengage procedures between endpoints and the gatekeeper on a well-known port. The signaling channel, which is used to exchange the RAS messages, is opened between an endpoint and a gatekeeper prior to the establishment of any other channels [11]. A complete list of messages that the gatekeeper makes with other H.323 entities is available as Appendix 7.1.

3.3.1. Gatekeeper Call Signaling

The gatekeeper's call signaling mechanism can be implemented in one of two ways, Direct Endpoint Call Signaling or Gatekeeper-Routed Call Signaling.

In *Direct Endpoint Call Signaling* (DECS), call setup messages are directed between each endpoint in the call, as it is done with the media communications H.323 provides. Usually, this is done in intra-zone calls, since the gatekeeper doesn't necessarily need to do any address translation. This method of call signaling is illustrated in Figure 1, where no gatekeeper exists. However, a gatekeeper may exist and still use DECS [12].

In *Gatekeeper-Routed Call Signaling* (GKRCS) the call setup messages are directed through the gatekeeper, and only RTP will possibly not be routed through the gatekeeper. This method of call signaling is shown in Figure 5, where the Gatekeeper routes the initial call setup and the rest is handled between the endpoints [12].

3.3.2. Gatekeeper Address Registration and Translation

Recall from Section 2.3.1 that endpoints within a private network are just that, private. Because this address space is hidden to the public address space, we needed to find a way to address these private endpoints. This is supplied to us by the functionality inherent in the gatekeeper's H.225 RAS protocol. The gatekeeper uses an addressing method called *alias addressing*. Each private host registers an alias, and the gatekeeper will keep track of their alias-to-address mapping. By having each endpoint register a unique alias to their gatekeeper, we have a unique method of addressing each private endpoint.

We can show how an H.323 call routed by a gatekeeper works using the example depicted in Figure 5. When Ethan wishes to call Ielene, he can use the gatekeeper that controls Ielene as a gateway to its zone. When the call is placed, the call setup message will contain the address of the gatekeeper in the destination address of the packet (199.17.0.3), and the alias address will be contained in the payload (Ielene). The gatekeeper would then use its knowledge of the H.323 payload, extracting the alias address, and forwarding the call onto Ielene. This is a direct example of GKRCS, as was discussed in Section 3.3.1. After Ielene responds back to Ethan, the call session will be established, and the gatekeeper is no longer needed. This is a simplified view of how the gatekeeper itself works, not how the gatekeeper interacts with all of the other components of our voice-aware firewall architecture.

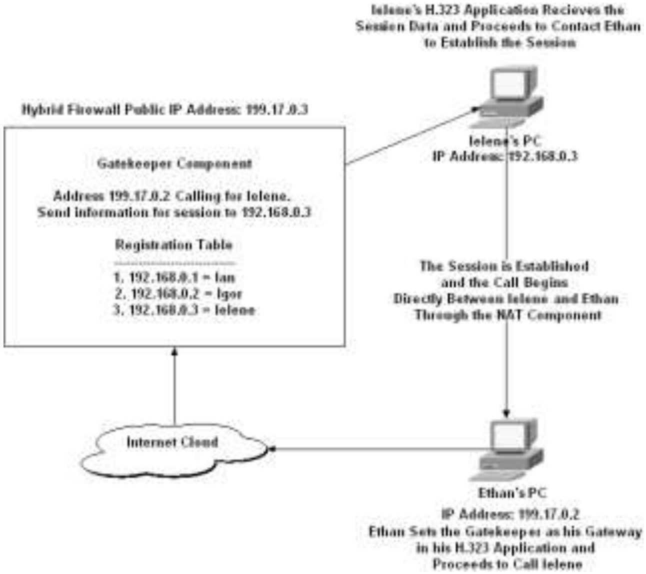

Ielene's H.323 Application Recieves the Session Data and Proceeds to Contact Ethan to Establish the Session
Hybrid Firewall Public IP Address: 199.17.0.3
Gatekeeper Component
Address 199.17.0.2 Calling for Ielene.
Send information for session to 192.168.0.3
Registration Table
1. 192.168.0.1 = Ian
2. 192.168.0.2 = Igor
3. 192.168.0.3 = Ielene
Ielene's PC
IP Address: 192.168.0.3
The Session is Established and the Call Begins Directly Between Ielene and Ethan Through the NAT Component
Internet Cloud
Ethan's PC
IP Address: 199.17.0.2
Ethan Sets the Gatekeeper as his Gateway in his H.323 Application and Proceeds to Call Ielene

Figure 5: H.323 Communications Between a Gatekeeper

Figure 6 shows an in-depth look at which messages are exchanged during a call when a gatekeeper is involved. In this example, Eric does not make use of the gatekeeper, but Ethan does. Since they are in the same address space they can directly communicate and DECS is used as discussed in Section 3.3.1. The initial gatekeeper discovery and registration could of happened any time before Eric's first message is sent.

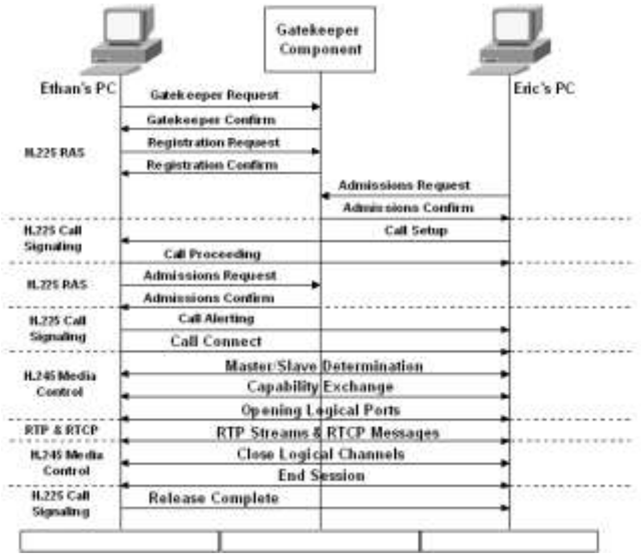


Figure 6: System Sequence Diagram with a Gatekeeper [14,19]

First off, Eric must use the gatekeeper as the gateway to its zone of control. After which, Eric receives admission to make a call to the gatekeeper's zone, placing it to Ethan, a computer in Eric's address space. The call setup and proceeding messages occur, and then Ethan must get permission from the gatekeeper to accept this call. After receiving confirmation from the gatekeeper, the call goes on very similar to how the call occurs in Figure 1. The session is created, RTP and RTCP transfer messages and media streams, and when the session is closed, the call ends.

3.3.3. A Gatekeeper Analogy

A Gatekeeper functions very similar to how someone who answers a telephone does. I live in a house with Eric, Joe, and Cory. This is my basic registration list for alias addresses. If someone calls my house, the telephone rings, alerting me to a call. This is the initial Terminal to Gatekeeper message. I then answer the telephone, asking whom they would like to talk with. If they tell me the name of one of the people I live with, I will simply hand then the phone and let them talk, which acts as the session establishment as well as an example of DECS. If the person asks for someone who doesn't live here, I will tell them so. If the person is someone that we don't want to talk to, I will tell them to stop calling here. These are all functions the Gatekeeper provides, that we know and use everyday when we answer a telephone so they are easy to relate to one another.

3.4. The Full Package: VAFW

Now that we have gone through the background of what it takes to create a voice-aware firewall architecture, and the problems that it solves, we propose our initial design package, clearly defined as VAFW and depicted in Figure 7.

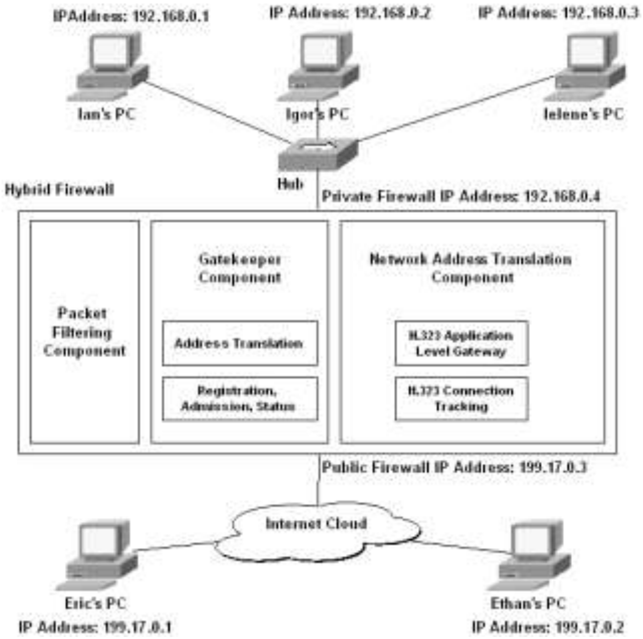


Figure 7: Our Voice-Aware Firewall Architecture Design

The main components involved in VAFW are:

- Packet Filtering Component
- Network Address Translation Component
    - H.323 Application Level Gateway Services
    - H.323 Connection Tracking Services
- Gatekeeper Component
    - Address Translation Services
    - Registration, Admission, Status Services

With our component design finalized, we are ready to show how all of our components interact in an H.323 call.

First and foremost, recall that each of the endpoints registers their aliases. This is the first step of the process for setting up an H.323 calling environment and must be done prior to any of the endpoints receiving a call, otherwise, it will never be authorized by the gatekeeper.

We are going to examine the call signalling in two parts. Once where there is a call signal coming from Ethan, through VAFW, then onto Ielene (Figure 8). Then we will examine when the call signal is replied to from Ielene to Ethan (Figure 9).

Refer to Figure 10 as well for a better look at exactly what messages are passed between each component.

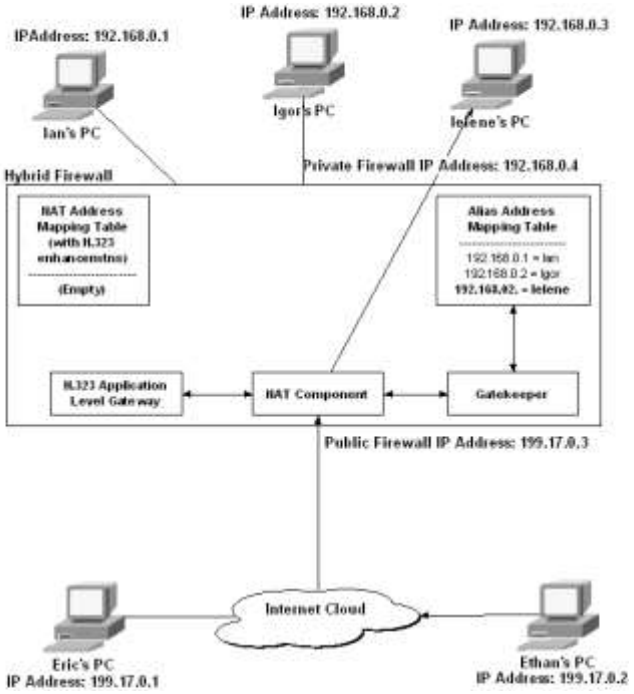


Figure 8: Incoming H.323 Call Signal

In Figure 8 the 'Call Setup' message originates from Ethan. He chooses to contact Ielene with an H.323 application on his computer. The application embeds the alias 'Ielene' in the payload as it sends off this initial message.

NAT receives this message from Ethan and asks the ALG if it needs to examine the packet. Since it is the initial call setup message, the alias 'Ielene' must be extracted and sent back to the waiting NAT component.

NAT then must communicate with the gatekeeper to have the alias 'Ielene' translated. The gatekeeper checks it's alias address mapping table to make sure 'Ielene' is in there. Upon finding the entry, the gatekeeper then returns Ielene's IP address of 192.168.0.3 back to the NAT component.

If you are following along on Figure 10, the 'Call Setup' message has just been let through, and now that NAT has the address it needs, it forwards the packet to Ielene's PC.

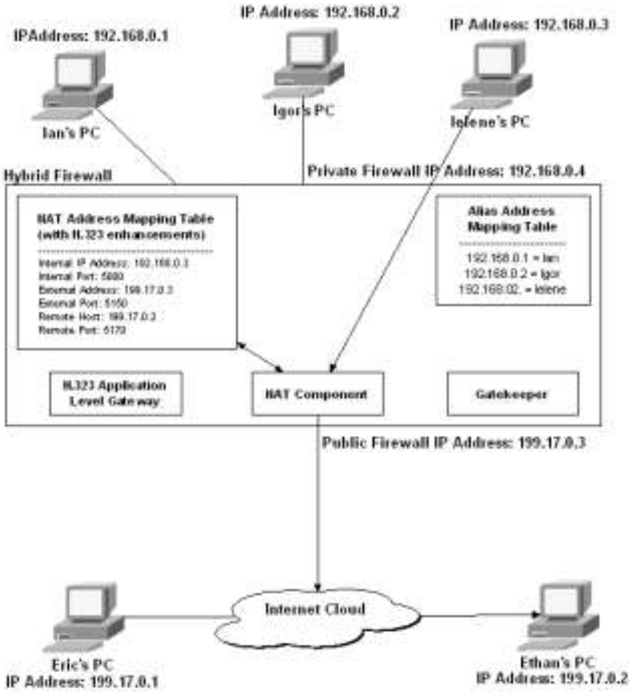


Figure 9: Outgoing Call Proceeding Message

In Figure 9 Ielene is going to send the 'Call Proceeding' signal. This does not require her to first get admission confirmation from the gatekeeper, so the gatekeeper is untouched during this exchange. Ielene sends the message to the NAT component, acting as her gateway to the public network. The NAT component will then translate the source address and port to it's own source address and a randomly unique port. It will then use the addressing information it modified in an address mapping so that when Ielene's response comes back, it will be mapped to her correctly. NAT then lastly forwards the message onto Ethan. Of course, he still has more messages to wait for before the call can proceed, but now you understand how our components interact to allow H.323 communications across the disparate address spaces.

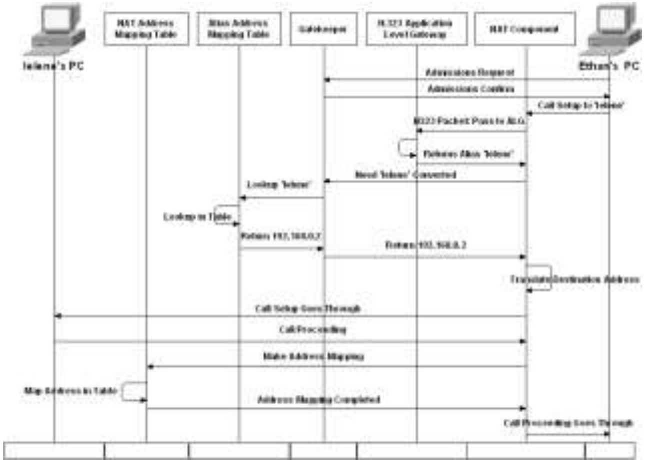


Figure 10: System Sequence Diagram for Figures 8 & 9 [11,12]

4. Conclusion

Voice over IP is a concept that deals with the multimedia conferencing over IP-based networks. Difficulties can arise with public addressed to private addressed H.323 conferencing because of the peer-to-peer nature of VoIP. To solve these problems, we put forward the VAFW design, which can be used to develop a voice aware firewall.

There are still many possibilities to research on this subject. Some of the following could greatly increase the efficiency of the VAFW design or the H.323 standard.

4.1. Dynamic Ports

Dynamic ports provide a very big pain when it comes to firewalls and how they filter traffic. RTP/RTCP and some of H.323's other media control framework allow for the use of a large range of dynamic ports with communications. Again, this information is kept payload specific, but it wouldn't matter either way. To allow for media exchange over these ports, all the ports would have to be opened by default, a very negative side effect is the ability for traffic on these ports that is unwanted (such as cyber attacks) will also be let through. There currently is not a good way of controlling these Dynamic Ports in the firewall architecture and still allow UDP driven protocols to work.

4.2. Encrypted Data Streams

Encryption cannot be used with VAFW currently because it encrypts the payload, which VAFW needs access to for alias address translation. A good way of allowing encryption with public/private keys, while still allowing alias address translation to be accomplished needs to be found.

4.3. Authentication of Aliases

There is a framework present in the H.225 protocol for authentication of aliases when registering. This would be a great feature to add to make sure that aliases can be unique to the zone they are in and the person registering the alias is unique as well.

5. Acknowledgements

My thanks to Derek Hunt, Dr. Gerald Cichanowski, and Dr. Joan Francioni for their help in preparing this project.

6. References

[1] Packetizer, Inc. H.323: A Primer on the H.323 Series Standard. http://www.packetizer.com/iptel/h323/papers/primer. Last accessed: April 5th, 2003.

[2] Kulathumani, Vinodkrishnan. Voice over IP: Products, Services and Issues. http://ftp.netlab.ohio-state.edu/pub/jain/courses/cis788-99/voip_products/index.html. Last accessed: April 5th, 2003.

[3] Karim, Asim. H.323 and Associated Protocols. http://www.cis.ohio-state.edu/~jain/cis788-99/h323/index.html. Last accessed: April 5th, 2003.

[4] Varshney, U., Snow, A., McGivern, M., Howard, C.. Voice Over IP. Communications of the ACM. Jan 2002, Vol. 45, No. 1.

[5] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.. SIP: Session Initiation Protocol. http://www.ietf.org/rfc/rfc3261.txt

[6] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., Rayhan, A.. Middlebox communication architecture and framework. http://www.ietf.org/rfc/rfc3303.txt. Last accessed: April 17th, 2003.

[7] Cisco Systems, Inc. Cisco Gatekeeper/Multimedia Conference Manager. http://www.cisco.com/en/US/products/sw/voicesw/ps4139/. Last accessed: April 17th, 2003.

[8] webmaster@gnugk.org. OpenH323 Gatekeeper – The GNU Gatekeeper. http://www.gnugk.org/

[9] Open H323 Project. Open H323 Project. http://www.openh323.org. Last accessed: April 5th, 2003.

[10] Arora, Rakesh. Voice over IP: Protocols and Standards. http://www.cis.ohio-stated.edu/~jain/cis788-99/voip_protocols/index.html. Last accessed: April 5th, 2003.

[11] International Engineering Consortium. IEC Tutorial on H.323. http://www.iec.org/online/tutorials/h323/. Last accessed: April 5th, 2003.

[12] Cisco Systems, Inc. VoIP Traversal of NAT and Firewall.

[13] Freed, N. Behavior of and Requirements for Internet Firewalls. http://www.ietf.org/rfc/rfc2979.txt. Last accessed: April 5th, 2003.

[14] Liben-Nowell, D., Balakrishnan, H., Karger, D.. Analysis of the Evolution of Peer-to-Peer Systems. Proceedings of the twenty-first annual symposium on Principles of distributed computing. July 2002.

[15] Holdrege, M., Srisuresh, P.. IP Network Address Translator (NAT) Terminology and Considerations. http://www.ietf.org/rfc/rfc2663.txt. Last accessed: April 5th, 2003.

[16] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.J., Lear, E.. Address Allocation for Private Internets. http://www.ietf.org/rfc/rfc1918.txt. Last accessed: April 5th, 2003.

[17] Holdrege, M., Srisuresh, P. Protocol Complications with the IP Network Address Translator. http://www.ietf.org/rfc/rfc3027.txt. Last accessed: April 5th, 2003.

[18] Welte, H. Netfilter Connection Tracking and NAT Helper Modules. http://www.gnumonks.org/ftp/pub/doc/conntrack+nat.html. Last accessed: April 17th, 2003.

[19] Netfilter Project. Netfilter/IPTables. http://www.netfilter.org. Last accessed: April 5th, 2003.

[20] Cisco Systems, Inc. Understanding H.323 Gatekeepers.

Lightcurves and Rotational Periods of the Asteroid Ariane

Christopher John Goeden
Computer Science Department
Saint Mary's University of MN
700 Terrace Heights
Winona, MN 55987
cjgoed99@smumn.edu

ABSTRACT

Asteroids are too small for Earth-based telescopes to resolve and therefore indirect methods are used to determine their size, shape and compositions. One method is to measure the lightcurve over a period of time to find the rotational period of the asteroid. The asteroid Ariane (1225) was observed in January and March of 2003 at the Tenagra Observatories, Ltd. With the software help of Canopus, the rotational period of Ariane was determined to be 5.5400 ± 0.001 hours, with an amplitude of 0.33 ± 0.001 magnitude.

Keywords

Blinking, Dark Frames, Flat Fields, MIR

1. INTRODUCTION

Astronomers have studied the skies throughout the age of humankind. In many cases, human survival depended on their ability to study the stars in order to know when to plant their crops or where to guide their sea vessels. This curiosity extends until today with asteroids. Asteroids, also known as minor planets, are primordial rocky fragments left over from the formation of the Solar System about 4.6 billion years ago. These remnants can be examined to help understand the development of our Solar System. Unfortunately, "with few exceptions, asteroids are too small to be resolved by Earth-based telescopes, so astronomers must rely on indirect methods to find their sizes, shapes, and compositions" (Chaisson & McMillan 2002).

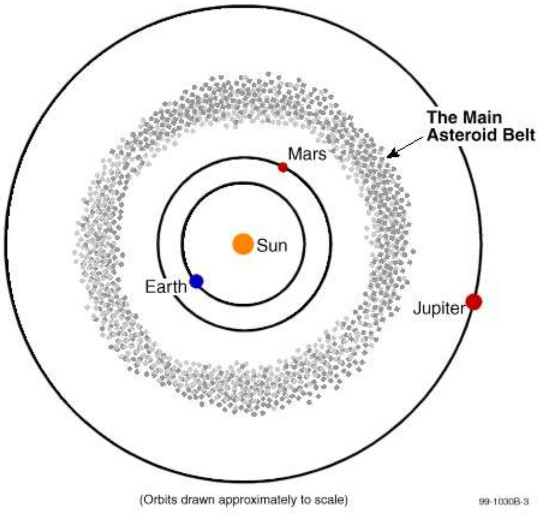
There are many indirect ways to study an asteroid, but in this study the intensity of light will be used to determine the rotational period of one asteroid in particular, Ariane (1225). The light intensity of Ariane was measured over four days to find its rotational period. Knowing the rotational delay, it will be possible to compute the shape of the asteroid with continued observations over several years. The results produced from this study are reported here and will be submitted to The Minor Planet Bulletin [1] to add one more piece to the collaborative study of the Solar System.

3rd Annual Winona Computer Science Undergraduate Research Symposium, April 2003, Winona, MN.

2. BACKGROUND RESEARCH

Ariane was discovered on April 23, 1930 by Johannesburg van Gent [2]. According to Greek mythology, Ariane (Ariadne in French) was the daughter of Pasiphae and the Cretan king Minos. She fell in love with the Athenian hero Theseus, and with a ball of thread she helped him escape the Labyrinth after he slew the Minotaur, a half bull and half man beast that Minos kept in the Labyrinth [3].

During the birth of the Solar System, planets formed from nebulas of gas and dust that fused into a disk around the developing Sun. This disk of dust grains coagulated into larger bodies called planetesimals, which eventually formed planets over 100 million years. However, gravitational interference from Jupiter's huge mass prevented protoplanetary bodies from growing larger than 1000 kilometers, thus asteroids formed. "Asteroids were first observed with telescopes in the early 1800s, and in 1802, the astronomer William Herschel first used the word "asteroid," which means "starlike" in Greek, to describe these celestial bodies" [4]. Asteroids range in size from 1000 kilometers in diameters to the size of pebbles. They have been located beyond Saturn and within the Earth's orbit, but most are contained within the Main Asteroid Belt that exists between the orbits of Mars and Jupiter [4]. Ariane is located at the inner edge of the Main Asteroid Belt.



Most asteroids have elliptical orbits that are very similar to the paths that planets follow. Unfortunately, there are a few asteroids that do not follow this path and some of their paths intersect the

Earth's orbit. "The potential for collision with Earth is real" (Chaisson & McMillan 2002). In 1968, the asteroid Icarus missed our planet by only 6 million kilometers. More recently, an unknown asteroid came as close as 170,000 kilometers from the Earth in 1991. Though most Earth-crossing asteroids are relatively small with a diameter of about one kilometer, such an object has enough energy to devastate an area of 100 kilometers in diameter [5]. From 1990 to 2000, at least 40 asteroids are known to have passed within 10 million kilometers of the Earth and, in this decade, at least 40 more are expected to pass within the same distance [5]. Today, there are several large telescopes that scan the skies looking for objects that are considered potentially hazardous (larger than 150 meters in diameter) or have orbits that could come within the Earth's atmosphere.

3. METHODS

Here is an ordered list of the methods this study used to determine the rotational period of the asteroid Ariane. First of all, a specific asteroid had to be selected to be studied. One list of possible asteroids that could be studied could be found at the Collaborative Asteroid Lightcurve Link (CALL) [6]. This website lists all the asteroids that would have the best observation periods for a given month because of their close proximities to the earth. Ariane was chosen to be studied for two reasons: its apparent brightness is dim and it had a close proximity to the earth (at the time of study) compared to other asteroids. Asteroids with magnitudes that are bright are usually problematic with very long or short rotational periods.

Secondly, the asteroid was reserved by making a request at the CALL website. By reserving the asteroid, the CALL website publishes who reserved it. Since there are many asteroids to study, reserving it on the CALL website removes it from the available list and allows others to study different asteroids. Unlike other science fields that compete for credit for their work, astronomy is a collaborative field that has a huge amount of objects to study.

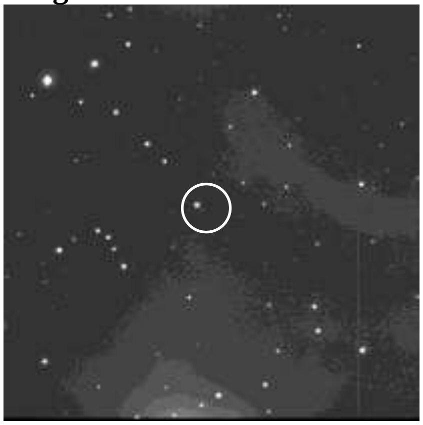
Thirdly, an observatory was chosen and the time to take data was purchased at that observatory. Tenagra Observatories, Ltd. [7] in Arizona was selected to take data for our study for several reasons. (1) The observatory is located in the Sonora desert with humidity less than 15%. (2) The observing season lasts from September 15 to July 1, which corresponded to our time of study. (3) The site has almost 100% coverage of the skies.

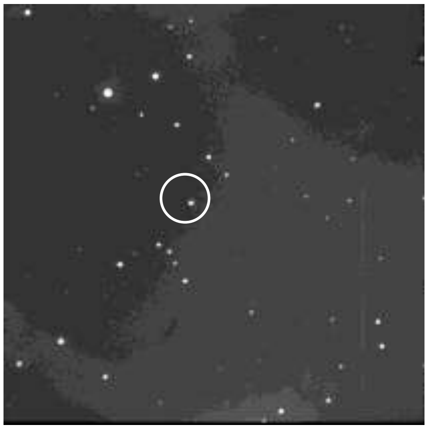
The fourth step involved making an observation request at Tenagra Observatories to photograph the asteroid. The asteroid, the date of observation, the number of exposures to be made and the exposure time were specified to the observatory. Communication with Tenagra was done through an FTP server, using personal space provided for each observer.

4. ANALYZING DATA

Once data is received for the night requested, it can be analyzed. The five methods that we used to analyze our data are discussed below.

4.1 Blinking





When data is first received for the night, the asteroid needs to be found in the pictures. When looking at the sequence of pictures, there should be one dot that changes its position while the rest of the stars travel relative to each other [8]. This is known as *blinking*. One method of blinking involves putting each picture into a slide show. The background stars will move relative to each other in the slideshow but the asteroid will stay stationary in the picture as the night progresses. Since the telescope was focused on taking pictures of the asteroid, the background stars will progress in their paths throughout the night, but the asteroid will appear not to move. Unfortunately, with current technology, telescopes can not always focus the asteroid in the center of the picture. This makes the process of finding the asteroid even harder because new stars can be introduced into the picture.

4.2 Dark Frames and Flat Fields

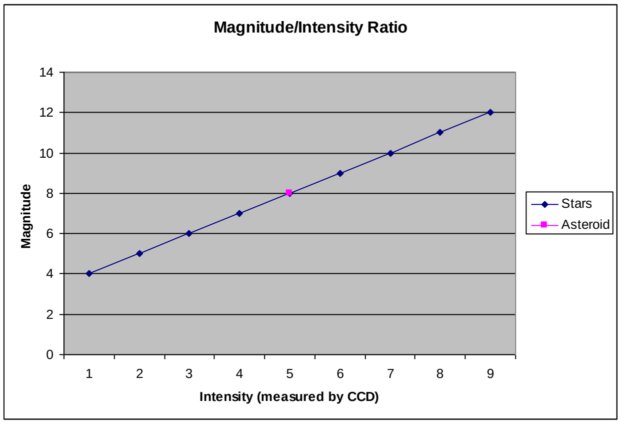
In astronomy today, objects are photographed with electronic detectors known as charge-coupled devices (CCD). "A CCD consists of a wafer of silicon divided into a two-dimensional array of many tiny picture elements, known as pixels" [5]. As light strikes the CCD, electric charge builds up on the pixels. "The charge is directly proportional the number of photons striking each pixel or the actual intensity of light at that point" [5]. The data collected from the CCD is sent directly to a computer, which builds a two-dimensional image of the picture. A typical CCD is a few square centimeters and contains several million pixels.

Though a CCD precisely measures the amount of light, it must be noted that the CCD count can vary for a variety of reasons, unrelated to the true changes in the asteroid's magnitude. For example, atmospheric conditions, electronic noise and instrumental defects can influence the data. Atmospheric turbulence is caused when rays of light from a distant star strike the detector in the telescope at different locations because of the changes in the Earth's atmosphere. Electronic noise in the CCD can influence the data generated from the detector. A common instrumental defect in CCDs is a pixel that is more or less sensitive than the normal sensitivity of the other pixels.

For electronic noise and instrumental defects, *dark frames* and *flat fields* are used to correct each picture. Dark frames are pictures of the natural noise in the camera at a given temperature and length of exposure. They are created by taking a series of images with the shutters closed and then averaging these pictures into a master dark picture. Flat fields are pictures that reflect the sensitivity of each pixel on the CCD. They are created by taking a series of images of a uniform light source over the CCD and then averaged into a master flat picture. The data collected to create the master dark frames and master flat fields are subtracted from each picture to eliminate the electronic noise and instrumental defects.

4.3 MIR

The count recorded by the CCD also depends on atmospheric conditions. Atmospheric conditions include temperature, pressure, humidity, and the presence of clouds. These elements influence the data; fortunately these effects can be accounted for by making comparisons to stars at known magnitudes. To do this it is necessary to establish the relationship between the intensity recorded on the CCD and the know magnitudes of certain published reference stars. This is known as the magnitude/intensity ratio (MIR). With the intensities and magnitudes known for the stars, these values can be plotted into a linear relationship like that shown in Example 1. Since the intensity of the asteroid is also known from the pictures, its magnitude can be found by finding its intensity on the linear relationship.



Example 1. Magnitude/Intensity Ratio

4.4 Detecting Flat vs. Round Asteroids

A dime is easily seen looking at either of its faces, but turn it on its side and much of the face is not visible. Now take a spherical ball and look at its face. No matter which side is shown, the same amount of face is visible. This same principal holds for asteroids in space. Instead of looking at its faces, the intensity or amount of light reflected back to the telescope will be examined. If the asteroid is flat, the intensity values in the data will change dramatically over time. But if it is spherical, the data for the intensity will remain relatively constant over time. Therefore, if the asteroid is a uniform sphere or if the asteroid is rotating slowly, it will be difficult to determine its rotational period.

Once the magnitude of an asteroid has been determined for each exposure, the data can be graphed. If the asteroid's shape is flat, the graph will have a shape similar to a sine curve, since the intensity of light changes over time. However, if the asteroid's shape is spherical, the graph will be a straight line, since the intensity of light stays constant over time. Either way, the rotational period of the asteroid can be found but will be more difficult to determine if the asteroid is spherical. Specifically, the rotational period is computed by subtracting two consecutive minima of the sine curve found on the light intensity over time graph. In our study, after all of the pictures from the different nights were analyzed and graphed, the sets of graphs were combined together to find the true rotational period and magnitude of the asteroid.

4.5 Canopus Software

Canopus is a software program, developed by Brian Warner, to help make the process of studying asteroids easier and efficient. Canopus provides functions to perform blinking and MIR on data, which would take an exhausting amount of time without its automated help. Specifically, the Lightcurve Wizard in Canopus takes the dark frames, the flat fields and the MIR relevant to an object and subtracts them from each picture taken for that night of data. A graphing tool is provided to plot the data of the light intensity over a period of time. Sessions can be created help to keep track of the pictures taken from each night.

5. RESULTS AND ANALYSIS

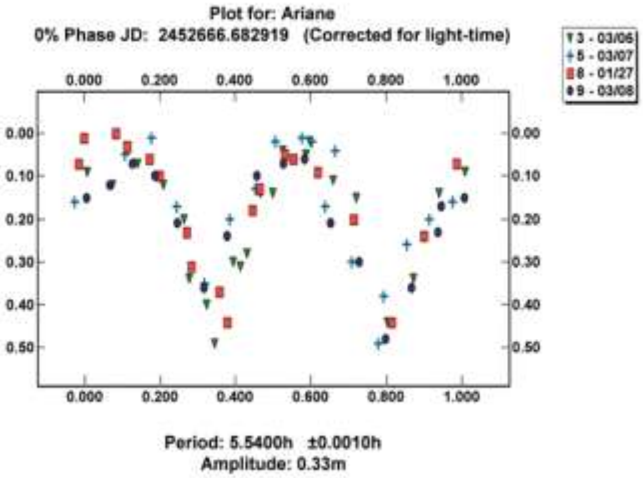
The first night's data was taken a little over a month before the rest of the data was taken. This was done to prove the validity of our research and to prove that the rotational period stayed the same over time. This was done by using Canopus to find the asteroid, subtract the dark frames, flat fields and MIR and then to graph the light intensity over time to determine the rotational period.

The pictures of the dark frames and flat fields that were subtracted from each picture taken are in Appendices A and B. The rotational period and magnitude of all the nights of data are shown in Table 1. The mean deviance of each night's rotational period can be found by subtracting the combined graph's rotational period from a night's rotational period and then dividing that number by the combined graph's rotational period. Each night's rotational period mean deviance was very close to each other with the greatest mean deviance being only two percent.

The graphs of each night are in Appendices C-F. Looking at the graphs, a distinct sine wave can be seen within each graph. By combining every graph for all four nights, it was computed that Ariane has a rotational period of 5.5400 ± 0.001 hours, with an amplitude of 0.33 ± 0.001 magnitude. The compilation of graphs is shown in Graph 1. In Graph 1, the data points form a sine curve of the rotational period of Ariane.

Date	Rotational Period	Mean Dev.	Amplitude
1/27/2003	5.5300 ± 0.070 hours	0.20%	0.37 mag.
3/6/2003	5.5400 ± 0.060 hours	0.00%	0.34 mag.
3/7/2003	5.5000 ± 0.050 hours	0.70%	0.28 mag.
3/8/2003	5.7000 ± 0.170 hours	2.80%	0.28 mag.
Combined Graphs	5.5400 ± 0.001 hours		0.33 mag.

Table 1. Rotational Period and Magnitude of Ariane



Graph 1. The combined data for the rotational period of Ariane.

6. CONCLUSIONS

Asteroids are remnants left over from the formation of the Solar System that can help explain how it formed. Since asteroids are too small to be viewed with Earth-based telescopes, the rotational period can be examined by measuring the light intensity over time. The goal of this project was to determine the rotational period of the asteroid Ariane by measuring the light intensity over time. Once Ariane was reserved, the pictures were taken from Tenagra Observatories, Ltd. Then the data was analyzed by using blinking, dark frames, flat fields and MIR. Following these analyzing methods, the rotational period of the asteroid Ariane was successfully measured to be 5.5400 ± 0.001 hours, with an amplitude of 0.33 ± 0.001 magnitude. The future plans of this project are to reserve other asteroids and measure their light intensities over time to find their rotational periods.

7. ACKNOWLEDGMENTS

Thanks go to Dr. Matt Vonk for his hard work at organizing meetings and helping our group with problems. Thanks also go to the Physics department for funding our project. The amount of data needed for this experiment would not have been possible without their donations.

8. REFERENCES

[1] Binzel, Dr. R. P., ed. Lightcurve Photometry of Asteroid 1058 Mussorgskia. *The Minor Planet* Bulletin 29, 4, 67, 2002.

[2] *DATES, lieux et découvreurs the astéroiedes (1001 à 1500).* Available from: http://www.bdl.fr/Granpub/Promenade/pages5/528.html. Accessed 2003 Mar15.

[3] Everhart, D. and Irvine, M. *A Brief Description of the Labyrinth Project.* 2002. Available from: http://www.georgetown.edu/labyrinth/info_labyrinth/ariadne.html. Accessed 2003 Feb 11.

[4] *Solar System Exploration.* 2003. Available from: http://solarsystem.nasa.gov/features/planets/asteroids/asteroids.html. Accessed 2003 Mar 14.

[5] Chaisson, E. and McMillan, S. *Astronomy Today.* Prentice-Hall, Inc., Upper Saddle River, NJ., 2002.

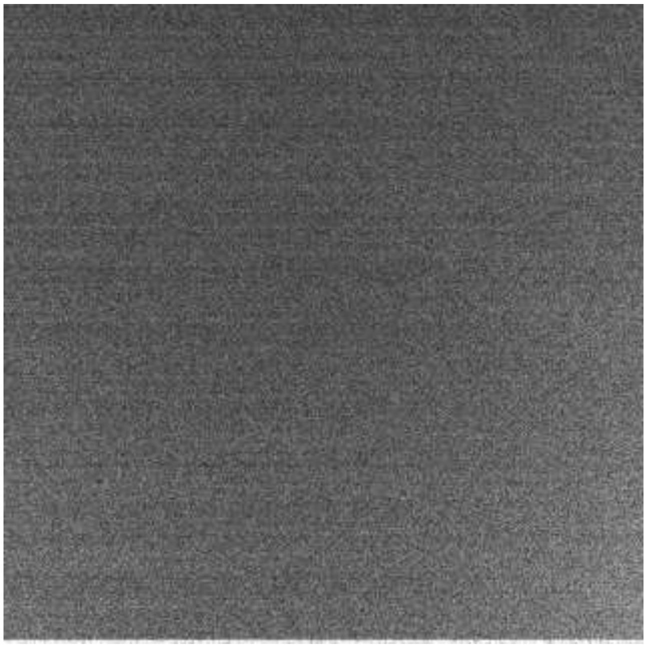
[6] Harris, A. W. *Collaborative Asteroid Lightcurve Link (CALL).* 2003. Available from: http://www.minorplanetobserver.com/astlc/default.htm. Accessed 2003 Feb 11.

[7] *Tenagra Observatories, Ltd.* 2003. Available from: http://www.tenagraobservatories.com/index.html. Accessed 2003 Feb 11.

[8] Warner, B. D. *MPO Canopus and PhotoRed Installation Guide and Reference Manual.* Bdw Publishing., 2002.4. Holvorcem, P. and Schwartz, M.

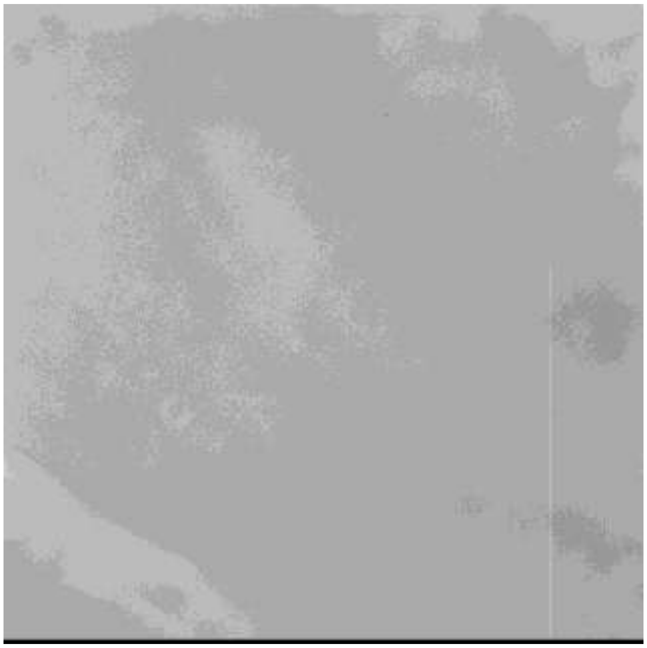
Appendix A

Picture of a Dark frame



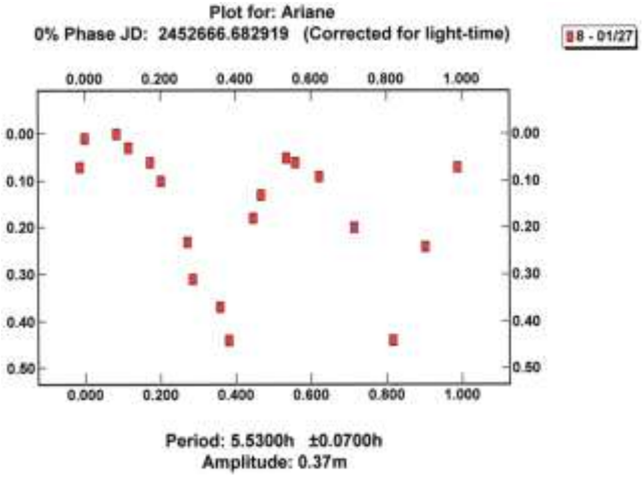
Appendix B

Picture of a Flat field



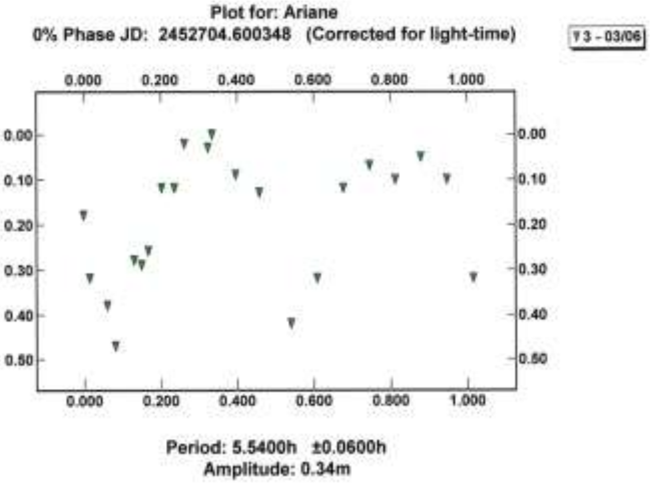
Appendix C

Graph of the rotational period of Ariane on January 27, 2003.



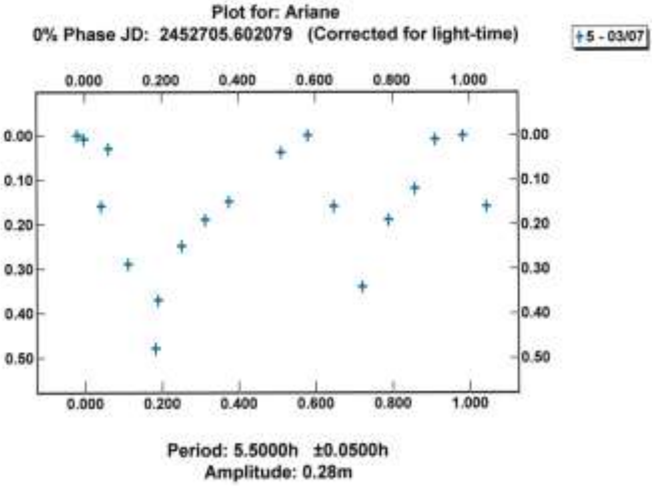
Appendix D

Graph of the rotational period of Ariane on March 6, 2003.



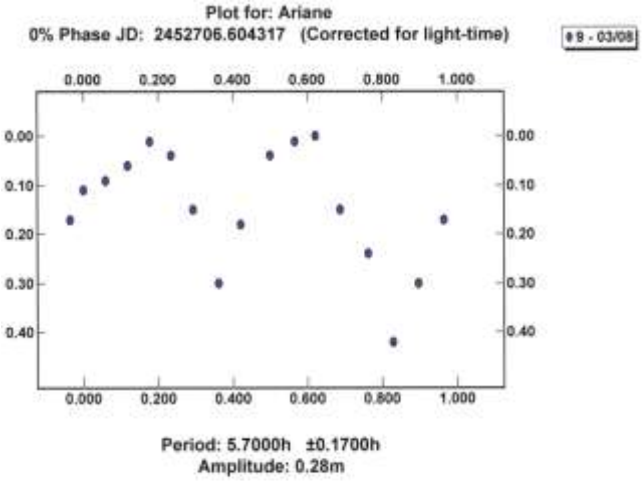
Appendix E

Graph of the rotational period of Ariane on March 7, 2003.



Appendix F

Graph of the rotational period of Ariane on March 8, 2003.



Digital Symphony:
Simulated Live Performance via Distributed System

David Just
Winona State University
Department of Computer Science
Winona, MN 55987
DJJust6229@webmail.winona.edu

ABSTRACT

Digital Symphony is a system of computers that attempts to re-create some of the atmosphere of a live performance. It demonstrates the possibility for computerized players to communicate much like the members of a human band. Digital Symphony (DS) uses a multicast network protocol to send messages between the members of the simulated group - which allows for communication and synchronization during the performance. This simulation can then be used to test new music without requiring a live band. It should also increase the believability of a synthesized performance.

Keywords

Jfugue, MIDI, Conductor, Member, Player, Entities, Digital Symphony.

1. INTRODUCTION

1.1 Topic

Digital Symphony is a band whose members are computer programs running on separate interconnected machines. It attempts to re-create some of the atmosphere of a live performance of a symphony. By using multiple computers, DS creates a system that can be used to preview music as it would be heard if played by a live band.

1.2 Problem

Normal systems for playing music on electronic devices such as Stereos or Computers are acceptable for playing music for personal enjoyment, but they are unable to reproduce the spatial, timing and variance aspects of a live performance. What is needed is a system that can re-create these aspects.

One solution is to simply play different components of a symphony on different machines. This would increase the spatial presence of a performance. However without some type of synchronization scheme between the components it would soon fall out of sync reducing the quality of the performance. It would sound like a band where each member cannot hear what the others are doing. It would not be able to adequately capture the rapture of the live performance. A system without constant synchronization would also not be able to be extended for other more dynamic purposes. What we need is a way for a group of computers to simulate the dynamic communication and performance that occur among real musicians during a live performance.

1.3 Approach to solution

Digital Symphony attempts to simulate this communication among musicians through a distributed software system. Each entity in the system will communicate with the other entities of the system. This communication will allow the members of DS to synchronize with each-other and to dynamically adapt to changing conditions.

1.4 Format of paper

This paper will cover the background of the technologies used in its design and implementation, along with some background into music and necessary aspects of communication that go on during a performance. It will then discuss the implementation of Digital Symphony in some detail. From there the paper will go on to discuss the methods of selecting the volunteers that are to be used for assessing the ability of Digital Symphony, and the methods of testing. The results of the test and the approach that was taken to get them will then be discussed. The conclusion that results from these tests will be explained in the later section of the paper.

2. BACKGROUND

2.1 Musical Aspects

During the performance of a song, there is a great deal of communication going on between the members of a band. Band members use nods, hand gestures and the music itself to communicate between each other. The bass player sets the tone and speed of a song by adjusting the tempo of the song. These cues are used to keep the song synchronized, change the tempo, and, in more dynamic environments are used to signal when things like improvised solos are going to begin and end.

2.2 Technology

2.2.1 MIDI

Digital Symphony relies upon MIDI (Musical Instrument Digital Interface). MIDI is an open standard that is used by most digital instrument manufacturers as the standard communication link between instruments [5]. When computers became part of standard music equipment MIDI was accepted as the way to store

3^{rd} *Annual Winona Computer Science Undergraduate Research Symposium*, April 23-24, 2003, Winona, MN.

and play music on them. MIDI allows a digital instrument to save and synthesize music. Unlike sampled music such as wav or mp3, which are just digital representations of sound waves, the sound produced by a MIDI instrument or computer is stored as a series of MIDI commands. A synthesizer is then used to produce music by interpreting this text. [2][5]

2.2.2 Jfugue

JFugue is a collection of Java classes that can convert human readable ASCII text into MIDI notation and can also synthesize music from the ASCII notation. JFugue is an Open Source project and was modified to add the necessary network capabilities to the package. The ability to play notes on demand instead of in a sequence will also be added to the package. The JFugue package can be downloaded from the JFugue Website. [4]

3. IMPLEMENTATION

3.1 Overview

DS is implemented using the Java language. Java was chosen because it offers an easy to use standard library for networking, and Jfugue was available to be used to convert a human readable music notation to the standard MIDI format. DS consists of two types of programs called entities. The entities communicate with each-other through a broadcast message system consisting of three classes of messages. These messages are used to coordinate the distribution, synchronization, and playing of the current song. There is also a type of message that is currently not used in Digital Symphony, but is there in hopes that DS can be extended.

3.2 Entities

There are two types of entities in the Digital Symphony system: the Conductor and the Players. The Conductor is the controlling program of DS it is where users interact and is responsible for the loading and distribution of songs to the Player entities. It is also in control of synchronizing the Player entities by using Beat and Song Status Messages. The Conductor controls the tempo using a special beat-keeper thread, that keeps a steady beat, to signal when a Beat Message should be sent. This simulates the bass line that helps keep a real band stay synchronized. Each Player entity is responsible for one part of a song. When a Player entity is started it sends out a Song Status sign on message asking to sign on to the system. In response it is given a Song Status song segment message that contains one part of the song to be played. It then proceeds to convert the song part into MIDI format so it is ready to play. At this point, the Player sends a Song Status ready message to the Conductor which registers that Player as ready. When all Player entities are in the ready state the Conductor can be told by the user to start the song. This invokes the Conductor to start the beat-keeper thread and to send out a Song Status start message. When the Players receive the Song Status start message they will start playing the first piece of their song and listen for Beat messages. This same sequence of events is repeated by the Player entities when they receive a Song Status new song message.

3.3 Communication Messages

3.3.1 Beat Messages

Beat messages are sent out by the Conductor. Beats are used by the players to communicate the tempo of the music. A Beat message consists of a single command string containing one phrase: “NextNote”. Upon receiving a Beat message, a Player will check to see if it is in the middle of playing a note. If it is the Beat message is added to a beat queue for processing. If Player is not currently playing a note then the next note in the note queue will be played. As soon as the current note is done the Player will check if the beat queue has any Beat messages in it, if it does then the Player will process those beat messages.

3.3.2. Song Status Messages

Song Status messages are used by all of the members to communicate the necessary data for playing of a song. The Conductor uses Song Status messages to check how many Players are in the current system and to distribute parts of the song to be played. See Table 1 for the actual format of Conductor Song Status Messages. Player entities use Song Status messages to inform the Conductor when they are started, when they are ready to begin playing, and when they are done playing a part of the song. See Table 2 for the actual format of Player Song Status messages.

Table 1. Conductor Entity Song Status Messages

newSong	Song Status new song message. Informs the Player entities that a new song has been loaded into the system. Each Player will respond with a Song Status sign on message
[playername]@ [songstring]	Song Status song segment message. It is used to distribute songstring to the Player with the name playername
start	the Song Status start message is used to tell the Player entities when to start playing the song.

Table 2. Player Entity Song Status Messages

signOn@ [playername]	Song Status sign on message sent to the conductor to inform when a new Player enters the system. This message is also used as a response to a newSong Song Status message.
segmentDone @ [playername]	Song Status segment done message is sent to the conductor to inform when a Player is done playing a segment of the song.
ready@ [playername]	Song Status ready message is sent to the conductor after a song segment message has been processed and the Player is ready to begin playing.

3.3.3. Extensible messages

Extensible messages are abstract and can be used to define new ways to communicate between the players. Extensible messages are implemented in hopes that Digital Symphony can be extended for further research later on.

3.4 Network System

DS is implemented on a 10/100Mbit Ethernet system. Each computer is connected to a single non-intelligent hub so that the max hop count between any two Members is one. This type of network has been shown to have better than acceptable latency, so that the delay between the sending and receiving of a musical event will not be noticeable to the user of the system. [1][6] To implement the DS communication system an underlying protocol had to be chosen. IP was the clear choice for a network protocol, but a choice had to be made between TCP or UDP for the transport layer.

TCP would have allowed for more complex messages to be passed between the entities and allow for more of the processing work to be done at the Conductor instead of down at the Players, but would require a true client server architecture. To send out a message to all of the Players the Conductor would have to iterate through the numerous connections and send the message to one Player at a time. This would have created a significant delay between the time the first and last Players hear the same message. The delay caused by this iteration would cause the Players to quickly become out of sync. TCP would not have been an acceptable choice for implementing DS.

The broadcast nature of UDP Multicast is much more similar to the way a real band communicates. UDP Multicast allows for a single message to be heard by multiple entities at the same time. This reduces the amount delay between different entities hearing the message and reduces the total amount of network traffic. UDP Multicast in Java only has APIs for work with byte data so all of the communication in DS had to be done using Strings which are easily converted from objects to bytes and back. UDP Multicast has the benefit that no connections exist between entities so any entity can communicate with any other entity simply by multicasting a message on the network.

Figure 1 shows the layout of this networking system. The Composer system does not have any speakers attached to it. Attached via a dumb hub are the Player members. There can be any number of Player machines, in our demonstration we will be using only four.

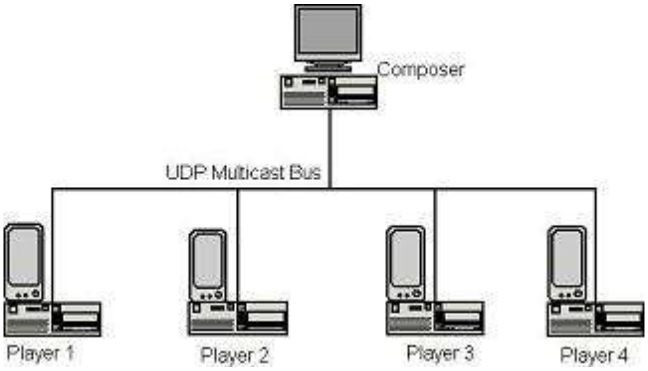


Figure 1. Network layout

3.5 Current Status

Due to time constraints and programming issues, we were not able to create a fully working version of Digital Symphony. We were able to create an intermediate version of DS that is much like the system described in paragraph two of our problem description. It distributes the song pieces to the Players and tells them when to start playing, but does not have constant synchronization using Beat messages. This is the version that we used in our testing process. In testing this intermediate version of DS we hoped to prove two things. First we hoped to prove that distributing different parts of songs to individual players would increase the spatial aspects of the song to make it sound more like a live performance. Secondly we hoped to prove that without the synchronization of Beat messages the song would start to become out of sync proving that this intermediate version would not be adequate to create a better way to listen to music.

4. TESTING SELECTION

4.1 Volunteers, Songs and Methods

A group of 20 volunteers was asked to listen to five pieces of music and was asked questions about what they thought of the quality of each piece. Each piece of music was played twice, once by a control system and once by DS.

4.1.1 Control System

To eliminate as many differences as possible between the control system and DS we created a separate system that uses the same song notation and conversion package as DS. It is also distributed across multiple computers, but instead of playing only one instrument on each machine all parts of the song will be played on all of the machines. By sharing as much code as possible with DS we were able to eliminate the possibility that one format of the song is of better quality than another. By using multiple computers, we were also able to eliminate the possibility that a spatial difference would be caused by using only one computer.

4.1.2 Song Selection

To eliminate as much personal bias from the test subjects answers we selected a number of different types of songs. Due to time constraints during the testing phase of DS and to not take much of our volunteers time, we used only five songs. Each song was about a minute in length. Two were classical songs which most people can enjoy and respect. We also used one song that was high on the rating charts when it was first released, and one Jazz song as an external variable. The final song that was played was alternated between the four previous songs, and a third classical song. Instead of being played by both the control system and DS the fifth song was played twice in a row by DS. This allowed us to compare the results of song five to itself to find out how consistent the volunteers were.

4.2 Survey

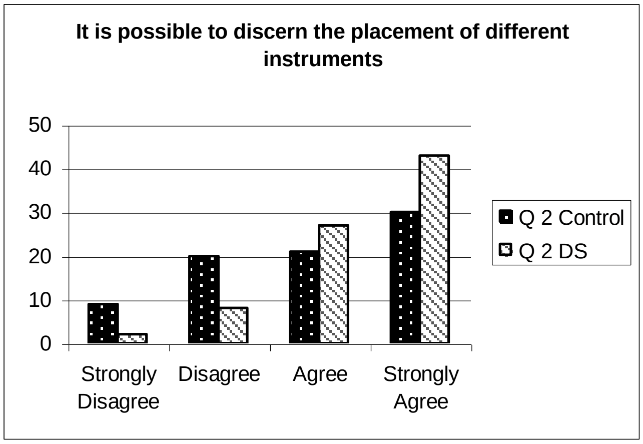
The questions that were asked during the testing of Digital Symphony were selected to be able to tell different things about the system. To ease the statistical analysis process, the questions were phrased so that they could be answered using the form “I strongly Disagree” to “I strongly Agree” on a one to four scale. Eight questions were asked about each run of the song.

Each of the eight questions was chosen to test a single variable out of three groups of properties: sound quality, spatial quality, and synchronization quality. By choosing questions that focus on each of these properties we were able to determine what aspects of DS improved the listening experience, and which aspects were not affected by DS. The individual questions and explanations on why they were used can be found in appendix 11.1 Survey Questions.

5. TESTING RESULTS

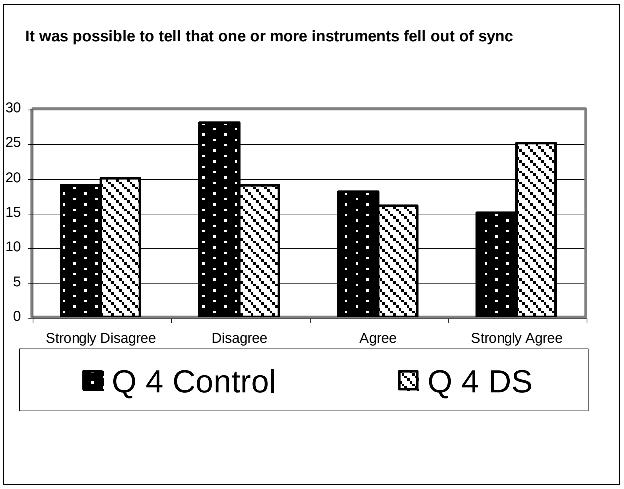
The initial testing of Digital Symphony was very successful. The results of our testing show that the version of DS that we tested, which does not have constant synchronization, does improve the spatial ambiance of a performance, and in doing so has an overall better listening experience than that of the control system. Question 2 of the survey shown in Graph 1 demonstrates this. The higher numbers (which represent the number of occurrences of each answer) for DS on the Strongly Agree side show that more volunteers could discern the placement of different instruments. Our two group dependent test for this question showed a confidence interval from -0.77 to -0.20 with a below zero answer meaning that DS was better at demonstrating spatial ambiance. This means that we can be 95% confident that DS improved the spatial ambiance.

Graph 1



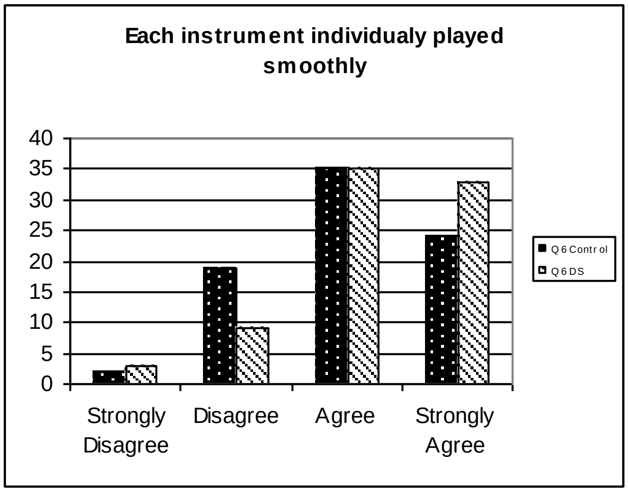
Our testing also showed that without having constant synchronization some instruments would fall out of sync during the performance. This was best demonstrated by Question 4 of our survey and is shown in Graph 2. The two group dependent test for question 4 gave us a confidence interval from -0.47 to 0.049 although this confidence interval contains zero which means that we cannot statistically say there is a difference between the control and DS we can definitely see that there is a skew into the negative range which favors DS

Graph 2



Interestingly. We also found in our testing that DS seemed to improve the smoothness of each individual instrument. We are attributing this phenomenon to the fact that by separating out the instruments the volunteers were better able to pick out which instruments were not playing smoothly as opposed to trying to decipher witch was the offending instrument when they were all mixed together. This was exposed by question 6 of the survey and is shown in Graph 3. The confidence interval for this question ranged from –0.44 up to 0.02 which being so far skewed onto the negative side of zero demonstrates the improvement by DS.

Graph 3



The fifth song that was played for the volunteers was used as a control song. Instead of playing the song once on the control system, we played it twice using DS. We did not tell the volunteers that we had done this and could therefore use the results of this question to determine the validity of our testing procedure. If our testing was perfect, we would end up with a confidence interval that perfectly surrounded zero on both sides for all eight questions. Unfortunately this did not happen. Some of the results on the analysis of song 5 showed that a difference may exist between the first and second time a song is played by

DS. We need to do more testing to determine the cause of this discrepancy, and to further our research.

The statistical analysis of all eight questions can be found in appendix section 11.2 Two group dependent tests

6. FUTURE POSSIBLITIES

The current version of Digital Symphony does not have the synchronization system that was originally planned for the project. We would like to continue our work to finish this section, and create an even better experience for listening to music. We also hope that our Extensible Messaging system can be used to extend Digital Symphony into new realms of research, such as Artificial Intelligence problems.

7. CONCLUSION

In this paper, we have presented Digital Symphony, a distributed environment that attempts to recreate the ambience of a live performance, and to improve the enjoy-ability of listening to synthesized music. In implementing DS we presented the design requirements for the system, and how we created a way for computers to act like members in a band. We have shown that the current implementation of DS does increase the spatial presence of music, but has significant problems with staying synchronized. Both of these properties of the current implementation support our original assumption that without constant synchronization a distributed music system would fall out of sync, destroying any benefit that was gained by the distribution. In the future, we plan on finishing the synchronization code for DS and testing to see if it improves the quality and ambiance of a performance.

9. ACKNOWLEDGMENTS

Our thanks go to Dr. Sudharsan Iyengar for helping create this project, David Koelle the creator of Jfugue who was a big help during the creation of DS, Dr. Chris Malone for helping set up the statistical test and analysis, and to Alex Caffari and Evan Dooley for converting pieces of music to DS notation for testing.

10. REFERENCES

[1] Byungdae Jung, Jaein Hwang, Sangyoon Lee, Gerard Jounghyum Kim, Hyunbin Kim, Incorporating co-presence in distributed virtual music environment, In Proceedings of the ACM symposium on Virtual reality software and technology, Seoul, Korea, 2000, ACM Press, New York, NY, 2000 pages 206-211

[2] David Brian Williams, Peter Richard Webster. *Experiencing Music Technology*. Schirmer Books, New York, NY., 1996

[3] David P. Anderson, Ron Kuivila. A system for computer music performance.*ACM,* Volume 8, Issue 1, 56-82, 1990

[4]David Koelle, JFugue Music Package, http://www.innix.com/jfugue/index.html, accessed Febuary, 2003

[5] Eleanor Selfridge-Field (Editor). *Beyond MIDI The handbook of Musical Codes*. The MIT Press, Cambridge, MA., 1997

[6] John P. Young, Ichiro Fujinaga. *Piano Master Classes Via the Interent*. [online] 1999. Available from : http://gigue.peabody.jhu.edu/~jpyoung/research/NetMIDI/netmidi.htm. Accessed on 2003 Feb 10.

11. Appendix

11.1 Survey Questions

11.1.1 The Quality of the song is acceptable.

This was asked as a general question to determine if the song sounded good. If it was found that overall quality on both the control and DS was not acceptable then we would not have been able to conclude any further information about the system. If only one system showed that quality was a problem then we could conclude that the system was not acceptable.

11.1.2 It is possible to discern the placement of different instruments.

The results of this question show if it was possible to tell that different instruments were in different places. If DS showed an increase in the percentage of volunteers who could tell that different instruments were coming from different places then we can say that DS increased the spatial presence of the music.

11.1.3 It is possible to discern the placement of different instruments.

This question exposes if it is possible to detect a difference in acoustic continuity between the control and DS. A difference in continuity between the control and DS would show that a problem exists the synchronization system. If both the control and DS have high numbers in the disagreeing range then there may be a problem with the conversion of the music or the way it is played by the synthesizer.

11.1.4 It was possible to tell that one or more instruments fell out of sync.

This question is meant to help detect if there is a synchronization problem that only seems to affect some of the Players in the system. This type of problem could be caused by lost Beat messages or by some java internal timing issue.

11.1.5 With eyes closed I feel more like I'm at a concert

This question is meant to test the believability of the system. If a larger percentage of the volunteers agree that DS sounds more like a concert that the control system then we have shown that DS improves the listening experience and does in fact sound more like a concert. If DS does not improve the listening experience then this question will show little to no difference between the control and DS.

11.1.6 Each instrument individually played smoothly

This question shows if there is a problem with instruments not being told to play fast enough or possibly a problem with the way the MIDI is being interpreted.

11.1.7 One or more instruments sounded like it was not in the correct spatial position

DS does not attempt to designate where instruments are placed within the system. Under normal circumstances this should not be a problem but we felt that it was a valid question to ask the volunteers. If a problem is discovered with this question we might have to go back and design a way to designate what Player entities get different parts of a song.

11.1.8 One or more of the instruments seems out of pitch with the others

This question is meant to help determine the overall sound quality of the system. Because we are using synthesizers that are built into the soundcards of normal computers, we may have a problem with the pitch quality of the music. If this does pose a problem then we need to look into getting better hardware for the system.

Procedural versus Object-Oriented Simulation Code: A Performance Comparison

Marcus R. Routsong
Department of Computer Science
Winona State University

ABSTRACT
The object-oriented paradigm is defended to be more intuitive, widely applicable, extensible and adaptable than procedural code. However, object-orientation is considered to be slower than procedural code. Computer simulations are significant in this case not only as an area where both forms of code are applied, but also because object-orientation lends itself to the modeling of actual phenomena, especially in cases where the number of entities being simulated is not fixed or cannot be predetermined. To determine the time performance differential between object-oriented and procedural simulations, the speed requirements of the same computer simulations implemented in both object-oriented and procedural code are measured and compared.

KEYWORDS
Object-Oriented, Procedural, Computer Simulation

1. INTRODUCTION

Proponents of the object-oriented paradigm find it preferable to procedural programming because the paradigm is more intuitive for the programmer, more extensible to other applications, and is more easily adaptable. However, it is also claimed that object-oriented software is slower than its procedural counterpart [11]. Research in this area concerns how to measure this discrepancy, as well as whether these differences accurately reflect superiority for one form of programming over the other.

Stensrud and Myrtveit suggest that the superiority of object-oriented programming over procedural programming would only be determined by an extended series of experiments in which a representative range of applications are composed in "identical pairs," one procedural and the other object-oriented. Whichever form of code has better performance over the entire series is superior to the other [16].

It can be argued that object-oriented programming and procedural programming are too different to be directly compared, especially on such qualitative bases as programmer simplicity and potential future applications. An analysis remains complicated since the ease of design, implementation, and intuition are difficult to measure. Further, extensibility and reuse would likely be easier to quantify, but would require a direct parallel from procedural applications to object-oriented applications.

In this case the applications are specifically computer simulations. An advantage of implementing computer simulations is that in some situations the number of objects instantiated may not be predetermined or predictable, which may affect the speed and memory consumption of the object-oriented program.

This experiment follows Stensrud and Myrtveit's call to further quantify the performance differences of object-oriented and procedural programs [16]. Should the disparity between the two be sufficiently quantified, it may be possible to justify the use of one form of programming over the other based on how much effort, time, and other resources the programmer is willing to devote. That is, measurable or calculable cost aids in making an educated decision when a programming design decision seems to be based upon subjective factors such as knowledge, experience, and devotion to alternate methods.

In this case, the effect of an indeterminate and unpredictable volume of entities created at runtime is explored. It is hypothesized that as in other comparisons, the procedural simulation will perform faster than the object-oriented simulation. However, the indeterminate number of objects may have an affect upon how much time is required per simulated item relative to the procedural simulation.

2. HISTORY

2.1 Past Research of Object-Oriented and Procedural Programming

Comparisons of the two programming methodologies have been drawn in several analyses. One study concluded that "structured" (procedural) analysis techniques are a more natural approach to systems analysis [17]. However, this experiment was limited to student subjects with relatively small problem domains. Another experiment found that object-oriented analysis models and data flow models required approximately the same time and effort to

3rd Annual Winona Computer Science Undergraduate Research Symposium, April 2003, Winona, MN.

construct and are therefore roughly equivalent [1]. Both of these studies relied upon the measurement of human experience: naturalness and effort are based upon subjective responses from the test subjects.

A more quantitative study was performed comparing the speed of linear mathematical functions in FORTRAN-90 (a procedural language) and in Java (an object-oriented language). A "well-structured object-oriented program" (a program that was created according to the object-oriented model) was written to solve these equations, as was a FORTRAN-like Java implementation in which every method was static and no objects were instantiated. These two implementations were run on the same machine, as was their FORTRAN-90 implementation. The findings were that the static object-oriented methods performed slightly slower than their procedural counterparts, and more importantly that the strictly object-oriented code was on average slower by a factor of ten [4]. In this case an object-oriented language was shown to be almost as fast as the procedural, but the object-oriented paradigm was shown to be detrimental to performance concerns.

2.2 Simulation Modeling and Object-Oriented Languages

Because object-orientation entails creating objects whose behavior and properties can duplicate those of real world counterparts, object-orientation lends itself to simulation modeling. This is especially the case when discrete physical objects are modeled and simulated, since the accuracy of the simulated object increases as properties and behaviors are added and updated [13]. The net result is an increasingly accurate simulation based on real world observations, which then increases the knowledge of the simulated subject during simulation.

The benefits of using objects that simulate real objects are threefold. Since each object may go through some internal process or change, those modifications can be made on the individual object level. A degree of concurrency is achieved in the simulation, which is desirable since the real world often consists of many individual processes occurring simultaneously. Also, each simulated object can keep its own time, initiating, continuing, or halting the processes appropriate for that moment. Finally, statistical information particular to an object is maintained by that object, and can be retrieved at any time by that object's accessor methods [3].

Procedural programming languages are not as well suited for simulation, because procedures do not have the same correspondence to their real world analogues. Rather, procedures correspond to methods and algorithms [10].

3. METHODOLOGY

3.1 Computer Simulation Construction

Computer simulation construction is based, in part, on the scientific method employed in physical experiments. The steps can be divided into two groups: the formulation and refinement of the problem domain and model, and the implementation and evaluation of the simulation program. The steps in the first group consist of:

- the problem formulation
- actual problem domain data collection and analysis
- the formulation of the mathematical model
- evaluation to find if the mathematical model simulates the actual domain data to a necessary degree

If this last step finds that the model is insufficient, the steps in this group are repeated, or the experiment is scrapped if the domain is considered impossible to simulate [12].

The second group follows after the model has been approved. This group consists of:

- the implementation of simulation code
- that simulation program's validation [15, 18]
- the design of the experiments to be conducted on that computer simulation, and the analysis of the simulation's results (obviously this step is preceded by multiple runs of the simulation) [12].

These steps were taken for this experiment with the following modifications. First, the domain data was borrowed from measurements derived many times in laboratory experiments [6]. The first step of the implementation code was executed twice, once for the procedural code and once for the object-oriented code. The

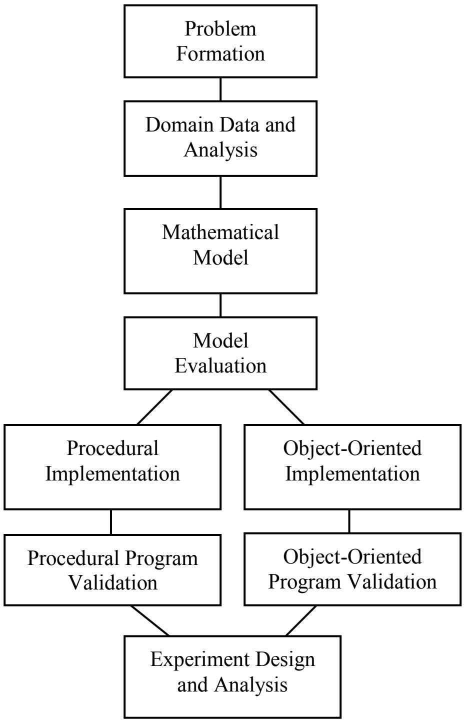

Problem Formation
Domain Data and Analysis
Mathematical Model
Model Evaluation
Procedural Implementation
Object-Oriented Implementation
Procedural Program Validation
Object-Oriented Program Validation
Experiment Design and Analysis

Figure 3.1.1. Procedural and object-oriented simulation experimental design

simulation program validation stage was also performed on each type of code, as were the analyses of their results. If these results were not identical, one or both simulations is inaccurate and this experiment is invalid. These differences, and the inclusion of the steps for simulation modeling and programming, are illustrated in Figure 3.1.1.

3.2 The Simulation Model

For this experiment, the behavior of a species of protozoa, the paramecium *P. aurelia*, is simulated (see Figure 3.2.1). For the simplification of the experiment (because the design is to test the programs' behaviors, not the behavior of the simulation), the protozoa are treated as existing in the relatively two-dimensional world of the microscope slide. This constraint dictates that the paramecia have behavior and responses that only occur in the plane of the slide.

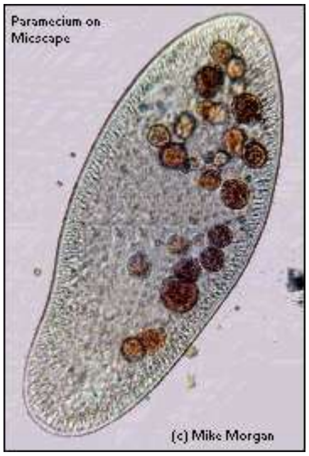


Figure 3.2.1. The paramecium *P. aurelia*

The simulated behaviors measured are the paramecia's food ingestion, reproduction, locomotion, and movement rates against the introduction of increased pH levels. These behaviors are a subset of the capacities of the actual microorganism, but are chosen as representing the main activities of the species.

The simulated paramecium is always foraging for resources. The food is randomly placed within the simulated slide. *P. aurelia* has only limited senses, and can only detect food (or anything else) in its immediate vicinity. Should the protozoan find food it its area, it will consume it. If there is no food accessible, *P. aurelia* will move linearly in search of more resources. This species cannot sense whether food is present at its destination until it arrives. After searching a specified interval without finding food, the paramecium effectively starves, and dies. If the paramecium ingests a sufficient amount, it will reproduce asexually. The parent manufactures a surplus of components, and then creates an offspring from this. If there is insufficient space to reproduce (the child can only be generated adjacent to the parent), the paramecium dies. The parent retains its original structures, and once an indicated age is reached, that paramecium dies [6]. *P. aurelia* is omnivorous (although not cannibalistic). The simulated microscope slide is a closed system, and when a paramecium dies it decomposes into potential food for other paramecia. However, the slide is also an entropic system, and the paramecium never returns as much energy to the system as it consumed.

3.3 Simulation Languages

After modeling and verification, the experimental simulation is coded in a procedural language, C, and in an object-oriented language, C++. These language choices were made based upon the success of these languages in simulation [9]. C and C++ are general-purpose languages. Special-purpose languages can be object-oriented as well as procedural, and comparison studies of special-purpose object-oriented simulation languages have also been undertaken [10]. The general-purpose languages used in this experiment were chosen because an experiment implementing these languages follows in the canon of comparisons described by Stensrud and Myrtveit [16].

Further, C++ and C share the same basis and implementation. The largest difference between the two languages is that C++ is essentially the language C with the additional capability to implement classes and objects. Therefore, the measurement of the price of object-oriented facilities can be more inductively derived with the comparison of C and C++ than with the comparison of two less related languages. In other more diverse language comparisons, the underlying implementation of non-object-oriented facilities might differ. Even simple operations could be implemented in different ways, which would in turn introduce another variance into time performance. Further, some objet-oriented languages use that orientation to implement features present in procedural languages. For example, Java (an object-oriented language) implements arrays as objects, instances of an Array class. In addition, the comparison of Java and FORTRAN-90 may have been unfairly tipped in FORTRAN's favor [4]. Java is a relatively new language when compared with FORTRAN's longer development, particularly when it is noted that much of FORTRAN's development was concerned with improving performance and memory management.

3.4 Application of Software Metrics

Each simulation is evaluated using software metrics, specifically Cyclomatic Complexity, Algorithmic Efficiency and Complexity, and Cohesion and Coupling measurement [7].

When originally implemented, the object-oriented simulation had a Cyclomatic Complexity of three, and the procedural simulation a complexity of four. For accurate experimental modelling, the object-oriented simulation was therefore effectively hamstrung to ensure that in this metric the simulations were equal.

The Algorithmic Efficiency and Complexity metric resulted in mediocre results. This relative deficiency was acceptable as long as the two simulations had the same algorithmic shortcomings.

Some metrics are applied to both procedural and object-oriented code, but applied differently. The measurement of Cohesion and Coupling are both extremely useful metrics, but have differing meanings when applied to objects.

Other metrics are specific to object-orientation, and differ from or are extensions of traditional software metrics. In particular, the Number of Children and Depth of Inheritance Tree metrics attempt to measure object instantiation and inheritance [2, 3, 8, 14].

Many metrics that can be applied to both procedural and object-oriented code without modification are problematic for this experiment. The Lines of Code and Comment Percentage metrics [14] in particular are not effective measurements of software quality in this experiment. The Anthropic Heisenberg Uncertainty Principle applies here, as these two metrics require external evaluation. The Comment Percentage metric should not be applied, as the programmer was aware that that metric would be used for the experiment. That awareness could have influenced how many or how few comments were included in the code as it was being implemented. The same problem exists with the Lines of Code metric, as well as the difference in lines of code measurement inherent with the two programming paradigms.

The goal for this portion of the experiment is to have the object-oriented and procedural simulations rate as closely as possible. Should the two simulations be too metrically diverse, optimizations will be included where necessary.

At the time of the experiment's execution, the Cyclomatic Complexity of both the procedural and the object-oriented implementations was four. The universal complexity was three cycles; if the condition that no food is immediately present is met, the function simulating movement added an additional cycle.

3.5 Execution of the Experiment

Both simulations were run on the same computer, with identical test values as input. The hard-coded input specified a volume of paramecia that must have been created. There was no input or output, save the time each simulation required, and that was measured and calculated outside of the simulations' execution. Each set of test data was input twenty times into each simulation, and the outputs resulting from those data were recorded and compared for mutual accuracy. This testing cycle was executed three times on different days at the same hour.

Each set of inputs into the simulations was recorded and charted separately with the rest of the input set. These results from each simulation were then compared to assure that each simulation was accurate. The execution time for each simulation for a particular set of inputs was also recorded to measure performance. Patterns, if any, were derived, and an overall relative performance prediction form was sought. It should be noted that computer simulations are usually tested hundreds (if not thousands) of times to find the appropriate pool of results and their probabilistic distribution. The experiment described in this paper is testing the behavior of such computer simulations as programs, not as simulations (see Graphs 4.3 and 4.5).

3.6 Limitations

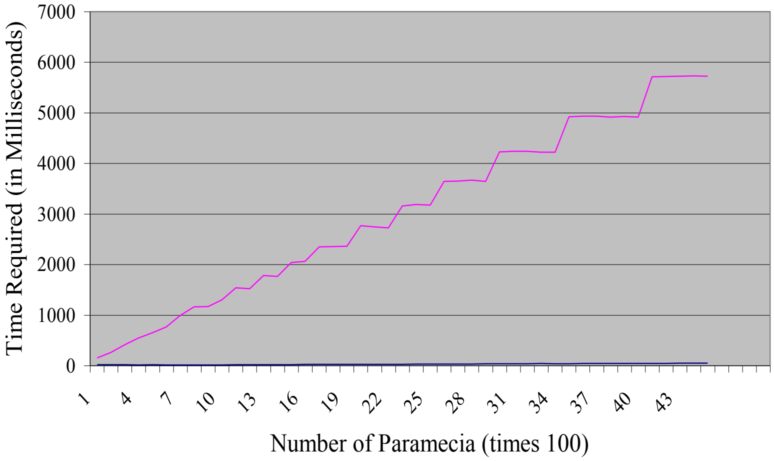
This experiment did no testing for maintainability on any of the simulation code due to time constraints. Maintainability is a significant metric and is often employed to determine software quality. Object-oriented code or procedural code could take the advantage in this area. In particular, one of the purported advantages of the object-oriented paradigm is that it requires less system maintenance and such maintenance is easier when it is necessary [11]. Further testing and experiments should be performed comparing object-oriented and procedural programs to determine which design strategy is more maintainable, and deriving a means of simply quantifying maintainability.

Another disadvantage of object-oriented programming that is often cited is that it requires increased development time [11]. Since the simulations in this experiment are simple programs, they did not require a long period of development. Further, since the simulations were modeled following the suggested methods, the part of the object-oriented development process concerned with the modeling of objects in the application's domain has already been partially completed. Any study of the development time would therefore be skewed.

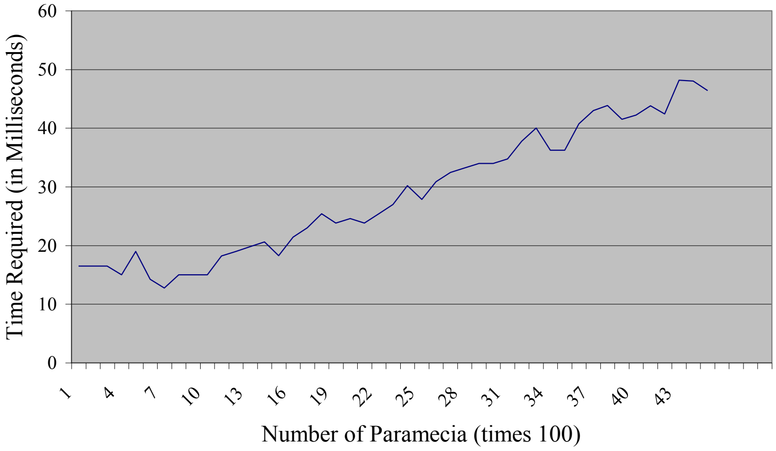
4. RESULTS

As anticipated, the procedural simulation performed in all cases in less time than the object-oriented simulation. Graph 4.1 displays the difference between the performances of the two simulations.

Graph 4.1. Procedural vs. Object-Oriented Simulation Performance



Graph 4.2. Procedural Simulation Performance

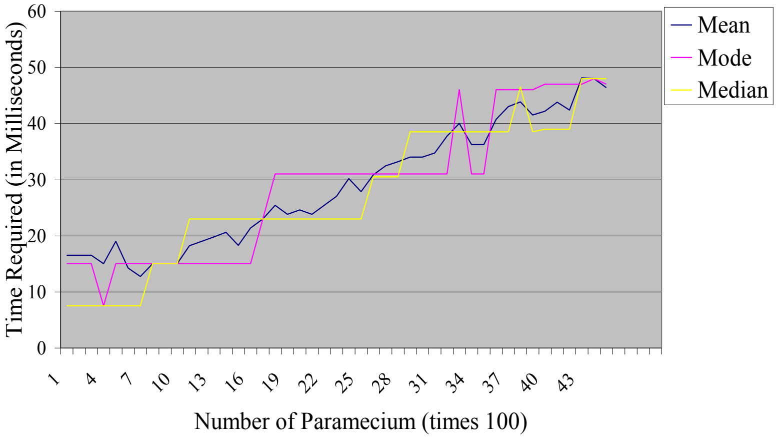


On average, the object-oriented simulation's execution time was two orders of magnitude greater than the procedural simulation (the procedural simulation's measurements are slightly distinguishable from the lower bound on the graph). Inductively, the conclusion can be reached that as the number of entities increases, the time required to instantiate those entities as objects will be greater than that required for the procedural implementation.

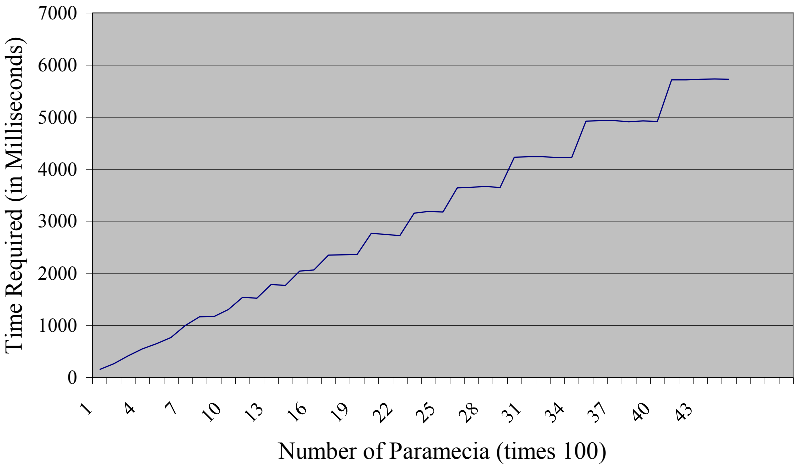
Further, *how* the two simulations performed is shown in their runtime performance. The procedural simulation consumed a certain amount time ad initio, and smaller populations of paramecia did not significantly affect this baseline cost (see Graph 4.2).

Aside from the initial cost of creating the slide and scattering food, the time required for the procedural simulation to generate a specific number of paramecia is a relatively linear function based on the input parameters. This is still relatively random, evidenced by the fact that each point on Graph 4.2 represents the average time over all executions of the simulation. Graph 4.3 shows how the average behavior of the procedural simulation accurately represents the general time requirements for the procedural simulation.

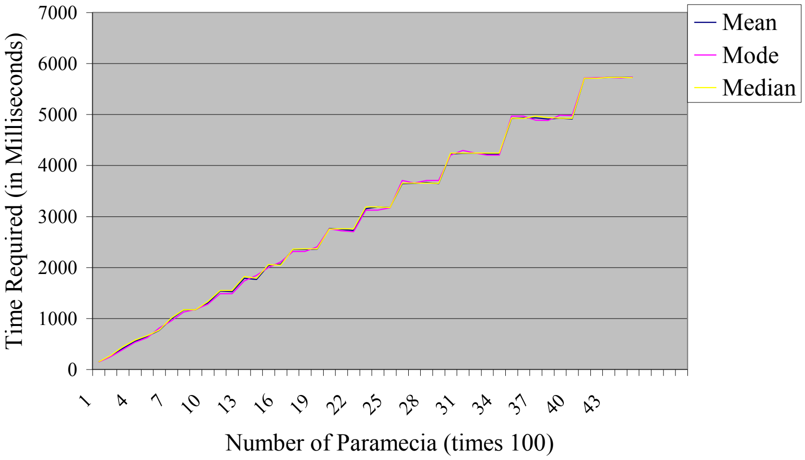
Graph 4.3. Procedural Simulation Performance: Mean, Mode, and Median



Graph 4.4. Object-Oriented Simulation Performance



Graph 4.5. Object-Oriented Simulation: Mean, Mode, and Median



The object-oriented simulation's behavior is much more indicative of the behavior of the paramecia as objects. The stair-step formation with increasing length and breadth in Graph 4.4 shows the reproductive behavior of the paramecia as generations. The majority of object-oriented paramecia tend to forage or reproduce as a group. This is derived from the fact that the required time increases more dramatically when new objects are being instantiated than when they are simply interacting. When more time is not required as the volume of paramecia increases, the paramecium objects are foraging collectively. Graph 4.5 illustrates the accuracy of the object-oriented simulation's average performance.

5. CONCLUSION

This experiment sought to discover whether an application involving an indeterminate quantity of objects has different time requirements than an application with a more static design. Computer simulations were chosen for this purpose due to their inherent unfixed nature.

Simulations also represent a subset of possible applications, and studies of their performance add to the corpus of information of the object-oriented and procedural paradigms. Unfortunately for the object-oriented paradigm, it is explicitly demonstrated that object-oriented simulations require much more time than procedural simulations. The intuitiveness of object-orientation is not called into question, merely its ability to perform efficiently compared to procedural programs. Therefore, while the paradigm seems to lend itself to computer simulations, object-orientation does not provide the most time-conservative execution of those simulations.

One conclusion reached before the experiment was conducted is that further study is necessary concerning the relationship between traditional software metrics and object-oriented software metrics. A determination is needed as to whether or not existing object-oriented metrics are equal to previous metrics, as well as what metric equality means. If no equality is determined, new metrics may be determined for use when both procedural and object-oriented programming are involved.

The immediate future of this study is to execute the same experiment on different platforms and under different conditions. These extensions should be simple to construct and execute since both C and C++ are largely machine-independent.

Similarly, an equal experiment should be undertaken that runs simulations procedurally and with objects. This next experiment should be implemented without the randomness of that experiment explained in this paper. The results of the first experiment should then be compared to those of this new second experiment. A corollary issue to this exploration is that a means must be found of comparing programs that generate an indeterminate number of objects with one that instantiates a fixed number of objects. The findings derived from this comparison will further the understanding of the effect of an indeterminate number of objects on the overall time performance.

This experiment can be easily extended into another area: the study of extensibility. The paramecia used in this experiment are ideal for extension. *P. aurelia* is capable of sexual reproduction as well as asexual. Although *P. aurelia* does not posses gender, two of these paramecia can commingle their (differing) genetic material, creating a new paramecium, genetically different than either parent. When *P. aurelia* reproduces sexually, the offspring is genetically identical to the parent.

An experiment could be constructed in which different paramecia are created based upon which particular parents breed. In this approach the object-oriented simulation would utilize the inheritance and polymorphism inherent to an object-oriented language. A procedural simulation is also possible to mimic this new form of reproduction. As the simulations are extended, the supposed benefits of object-oriented programming, notably extensibility, would be brought into play. However, the problem of how to measure extensibility remains before this new experiment can be considered valid.

Further research is called for in the area of computer simulations and object-orientation. Of immediate interest are the testing of extensibility and maintainability of object-oriented simulation code relative to its procedural counterpart. A worthwhile first step in researching object-oriented applications as a whole is finding the means to measure that extensibility and maintainability, as well as reuse.

REFERENCES

[1] Abernathy, K, and Kelly, J. C. “Comparing object-oriented and data flow models – A case study,” *Proceedings of the ACM 20th Annual Computer Science Conference*, 541-547, March, 1992. [ACM Digital Library]

[2] Berard, E. V. “Metrics for object-oriented software engineering,” The Object Agency, Inc, August, 1998.

[3] Bischak, D. P., and Roberts, S. D. “Object-oriented simulation,” *Proceedings of the 1991 Winter Simulation Conference*, 194-203, 1991. [ACM Digital Library]

[4] Budimlić, Z., Kennedy, K., and Piper, J. “The cost of being object-oriented: A preliminary study,” *Scientific Computing,* 7(2), 87-95, 1999.

[5] Coppick, J. C., and Cheatham, T. J. “Software metrics for object-oriented systems,” *Proceedings of the 1992 ACM Annual Conference on Communications*, 317-322, 1992.

[6] Dogiel, V. A. Revised Poljanskij, J. I., and Chejsin, E. M. *General Protozoology*. Oxford University Press. London, England, 1965.

[7] Fenton, N. E. *Software Metrics: A Rigorous Approach*. Chapman & Hall. London, England, 1991.

[8] Fetcke, T. “Investigations on the properties of object-oriented software metrics,” *Workshop on Quantitative Methods for Object-Oriented Systems Development, ECOOP '95,* Århus, Denmark, Aug. 7, 1995.

[9] Joines, J. A., and Roberts, S.D. “An introduction to object-oriented simulation in C++,” *Proceedings of the 1997 Winter Simulation Conference*, 78-85, 1997. [ACM Digital Library]

[10] Joines, J. A., and Roberts, S. D. "Simulation in an object-oriented world," *Proceedings of the 1999 Winter Simulation Conference*, 132-140, 1999. [ACM Digital Library]

[11] Johnson, R. A., Hardgrave, B. C., Doke, E. R. "An industry analysis of developer beliefs about object-oriented systems development," *The DATA BASE for Advances in Information Systems*. 30 (1): 47-64, 1999. [ACM Digital Library]

[12] Naylor, T. H., et al. *Computer Simulation Techniques*. John Wiley & Sons, Inc. New York, 1966.

[13] Pancake, C. M. "The promise and the cost of object technology: A five-year forecast," *Communications of the ACM*, 38 (10): 32-49, 1995. [ACM Digital Library]

[14] Rosenberg, L. H. "Applying and interpreting object oriented metrics," *Software Technology Conference.* 1998.

[15] Sargent. R. G. "Some approaches and paradigms for verifying and validating simulation models," *Proceedings of the 2001 Winter Simulation Conference*, 106-114, 2001. [ACM Digital Library]

[16] Stensrud E., and Myrtveit, I. "Measuring productivity of object-oriented vs. procedural programming languages: Towards an experimental design," *Proceedings of the 18th Information Systems Research Seminar in Scandinavia,* Gothenburg Studies in Informatics, Gothenburg Sweden, 7, 665-679, 1995.

[17] Vessey, I, and Conger, S. "Requirements specification: Learning object, process and data methodologies," *Communications of the ACM*, 37 (5): 102-113, 1994. [ACM Digital Library]

[18] Ziegler, B. P. *Theory of Modelling and Simulation.* John Wiley & Sons. New York, 1976.

Audio-Cued Computer Maze Traversal

Rachel M Noack
Computer Science Department
Winona State University
Winona, MN 55987
rachelnoack@hotmail.com

ABSTRACT

Nearly all computer games are played based on visual prompts and stimulations. This research project investigates one way that computer games can be played with only audio cues, no visual graphics. We examine how sounds can be used to enable players to traverse a maze by testing which sounds and types of cues are helpful for players.

General Terms

Measurement, Design, Experimentation, Human Factors, Theory.

Keywords

Audio, Blind, Computer Game, Cues, Maze, Sound, Vision.

1. INTRODUCTION

What comes to mind when you think of computer games? Have you ever considered playing one with your eyes closed? The essence of most games is sharp computer graphics and visual stimulations that you must pay careful attention to in order to play. Some online card games exist which are possible for players to play with no visual prompts, but little is out there for those who want to play an action or adventure game that does not contain any graphics.

The idea of a computer game with no visual cues may seem insignificant and unimportant, but it would be beneficial to many people in numerous of ways. First, there are 7.673 million people ages 15 and over in America alone with varying degrees of visual disabilities [10]. A game based solely on audio cues would allow persons with limited or no sight to participate in playing traditionally vision-oriented computer games. People with perfect eyesight might enjoy playing an audio cued game as well; gathering information that is conventionally visual by using auditory senses may provide an additional layer of difficulty to the game, and many players might find this to be a fun twist.

In order to create a game that is entirely driven by sound cues, it first has to be established that sound cues allow people to effectively navigate computer games. This research project tested to see if players were able to effectively traverse a maze—move from start to finish through a labyrinth of hallways—using audio cues as the only navigation guide. Although traveling through hallways is only a small portion of what computer games entail, it provides a good starting point to understanding the general mappings of visual context to sound.

The hypothesis of this work is that it is possible to add audio cues to a maze computer game that will enable players to navigate through hallways and doorways based on sound cues alone. Adding to this hypothesis, a humming noise that appears just before players approach doorways is the preferred audio navigation tool.

2. BACKGROUND RESEARCH

Much is still unknown about transforming visual figures into audio representations. Research being done in this area is with regards to making computer tools and applications more accessible for people who are blind or have low-vision [4, 7, 8, 9]. This is important because studies [3] show that only 13% of Americans with visual disabilities use computers regularly, compared with 51% of people without vision limitations. Researching ways to map non-text images to sound could help people with visual disabilities utilize an increased subset of computer applications, including games.

Computer games currently exist that are played with sound as the only navigation tool. Most of these games are card games or board game replicas, but a small subset of first-person action or adventure games are available [2]. Zform, one of the leading software producers of accessible games, is working to develop a 3D game environment that would be compelling and equally challenging for both sighted and non-sighted players. The technique used by Zform to signal doorways within the game is to emit a quiet hum from the center of all hallways that can be heard from adjoining rooms and hallways [1].

Companies that produce accessible computer games each use different types of audio cues and have different methods of indicating navigational signals. Although these companies may be successful in producing audio-cued software, research must be done to determine the underlying principles involved in visual to audio mapping.

One field of study, sonification, examines the use of nonspeech audio to convey information to a human listener. Sonification explores the psychological, perceptual, and cognitive aspects sound has on people.

Past sonification research has determined that human understanding of information in certain types of situations is increased by the presence of audio cues as opposed to the use of visual cues. Human hearing is especially receptive to sound changes over a period of time and can detect small changes in sound frequency [6].

3rd Annual Winona Computer Science Undergraduate Research Symposium,
April 2003, Winona, MN.

Some sonification researchers argue that sonifications allow humans to process greater amounts of data simultaneously than they could process with a visual presentation. A key aspect of this research involves determining the optimal amount of audio cues to provide users. If too many aural cues are provided, people become overwhelmed and the sounds become detrimental to providing data. Research is being performed to determine the optimal amount of audio cues to provide users by studying human memory and attention capabilities as well as other cognitive processes that influence aural data collection [6].

3. METHODS

To test the hypothesis, we modified an existing 3-dimensional maze environment displayed in a Java applet to include audio cues for navigation. Sounds transmitted through a set of speakers indicate to players where doorways to adjoining hallways are located. For instance, if there is a door located to the left of the player, audio cues are given through the left speaker to signal the entrance. Likewise, audio cues given through the right speaker are used to indicate doorways to the right of the player.

3.1 Visual Development

The original maze applet [5] is open-source code written in Java that is available to anyone who wished to use or modify it. The maze is 10 units long and 11 units wide. It contains few hallways and does not have a completion point or any way to “win” the game (refer to Appendix).

We modified this original code by making the maze bigger and by redesigning the hallways. The resulting maze is 25 units long and 25 units wide and contains a designated finishing point (refer to Appendix).

The applet design was also altered to allow players to navigate the maze by pressing the left, right, and up arrow keys on the keyboard instead of using a mouse to click on left, right, and up buttons.

After modifying the code to produce a new visual maze, the code was altered again to add audio cues.

3.2 Audio Development

We created four slightly different versions of the maze to determine which sounds and types of cues are most successful for players. The small variations helped to determine whether players prefer beeping or humming noises, and also how far in advance they like to know that a hallway is approaching.

- Versions 1 and 2 use a beeping noise to indicate where the doorways are located. As the player gets closer to the door, the beeps speed up becoming faster and faster. Then, when the doorway is directly next to the player, a bell sounds and the fast beeping continues.

- Versions 3 and 4 of the game use a humming noise to indicate where the doorways are. As the player gets closer to the door, the humming gets louder and louder. Then, just as in versions one and two, when the player is directly next to the doorway, a bell sounds and the loud humming continues.

- Versions 1 and 3 begin their sound cues as soon as a new hallway is discovered. The instant a hallway would be visible in the visual version, the audio signals begin. After the player chooses to either turn down the new hallway or chooses to continue down the original hallway, the sounds stop immediately.

- Versions 2 and 4 of the game do not indicate that a new hallway is available until the player is only a short distance away from the entrance. But similar to versions one and three, after the player chooses to either turn down the new hallway or chooses to continue down the original hallway, the sounds stop immediately.

In addition to adding sound cues to represent doorways, sound cues were put in place to let players know if they ran into a wall or if they successfully completed the game. When a player tries to advance forward but cannot because a wall is in the way, a beep—unique only to walls—is heard through both speakers. When a player completes the maze, cheering sounds are played to signify that the end has been reached. The audio cues for walls and game completion are the same in the four versions of the game.

3.3 Pre-Testing

Before formal testing took place, some pre-testing experiments were run. Four test subjects each played the four different game versions. They were timed to see how long it took to complete the mazes, indicating approximately how long the formal tests would take. On average, it took players 5 minutes and 3 seconds to complete the mazes. The pre-test participants also gave feedback regarding their preferences and suggestions. Three of the four players preferred the humming noise to the beeping. Three of the four also favored audio cues representing doorways to begin a short distance from entrance as opposed to far from the entrance.

3.4 Phase One Testing

In phase one of the testing process, the four audio-cued versions of the game were played by 32 test subjects. Each subject played all four variations and ranked them based on their personal preferences. A Complete Latin Square was used to determine the audio-cued version order in phase one testing.

1	2	3	4
3	1	4	2
2	4	1	3
4	3	2	1

Table 1. Complete Latin Square

The Complete Latin Square ensured that different test subjects attempted each game variation first, second, third, and fourth.

The first eight test subjects played the games in the following order: Version 1, Version 2, Version 3, and Version 4. The second eight test subjects played the games in the following order: Version 3, Version 1, Version 4, and Version 2. The third eight test subjects played the games in the following order: Version 2, Version 4, Version 1, and Version 3. The second eight test subjects played the games in the following order: Version 4, Version 3, Version 2, and Version 1.

Having players test the mazes in these orders ensures that the data is un-biased.

After examining the data gathered in phase one, it was determined that Version 4 is preferred by the majority of the players tested based on their rankings. Therefore, Version 4 became the final version of the game that was used in phase two testing.

3.5 Phase Two Testing

Phase two of the research project involved testing the selected audio-cued version of the game against the original visual-cued version of the game. To accomplish this, we used 40 new test subjects: 20 playing the audio version and 20 playing the visual version.

First, we examined what percentage of test subjects from each group were able to successfully complete the maze. Next, we considered the successful game attempts and calculated the amount of time it took each of the two groups to finish on average. Comparing these results helped to determine how successful the audio cues are at helping players traverse the maze.

4. RESULTS/ANALYSIS

In phase one testing, Version 4 was preferred by 12 of the 32 players tested. Version 2 was preferred by 10 of the players tested; Version 3 was preferred by 7 of the players tested; Version 1 was preferred by 3 of the players tested.

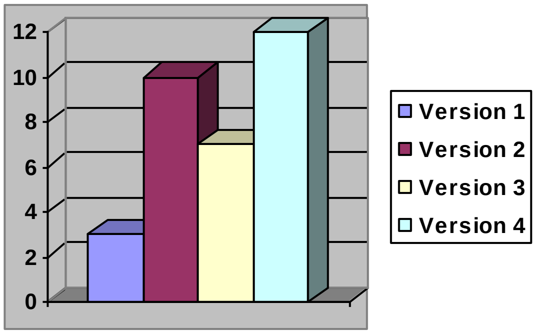


Figure 1. Phase one version preferences.

These results conclude that players prefer audio cues begin shortly before new hallway entrances are available instead of back further where a hallway would be visible in the visual maze version. This is determined because Versions 2 and 4, the two top ranking versions, both follow this method of audio cuing. In addition, players preferred the humming noise to the beeping noise 19 to 13. As you recall, Versions 3 and 4 used humming to indicate doorways, and Versions 1 and 2 used beeping to indicate doorways.

In phase two testing, 100% of the players in both groups completed the game. On average, the players who played the audio cued version took 8 minutes and 8 seconds to finish the maze. Players testing the visual version took 3 minutes and 47 seconds on average. Analyzing the times of the two testing groups showed the findings to be not statistically significant (P = 8.738686E-06, df = 33, t = 5.25327, two-tailed t-Test).

Although players testing the audio cued maze took just more than twice as long to complete the game as players testing the visual version, the hypothesis that it is possible to add audio cues to a maze computer game that will enable players to navigate through hallways and doorways based on sound cues alone still holds true. All players who tested the audio version of the game were able to complete the maze, with the longest testing time being 12 minutes and 31 seconds.

5. CONCLUSIONS

Although most computer games are played with the use of visual graphics, this research moves us one step closer to creating games that are navigated with audio cues. We examined how sounds can be used to enable players to traverse a maze by testing which sounds and types of cues are helpful for players.

The results of this study support the hypothesis that it is possible to add audio cues to a maze computer game that will enable players to navigate through hallways and doorways based on sound cues alone. All test subjects were able to complete the maze, regardless of whether they were given audio cues or visual cues. In addition, test results supported the hypothesis that a humming noise appearing just before players approach doorways is the preferred audio navigation tool. Although this research aids in gaining understanding regarding the general mappings of visual context to sound, there is further research to be done.

It is yet to be determined exactly how much sound is too much sound. At some point, too many sound cues become overwhelming and unhelpful to users. Further research should investigate the ideal balance of auditory cues to best enhance player navigation.

The research we have performed provides some insight about the user preferences that exist with regards to designing navigational cues for nonsighted computer games. To expand this idea, studies must be done to determine the underlying principles involved in all visual to audio mapping. The ultimate goal is to be able to follow a set of guidelines and principles to produce products and applications that always work well. It is not enough to develop individual successful projects; instead, a general understanding of sonification design principles is desired.

6. REFERENCES

[1] Anderson, G. Playing by Ear: Using Audio to Create Blind-Accessible Games. *Game Developer Magazine* [online] 2001 Oct. Available from: www.gamasutra.com. Accessed 2003 Feb 3.

[2] [Anonymous]. Game List. *Audyssey Magazine* [online] Available from: http://www.audysseymagazine.org/gamelist.htm. Accessed 2003 Feb 7.

[3] Gerber, E. and Kirchner, C. Who‟s Surfing? Internet Access and Computer Use by Visually Impaired Youth and Adults, *Journal of Visual Impairment & Blindness*, 95 (3), 176-181, 2001.

[4] James, F. Lessons from developing audio HTML interface. *Proceedings of the third international ACM conference on Assistive technologies*, Marina del Rey, California, United States, January 1998, ACM Press, New York, NY, 27-34.

[5] Jensen, J. Doughboy‟s Maze. http://home.globalcrossing.net/~jjens/maze.html. Accessed 2003 Feb 3.

[6] Kramer, G. et al. Sonification Report: Status of the Field and Research Agenda. http://www.icad.org/websiteV2.0/References/nsf.html. Accessed 2003 April 10.

[7] Mynatt, E.D. and W.K. Edwards. Mapping GUIs to Auditory Interfaces. *Proceedings of the 5th annual ACM symposium on User interface software and technology*, Monteray,

California, United States, 1992, ACM Press, New York, NY 61-70.

[8] Mynatt, E.D. and Weber, G. Nonvisual Presentation of Graphical User Interfaces: Contrasting Two Approaches. *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, Boston, Massachusetts, United States, April 1994, ACM Press, New York, NY 166-172.

[9] Rothberg, M. and Wlodkowski, T. Adapting Multimedia Software for Blind Students: Choices and Challenges. *Proceedings of the Technology And Persons With Disabilities Conference*, Northridge, California, United States, 1999. Available from: http://www.csun.edu/cod/conf/1999/proceedings/session0156.htm. Accessed 2003 Feb 7.

[10] The U.S. Bureau of the Census. Survey of Income and Program Participation (SIPP). 1999.

7. APPENDICES

The original maze is 10 units long and 11 units wide. It contains few hallways and does not have a completion point or any way to "win" the game. Players begin the maze in the cell marked by an „S‟ (refer to Figure 2).

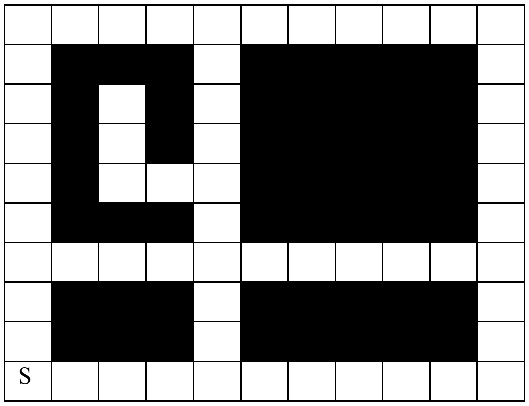


Figure 2. Original Maze.

The modified maze is 25 units long and 25 units wide and contains a designated finishing point. Players begin the maze in the cell marked with an „S‟ and finish the maze by reaching the cell marked with an „F‟ (refer to Figure 3).

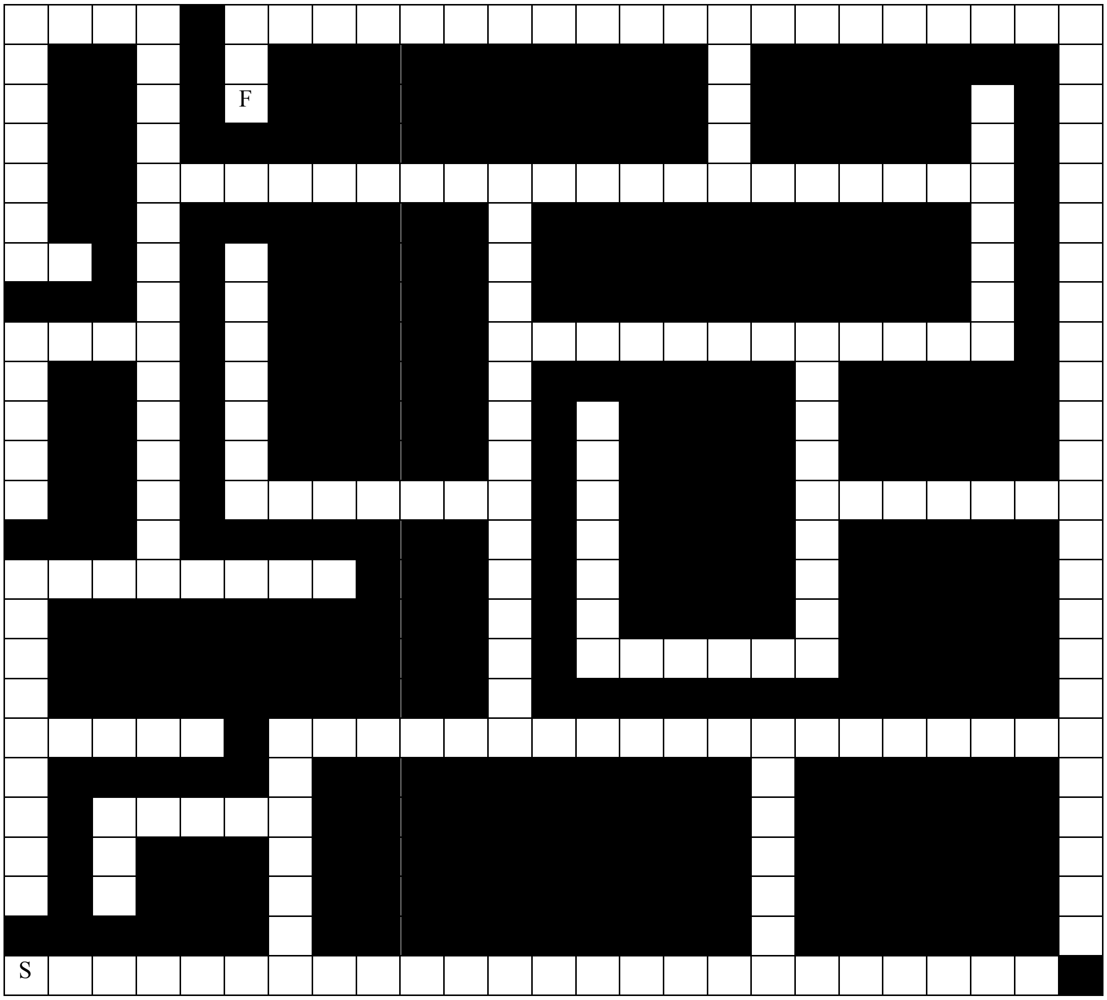


Figure 3. Modified Maze.

Optimizing GIS Web Delivery

Matthew C. Johnson
Saint Mary's University of Minnesota
700 Terrace Heights # 1350
Winona, MN 55987
507-457-7107
mcjohn99@smumn.edu

ABSTRACT

This paper details an exploration into the efficiency of web-based Geographic Information Systems (GIS). Systems designed to deliver GIS maps and data, such as ArcIMS (Internet Map Server), successfully provide a solution to delivering web-based GIS. However, the efficiency of these systems still leaves something to be desired. The process of delivering interactive spatial maps over the Internet is obviously hindered by the fact that modern networks are not designed to deliver such large data sets instantaneously. Since web-based GIS systems are comprised of a number of entities, such as a database, an application server, a web server, and a client browser, it is possible to tune these entities, particularly the database and application server, to optimize web-based GIS. Experimentation monitoring the effect database structure, size and layer properties have on web-based GIS performance and efficiency are described. Particular attention is given to the effect indexing a spatial database has on the efficiency of web-based GIS delivery. Our results show that ArcIMS websites are faster when their databases are indexed and the queries made to the databases are typical queries in general.

Keywords

GIS, ArcIMS, Database Efficiency, Schema, GIS Performance.

1. INTRODUCTION

Geographic Information Systems (GIS) is the integration of spatial and technical databases, and an application that graphically displays that data spatially. Web-based GIS delivery is a technology that is rapidly evolving with the continued growth of the Internet. ESRI (Environmental Systems Research Institute, Inc.), the developer and distributor of ArcIMS (Internet Map Server), claims to provide the only true distribution of GIS applications on the Web [1]. Providing in a graphical representation and an interface to view and manipulate that data. ArcIMS specifically provides the capability to deliver GIS over the Internet. Like any Internet delivery of large-scale information, the desirable and essential speed of this delivery is a byproduct of the efficiency of the system providing the service. Delivering such large datasets, converting them to a graphical format, and displaying them through a browser, clearly have major efficiency hurdles. Simply put, the time it takes to display a typical GIS map is more time than most Internet and or GIS users are willing to endure [2]. Specifically dealing with ArcIMS, numerous factors play a role in the efficiency of this system.

1.2 GIS Delivery Explained

GIS applications use a relatively structured process of delivering spatial data from database to a visual representation. Data is first requested either by the user or the application. A database housing this data is then queried, in most cases multiple times, and the requested data is returned. Then the application uses the delivered data to create a graphical or spatial display of that data. As stated previously, there are many factors that play a role in the sluggishness of web-based GIS. For example, because GIS is so dependant on the database that the data is being delivered from, the structure of the database plays a major role in efficiency. Structure plays a role in the efficiency in which GIS data is requested, retrieved and displayed. This paper explores and explains the severity these factors play in optimizing the efficiency of displaying spatial data on the Internet. Discovering and documenting specific performance factors involved in Web-Based GIS will benefit the advancement of entities using GIS applications and other similar systems that deliver large sets of data from a database through the internet. Professionals in such fields as Natural Resources, Business Administration, Public Administration/Local Government and Criminal Justice who all use GIS technology on a regular basis will benefit from greater optimization of GIS information delivery and access [3].

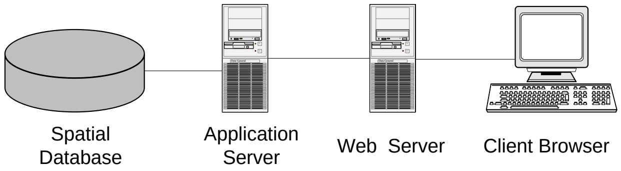


Figure 1. Web-Based GIS Structure

1.2.1 Layers

In most GIS applications, ArcIMS included, the map image that is generated from the spatial data is a set of layers. These layers are similar to various attributes on a map. For example consider a typical state road map. In GIS terms one layer would be the counties in that state. Spatially the information for each county is stored as a set of points and the lines connecting those points to create the polygon that is the county. This information is stored in a file called a shapefile. Another layer on a typical roadmap is the roads themselves. Also a shapefile this file contains sets of points connected by lines representing the roads within that state. Additional layers on the roadmap would be represented using

3rd Annual Winona Computer Science Undergraduate Research Symposium, April 2003, Winona, MN.

these basic concepts. Simply a layer is a shapefile or combination of shapefiles included with the databases associated with what each spatial feature represents. A layer may also be defined as an image, typically an aerial or satellite photograph. An ArcIMS site may have one to many such layers. Obviously, the more layers a site has, the more complex the site.

1.3 Controllable Factors

Besides relatively uncontrollable factors that play a role in most Internet applications, such as connection speed and network traffic, there are a number of manageable factors in ArcIMS. First, the size and structure of the database where the requested data is stored and queried from is a controllable factor. Second, the number and the size of active layers on the mapping interface. Simply, an active layer is the data available for querying. Further a visible layer is an active layer that is used to construct a graphical representation of that layer. The manipulation of these factors and analyzing experimentation results has lead to findings that will be applicable towards optimizing web-based GIS, ideally resulting in a set of quantitative performance results.

1.4 Hypothesis

Manipulating database structure schema, primarily through the use of indexing, will enable a Web-Based GIS system to deliver spatial data more efficiently with less loading time perceived by the end user. Further, changing the attributes of active layers and then analyzing the results will lead to findings that will be applicable towards the further optimization of web-based GIS, specifically the web-based GIS application ArcIMS.

2. Background Research

A limited amount of research has been done relating to this topic specific to GIS. Most research that has been done is relating specifically to the problems caused by the architecture of the systems designed to deliver web-based GIS, explaining the performance relationship between Database, Application Server, Web Server and Client Web Browser. Common conclusions have been client side caching and using supercomputers to serve the spatial data. Evidence that performance relative to GIS delivery is largely reliant on the machine processing the requests and delivering the data and image associated with that data. [4] One study in particular has proposed the use of tiles, a dataset extremely similar to a layer in ArcIMS. Further explored and explained was the effect using tiles has on indexing related databases. Because this method breaks the data into smaller subsections based on area and not by attribute association, there are specific strategies to indexing and transmitting the returned data [5].

2.1 ESRI Research

Most effected by the leaps and bounds made in the online delivery of GIS, ESRI, has done considerable research and analysis relating to the optimization of their products. Extensive research has been done on security issues with ArcIMS because of the sensitive or confidential data that might be shared using an ArcIMS solution. Most of such research has proven that security is optimized by a two-fold solution, first, the installation of a firewall between the database and application server, plus an additional firewall between the web server and incoming network traffic. Second, is the implementation of an alternate servlet engine to further customize the connection between application and web servers [6]. Research on the effects indexing has on a database is relatively cut and dry. The larger the database, the faster data is retrieved when a query is executed involving the indexed attribute [7].

3. Methodology

Several experiments have been implemented to test performance factors. These tests were designed with the intent of producing applicable, quantitative and representational results. The first set of experiments was designed to analyze the effects database structure and size has on performance. The intention was to create a set of representative map sites of varying size and complexity and then apply different database structures to them. Then querying each of the databases with the same representative queries producing quantitative results displaying the effects structure and size has on performance. Various factors were measured during these tests. Timestamps were recorded when a query was issued to when the query results were returned resulting in the time it took to process the query. Additional times are recorded in logs between the time an image is requested, processed and displayed, resulting in the time it took to build and display the image. Further results were taken recording the raw amount of data received and sent from request to request and the amount of time that it took for the requested data to be processed and displayed in the browser window. Using the net statistics the amount of bytes received and sent has been recorded from the terminal of the web browser. Relatively uncontrollable factors such as network traffic and connection speed have been minimized but were also recorded to show the effect these factors have on performance. By carrying out these experiments quantitative results have been achieved by measuring these various outputs. Recording and analyzing all of these statistics have resulted in a quantitative set of data for analysis and conclusion.

3.2 Database Manipulation

3.2.1 Experiment set 1 - A number of tests have been performed using a relatively small and focused dataset. The goal for this experiment was to use a low-resolution data set representing a broad set of information; representative of most generalized GIS applications on the web intended to give an overview of a given location. A site was set up using urbanized area spatial data in Central Europe. The layer properties were not tampered with. The specific layers used were shapefiles of the countries, urban areas and major bodies of water in Central Europe. The databases associated with these layers contain mostly identification and association attributes. The country database also contains various demographic data. A multiple number of database structures were applied to this set of databases such as multiple levels of indexing, and the same relevant and representative queries were made to the various databases for each structure change, producing results showing the specific results the variations on database structure had on the retrieval and delivery of spatial data. The database used to query in this experiment has 616 records.

3.2.2 Experiment set 2 - Similar to experiment set 1, a second set of experiments were performed with a much larger and increasingly focused dataset, representative of more specific internet map applications. The site created for this experiment was an interactive map displaying the epicenter of every earthquake recorded in the United States since 1568. There have been 1264 earthquakes since then so there are 1264 records in the database used for this experiment. Various layers were used in this experiment. Specifically, shape files of States and Urban Areas were used. Earthquake data was represented in a point file depicting the known or estimated epicenter of each Earthquake. Replicating the previous experiment set a multiple number of database structures were used. Specifically for the indexed database, two attributes were indexed, the month the earthquake took place and the day the earthquake took place. Also, similar to the previous experiment, the same set of representative queries was applied to each of the database structures, producing quantitative results for detailed analysis and conclusion.

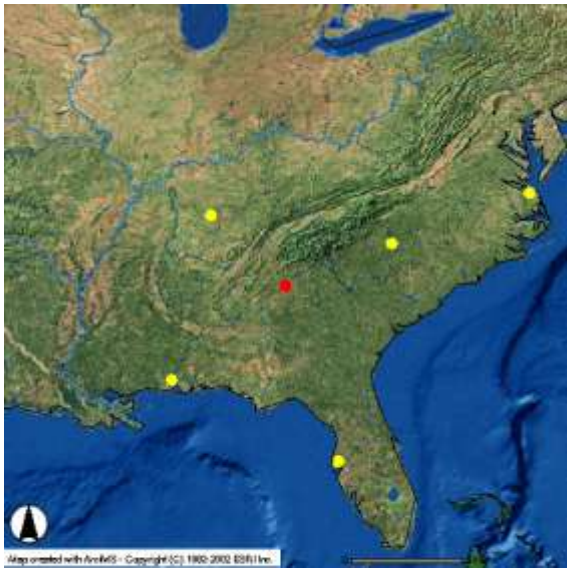


Figure 2. Sample ArcIMS Delivered Image from Experiment set 3

3.2.3 <u>Experiment set 3</u> – To model an actual real world application a third experiment was implemented using a site similar to a digital globe. The various layers used in this site were two shapefiles and an image. One of the shapefiles used depicts major cities of the world, the table associated with this shapefile has various fields besides the name and object ID of the city, such as population rank, member country, status and port ID. This layer was used extensively for query experiments. The second of the shapefiles used represented countries and their boundaries. The image file used in this map is most plainly described as a cloud free satellite image of the entire earth. The same experimentation methods as the previous experiments sets were applied to this site, producing more quantitative and detailed results. The database to be indexed used in this experiment has 2539 records.

3.2.4 <u>Experiment set 4</u> – After conducting the first three experiments a fourth experiment was designed and implemented with the intent of producing more evident results the effect that indexing a database has on the efficiency of ArcIMS. The previous experiments used large spatial databases. To be specific sizes of 2539, 616 and 1264 records were used in the previous experiments. For this final experiment a database of 65337 records was used to show the effects indexing a database has on a relatively huge database. Similar to experiment 3 a site with a map of the United States was used, however instead of a layer of earthquake data a layer of Census tracts. These tracts are best described as an area containing between 1,500 and 8,000 people. The entirety of the United States is broken up into these tracts for the collection and delivery of Census data. The indexed attribute used for this database was the attribute corresponding to the county each tract is in. Each county in each state is assigned a number from 1 to 840. As in the previous experiments the same experimentation methods were applied to this site.

4. RESULTS AND ANALYSIS

Due to availability of data and unforeseen data collection complexities the experiments did not take place in the same order mentioned in the methodology. Experiment set 3 was the first experiment to be implemented. Also Experiment set 4 was designed and implemented after the first 3 experiments had been conducted and briefly analyzed. The sites were served from a server with a 1000mb/s connection and the machine used as the client had a connection speed of 100mb/s. The connection rate was monitored and did not show a change over the entire period that the experiments took place. The experiments were also done when there was a 0 to 9% load on the machine serving the data and there were no internal network connections to the machine at any time besides the machine used as the client browser in the experiments. The first set of experiments to be explained is set 2 because of the quality of examples of data.

4.2 <u>Experiment Set 2 Results</u>

A total of 10 different representative queries were made to the site, each varying in complexity. Similar to the previous test there was no evidence that the site loaded faster or more efficiently with an indexed database during this particular experiment. The queried database or layer used in this experiment has 1264 records. The earthquake history table was indexed according to the month it occurred in and a second index was put on the day the earthquake occurred on. Specifically these attributes are labeled “MONTH” and “DAY” in the table. The averages of the results from this experiment are displayed in tables 2 and 3. Table 1 displays the average number of bytes transferred to and from the client machine during the various stages of the tests during experiment 2. The average data rate for this site with an un-indexed database was 106458.5 bytes received per second and 23679.79 bytes sent per second. The average data rate for this site with an indexed database was 181693.7 bytes received per second and 38972.69 bytes sent per second.

4.2.2 <u>Differences</u>

As displayed in the table there was an average of 2116.5 more bytes received and 1 more byte sent by the indexed database during the query process. The query itself took .0177 seconds less to send, .0159 seconds less to process on average. The indexed database received 4.9 more bytes and sent the same number of bytes on average while processing the image. The indexed database took .1943 seconds less to load the image. Overall the indexed database received 2120.4 more bytes, sent 2 more and took .2103 seconds less to complete the process. The indexed database received 75235.2 more bytes per second and sent 15293.2 more bytes per second. Every test during this experiment there was more data sent and received from the browser terminal.

4.2.3 Analysis

The results did show that the site was faster using an indexed database in every test. The total time for each test to execute was faster in 10 of 10 tests for the site with the indexed database. Also the amount of bytes sent and received per second was greater in every test using the indexed database. However, the query time was not always faster. In half of the tests the specific time for the query to send, execute and return was slower. The difference was an average of .01786 seconds, which was not great enough to skew the average results. This is evidence that indexing has an effect on more than just the querying of a database in an ArcIMS application. There needs to be further analysis done on this speculation as more data is collected when more experiments are completed. Even though, these experiments are not designed to test such factors. These results are further evidence that indexing a database increases the performance of an ArcIMS site.

4.3 Experiment Set 1 Results

The results from experiment set 1 have shown an improvement in speed with the indexing of their relating databases. A total of 9 different representative queries were made to the site, each varying in complexity. The site was also loaded separate times from a separate browser location, once with an un-indexed database and once with a database indexed. There was no evidence that the site loaded faster with an indexed database. The average data rate for this site with an un-indexed database was 33885.7 bytes received per second and 4083.53 bytes sent per second. The demographic table was indexed according to the country it is located in. Specifically this attribute is labeled "CNTRYABBR" in the table. The average data rate for this site with an indexed database was 33885.7 bytes received per second and 4034.514 bytes sent per second. The compiled average results for this experiment are displayed in tables 4 and 5.

4.3.2 Differences

Of the 9 queries made to the database, 8 proved to be faster and more efficient using an indexed database. The abnormal query skewed the averages for this particular experiment and will be explained in the analysis. There were an average of 1064.7 more bytes received and 6.22 more bytes sent by the indexed database during the query process. The query itself took .1646 seconds less to send on average. Overall the indexed database received 1067.3 more bytes, sent 12.44 bytes and took .0472 seconds longer.

4.3.3 Analysis

This experiment had very conclusive results excluding the one query that produced contradictory results and skewed the averages. 8 of the 9 queries ran faster on the indexed database. Those that were faster were an average of .1888 seconds faster. The query that was slower was .0288 seconds slower. The specific query that ran slower with the indexed database in this experiment was, "CNTRYABBR = „GB‟ OR (POPULATION > 7000000 AND FEMALES > 3500000)," Also an average of 2692.165 more bytes were received per second excluding the abnormal query and 300.124 more bytes sent. Clear evidence that the ArcIMS site setup to display a spatial representation of demographic data for central Europe was both faster and more efficient when the database containing the demographic data was indexed by the attribute "CNTRYABBR."

4.4 Experiment Set 3 Results

A total of 9 different representative queries were made to the site, each varying in complexity. The site was also loaded 4 separate times from a separate browser location, twice with an un-indexed database and twice with a database indexed. For each of those 4 loads an average of 1434196 bytes were received, 68491.17 bytes sent, and the average time was 25.888 seconds for the site to load to it‟s initial state. There was no evidence that the site loaded faster or more efficiently with an indexed database during this particular experiment. The average data rate for this site with an un-indexed database was 60722.8 bytes received per second and 1880.9 bytes sent per second. The cities table was indexed according to the population rank. Also a second indexed attribute was the country each city is located in. Specifically these attributes are labeled "POP_RANK" and "FIPS_CNTRY" in the table. This table was already in somewhat of an indexed state because each city is associated with the population rank of 1 to 7 depending on the size of the city. The average data rate for this site with an indexed database was 71033.73 bytes received per second and 1984.98 bytes sent per second. The compiled average results for this experiment are displayed in tables 4 and 5.

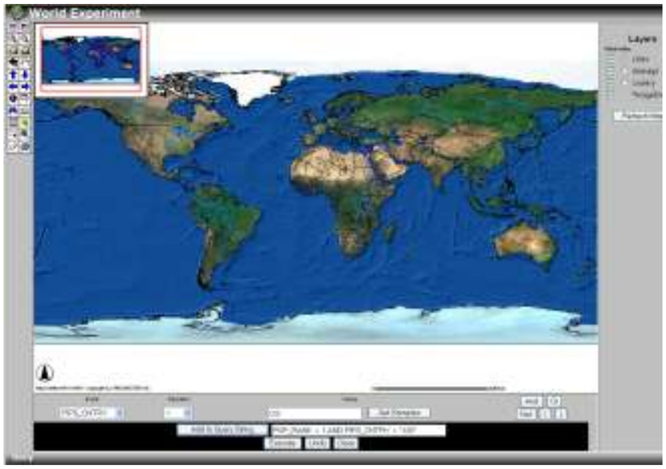


Figure 3. ArcIMS Interface for Experiment set 3.

4.4.2 Differences

There were an average of 135873 more bytes received and 99.9 more bytes sent by the indexed database during the query process. The query itself took .0245 seconds less to send, 1.0753 seconds less to process on average. The indexed database received 41735.1 more bytes and sent 905 less on average while processing the image. The indexed database took .43631 seconds longer to load the image. Overall the indexed database received 177608 more bytes, sent 1005.1 more and took .621 seconds less to complete the process. The indexed database received 10187.4 more bytes per second and sent 104.1 more bytes per second.

4.4.3 Analysis

The results did show that the site was faster using an indexed database. The tests have shown that 7 of 9 tests had better performance with the ArcIMS site that used the indexed database. However, the 2 tests that were not faster or more efficient did not cause a noticeable change in the average results. One of the less efficient tests was .0021 seconds slower with the indexed database. A fairly small difference compared to the other differences found during experimentation.

4.5 Experiment Set 4 Results

The results from experiment set 4 have shown an improvement in speed with the indexing of their relating databases. A total of 9 different representative queries were made to the site, each varying in complexity. The site was also loaded separate times from a separate browser location, once with an un-indexed database and once with a database indexed. There was minor evidence that the site loaded faster with an indexed database. The indexed database loaded .991 seconds faster. The average data rate for this site with an un-indexed database was 31312.2 bytes received per second and 3788.914 bytes sent per second. The tracts table was indexed according to the country it is located in. Specifically this attribute is labeled "CNTY_FIPS" in the table. The average data rate for this site with an indexed database was 25365.66 bytes received per second and 2999.522 bytes sent per second. The compiled average results for this experiment are displayed in tables 4 and 5.

4.5.2 Differences

Of the 9 queries made to the database, unlike the previous experiments, only two proved to be faster and more efficient using an indexed database. These abnormal results skewed the averages for this particular experiment and are to be explained in the analysis. There were an average of 2515 more bytes received and 157.33 more bytes sent by the indexed database during the query process. The query itself took .9104 seconds longer to send, .109 seconds longer to process on average. The indexed database received 1158.33 less bytes and sent 184.44 less on average. The indexed database took .43631 seconds less to load the image. Overall the indexed database received 1356.67 more bytes and took .9919 seconds more.

4.5.3 Analysis

This experiment should be viewed with a selective eye. During the experiment 6 of the 9 tests resulted in abnormal results. Only 3 of the queries ran faster on the indexed database. Those that were faster were an average of .7858 seconds faster. The queries that were slower were an average of 1.716 seconds slower. This can be attributed to the specific queries that ran slower. For example one of the queries that ran slower did not use the indexed attribute. Four of the queries that ran slower had a different form than previous queries. A specific query that ran faster with the indexed database in this experiment was, "CNTY_FIPS = „223‟," although this query is relatively simple compared to others in this experiment, this query was .1.599 seconds faster using the indexed database. An extremely similar query, "CNTY_FIPS < „065‟," was 1.953 seconds slower. Three other queries had a similar form, where the comparison of the indexed attribute, CNTY_FIPS, was greater than or less than a given number. Every query that had this form ran considerably slower on the indexed database, an average of 1.891 seconds slower to be exact. Excluding these 5 abnormal queries the remaining queries were an average of .325 seconds faster with the indexed database. Also an average of 4256.847 more bytes was received per second excluding the 5 abnormal queries. These results show that the "normal" queries ran faster and were more efficient transferring the spatial data.

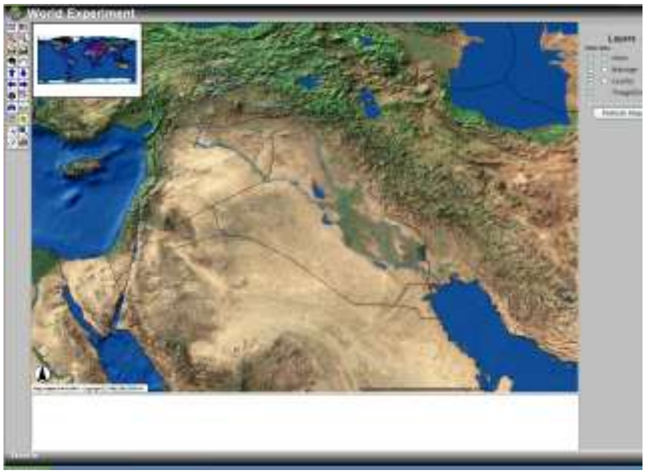


Figure 4. ArcIMS Viewer.

4.6 Combined Results and Analysis

After conducting the 4 experiments, 29 of the 37 tests have shown that an ArcIMS site using an indexed database is more efficient than a site using an un-indexed database. Excluding the 8 previously explained tests that resulted in abnormal results because of various reasons. The remaining results show conclusive evidence proving the basic hypothesis, that the use of indexing will enable a Web-Based GIS system to deliver spatial data more efficiently. These 29 successful tests were an average of .141805 faster. The number of bytes received per second was 23092.9 more and 4069.6 more sent.

Experiment Set	Query Difference	Received Rate Difference	Sent Rate Difference	Tests Included
1	-0.1888	2692.165	300.124	8
2	-0.01776	75235.2	15293.17	10
3	-0.03566	10187.42	104.1	7
4	-0.325	4256.87	580.834	4
Average	-0.141805	23092.91375	4069.557	

Table 1. Compiled Differences Excluding Abnormalities

5. CONCLUSIONS

The results have shown that as expected indexing spatial databases has positive effect on performance when implemented properly. However, the results have shown that the improvement in speed and efficiency is minor. The improved speeds were fractions of a second. In the general scheme of things this does increase performance. However, indexing a database increases the speed when pulling spatial information out of a database. Indexing a database creates significant problems when inserting or updating information in a database on the other hand. Future experiments should show and make more evident the effects indexing has on efficiency besides the query process itself. The results have shown that indexing the database does have positive effects on the performance in other areas than just the interaction of querying the database. These results should be further examined and explained.

The delivery of spatial data is still in early stages of development and evolution. As the demand increases for web-based GIS, further research will be done resulting in advancements of great proportions. Since spatial data is so large in nature, affordable technology to deliver spatial data as quickly as it is demanded is years down the road. Simply, the concept of pushing a grape through a straw is most applicable to what web-based GIS systems are designed to accomplish. The increasing demand and technology advancements in this field will result in further findings and studies similar to this exploration.

6. ACKNOWLEDGMENTS

I would like to thank the academy, Barry Drazkowski, Michael Montemayor, Erin Sperry, Chris Johnson, Jim Bloedorn, Patrick Thorsel, John Gabbert, Travis Ludwig and Dave Hajoglou.

7. REFERENCES

[1] ESRI. ArcIMS. *GIS for the Internet* [online] Updated 2003 Feb 6. Available from: http://www.esri.com/software/arcims. Accessed 2003 Feb 8.

[2] Harder, Christian. *Serving Maps on the Internet: geographic information on the world wide web.* ESRI (Environmental Systems Research Institute, Inc.), Redlands CA, 1998.

[3] Plewe, Brandon. *GIS Online: information retrieval, mapping, and the Internet.* OnWord Press, Albany NY, 1997.

[4] Tu, S., He, X., Li, X., Ratcliff, J.J. A systematic approach to reduction of user-perceived response time for GIS web services. In *Proceedings of ACM GIS '01,* Atlanta, GA, November 9-10, 2001.

[5] Wei, Z., Oh, Y., Lee, J., Kim, J., Park, D., Lee, Y., Bae, H. Efficient spatial data transmission in web-based GIS. In *Proceedings of the WIDM 99.* Kansas City, MO, 1999.

[6] Peters, David. *System Design Strategies.* An ESRI White Paper, February 2003. ESRI (Environmental Systems Research Institute, Inc), Redlands CA, 2003.

[7] Silberschatz, Korth, Sudarshan. *Database System Concepts: 4th Edition.* McGraw-Hill, New York NY, 2002.

Event	Un-Indexed Bytes Received	Indexed Bytes Received	Received Bytes Difference	Un-Indexed Bytes Sent	Indexed Bytes Sent	Sent Bytes Difference
Query Average	34529.5	36646	2116.5	7156	7157	1
Load Average	22544.1	22549	4.9	5540	5540	0
TOTAL For Query + Load	57073.6	59194	2120.4	12696	12697	1

Table 2. Experiment 2 Average Bytes Transferred Results

Event	Time1 Un-Indexed Seconds	Time 1 Indexed Seconds	Difference in Seconds	Time2 Un-Indexed Seconds	Time2 Indexed Seconds	Time 2 Difference Seconds	Total Un-Indexed Seconds	Total Indexed Seconds	Total Difference Seconds
Query Average	0.022333	0.004578	-0.01776	0.26733	0.26911	0.00178	0.289667	0.273689	-0.01598
Load Average	0.246444	0.052101	-0.19434				0.264440	0.052101	-0.19434
TOTAL For Query + Load							0.536110	0.325970	-0.21032

Table 3. Experiment 2 Average Time Results

Event	Un-Indexed Bytes Received	Indexed Bytes Received	Received Bytes Difference	Un-Indexed Bytes Sent	Indexed Bytes Sent	Sent Bytes Difference
TOTAL For Experiment 1	123214.0	124281.3	1067.3	14848.4	14860.8	12.4
TOTAL For Experiment 2	57073.6	59194.0	2120.4	12696.0	12697.0	1.0
TOTAL For Experiment 3	1305771.0	1483379.0	177608.0	40446.8	41452.0	1005.1
TOTAL For Experiment 4	125346.2	126702.8	1356.6	15167.4	14983.0	-184.4

Table 4. Compiled Average Bytes Transferred Results

Event	Query Average Un-Indexed Seconds	Query Average Indexed Seconds	Difference in Seconds	Total Un-Indexed Seconds	Total Indexed Seconds	Total Difference Seconds
Total for Experiment 1	0.174622	0.009995	-0.16463	3.636178	3.683439	0.047261
Total for Experiment 2	0.022333	0.004578	-0.01776	0.536110	0.325970	-0.21032
Total for Experiment 3	0.045111	0.009450	-0.03566	21.503780	20.882740	-0.62104
Total for Experiment 4	0.239556	1.150000	0.91044	4.003111	4.995056	0.99194

Table 5. Compiled Average Time Results

Emergent Behavior in Multiagent Systems

Alexander U. Berezhnoy
Computer Science Department
Winona State University
Winona, MN 55987
alexanderub@hotmail.com

ABSTRACT

Multiagent Systems (MASs) is a recent but widely recognized subdiscipline of Artificial Intelligence (AI). An MAS is a collection of intelligent software agents that coordinate to achieve certain goals. One of the most interesting aspects of an MAS is emergent behavior. Emergent behavior is that which is not attributed to any individual agent, but is a global outcome of agent coordination. We explore emergent behavior in MAS to try to find the ways in which it can be initiated and controlled. We discuss why emergent behavior has useful implications in AI research that is aimed at the modeling of natural intelligence. During this initial research we create an MAS that is capable of emergent behavior. First we describe the essence of the system and show why it can be considered a legitimate MAS. Then we present the behavior exhibited by the system during the experiments and corroborate why it should be considered emergent.

KEYWORDS

Emergent behavior, multiagent systems, agents, artificial intelligence.

1. INTRODUCTION

1.1 Multiagent Systems

The study of intelligent agents and especially Multiagent Systems is one of the most recent and promising branches of Artificial Intelligence [3]. According to Victor Lesser, "multiagent systems are computational systems in which several semi-autonomous agents interact or work together to perform some set of tasks or satisfy some set of goals" [2]. A more generalized definition is that a multiagent system is "a loosely coupled network of problem solvers that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver" [5], where "problem solvers" refers to agents, which can be both homogeneous and heterogeneous. The radical difference of this subdiscipline from its parental field is that the decision-making is not centralized, but distributed. Therefore, the true power of multiagent systems is not contained within individual agents, but it is an outcome of their coordination, which can take forms of both cooperation and contention [6].

1.2 Emergent Behavior

There are many aspects of this relatively new field, each with its challenges, issues and mysteries. One that interests us the most, and is the topic of this research, is emergent behavior of multiagent systems. The term "emergent" is not without ambiguity. By the term "emergent behavior" in the context of this research we refer to the behavior of a multiagent system that is not attributed to any individual agent, but is a global result of coordination between multiple agents. Note, that no individual agent has a capability of exhibiting such behavior; it can only occur as a "joint effort." Most emergent behavior can be deduced from the set of actions that agents are able to perform. However, in scenarios where agents have a wide range of functionalities available to them and where the coordination among them is flexible, emergent behavior cannot always be predicted with high certainty. One might wonder: how good is emergent behavior that is unpredictable and what can it be used for? The answer to this question takes the form of a philosophical elaboration, which we summarize in the next section.

1.3 Implications of Emergent Behavior

The primary goal of Artificial Intelligence is to artificially create intelligence equivalent to natural human intelligence

3rd Annual Winona Computer Science Undergraduate Research Symposium, April 2003, Winona, MN.

[3]. We strongly believe that one of the prominent ways of achieving artificial intelligence is by closely modeling the natural one. Since human intelligence is mainly a product of the brain and the environment in which it resides, it is reasonable to state that our intelligence can be seen as an emergent behavior of our brain cells. In fact, if we take a low level view of our bodies, they can be regarded as systems of coordinating cells. There are many different kinds of cells, all with their own tiny tasks and goals, yet the outcome of their interaction and organization is strikingly amazing. However, let us not stop here, because cells can be viewed as complex interactions of the molecules that make them up. The molecules are in turn made up by coordinating atoms, and atoms can also be broken down into different elementary particles. At the lowest level of view, it is precisely these elementary particles that comprise everything, including humans. [4,7,9]

The behavior of these basic building blocks can be explained by the laws of quantum mechanics. According to the theory of expanding Universe, the Universe initially started from an infinitely hot and dense state, and shortly after its expansion (10^{-10} seconds) it primarily consisted of elementary particles. After a while these particles formed atoms, which later linked into molecules, then came the first cells, and eventually, us. But do the laws of quantum mechanics predict the formation of complex molecules like DNA? Or do the laws of chemistry predict that molecules organize themselves into the living cells? For that matter, looking a few billion years back at the first cells populating the Earth, could we have predicted what their interaction would result in? Perhaps physicists, chemists and biologists have incorrect theories or their laws are incomplete. But what if not everything can be predicted from the basic laws, and uncertainty of emergent behavior played an important role in our development? And if so, could it be possible to create a software simulation of the cells and their environment so precise that their interaction would result in the formation of more complex entities? These ideas are the reasons why we are interested in studying the unpredictability of emergent behavior and its potential benefits. [4, 7-9]

1.4 Achieving Emergent Behavior

Before we can start researching emergent behavior it would be meaningful to first verify that it is possible to achieve. Therefore, the goal of this research is to demonstrate the following:

Given a simple, but fully functional multiagent system and the environment in which it situates, the interaction among the agents will result in an emergent behavior of the system.

By fully functional multiagent system we consider the system that adheres to the following Guidelines established by the Multiagent Systems researchers at Carnegie Mellon University [5]:

1) each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint;
2) there is no system global control;
3) data are decentralized;
4) computation is asynchronous.

Based on the principles above, we designed a software simulation that can be regarded as a multiagent system. It is a virtual 2-dimentional environment that can be populated with heterogeneous agents. Each agent has a set of capabilities that it can perform in the environment. These agents can also interact among themselves and influence one another"s behavior. As the overall system state changes, it can be monitored through a graphical display. Later in the paper we analyze the results obtained from system monitoring and argue that some of the observed behavior can be considered emergent.

2. BACKGROUND RESEARCH

There is a tremendous amount of research taking place in the field of Multiagent Systems (MASs). It is conducted in the areas of agent architectures, communication protocols used between agents, models of agencies, agent-based software engineering and many others [1]. One of the originators of the idea of using multiagent approach in Artificial Intelligence, Marvin Minsky, based many of his theories of human mind on societies of agents and their complex interactions [4]. Among the high volumes of information available about MAS we encountered only brief mention of emergent behavior [1,3,4,6]. It currently seems to be regarded more as a topic for science fiction deliberation; but so were once television, airplanes and thermonuclear weapons. Scarcity of studies closely related to emergent behavior in MASs only increased our interest in the matter.

3. METHODS

It proved very difficult to decide what our MAS should simulate. Various domains ranging from colonies of insects to energy exchange in cell mitochondria were considered. All of them were eventually rejected due to not apparent, yet present complexity. Finally we decided that it would be sufficient to simulate a very simple, imaginary world based on rules that we dictate ourselves. The ideas from Swarm

Development Group [10] served as a direct inspiration to the creation of the world of Warmers and Chillers.

We chose the Java language as the tool for implementing our multiagent system. There were several reasons for this decision:

- Since agents can be viewed as objects, they naturally map into the object-oriented paradigm;
- Java comes with the system development kit, which provides numerous helpful features and reusable data structures;
- Java has built-in support for threads, which are imperative in asynchronous computing.

3.1 Environment

The simple world of Warmers and Chillers is represented by a virtual two dimensional plane (see Figure 1). The plane serves as an environment in which agents function. It has only one global property – its dimensions.

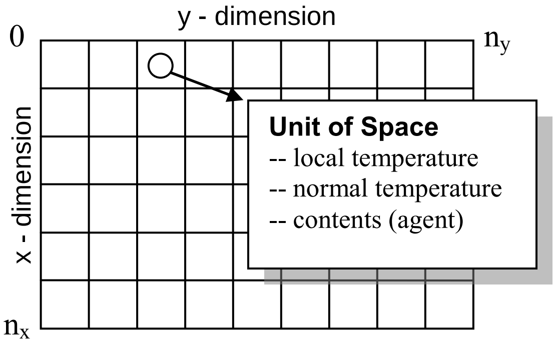


Figure 1. Environment

The plane is broken up into discrete units of space. Each unit can contain a zero or one agent at any given time. Apart from agents, every unit of space is capable of storing local information about the properties of the environment, more specifically, its local temperature (as you might have guessed from the name of the world). Besides the local temperature every unit also has normal temperature associated with it. If the local temperature ever deviates from normal, it is then gradually changed back to normal, just like an ice-cube melts in your hand, or a cup of tea cools down sitting on the kitchen table. We refer to this process as normalization. Note that we decentralize the environmental temperature data in order to comply with *Guideline 3*.

We mentioned that sometimes local temperature can deviate from normal. This happens due to the activity of agents populating the environment.

3.2 Agents

There are two types of agents populating the environment – Warmers and Chillers (W&C). These two types of agents are very much alike. In fact, in the implementation of the system they both are the descendants of one common ancestor – Thermal Agent. The radical difference between the two types of agents is that they prefer and seek completely opposite conditions, and affect the environment in complimentary ways:

- **Warmers** – always try to get into the parts of the world with higher temperature and warm up the environment by generating heat.
- **Chillers** – seek lower temperature and cool off the surroundings.

Both Warmers and Chillers have a limited view of the environment in accordance with *Guideline 1*. Depending on their sensor range, which is inversely proportional to an agent‟s degree of satisfaction (Warmers are more satisfied when they are warmer, opposite for Chillers), they can examine the properties and contents of nearby units of space. There is no centralized control over agents‟ actions; they all act autonomously. Each agent is implemented as a separate thread, so their decision making is asynchronous. Hence *Guidelines 2* and *4* are also followed. Therefore our system adheres to all four principles and can be considered a legitimate multiagent system.

3.3 System Monitoring

The state of the system at any given time is represented by the properties of each individual unit of space and also by the states of all present agents. The system state is outputted to the screen as an animation. Each image comprising this animation is generated by a special Monitor agent. Every time Monitor agent gets its turn to run, it temporarily blocks all other agents and takes a snapshot of a system, this essentially constitutes a system state change.

Figure 2 shows a sample of a system monitor. This basic monitoring facility enables us to observe the environment as it dynamically changes. In order to preserve the evidence of what went on during the simulation, the system can also be commanded to save an image of its state on to secondary storage. This can be done interactively (simply by clicking on the system monitor window), or the system can be configured to save images periodically.

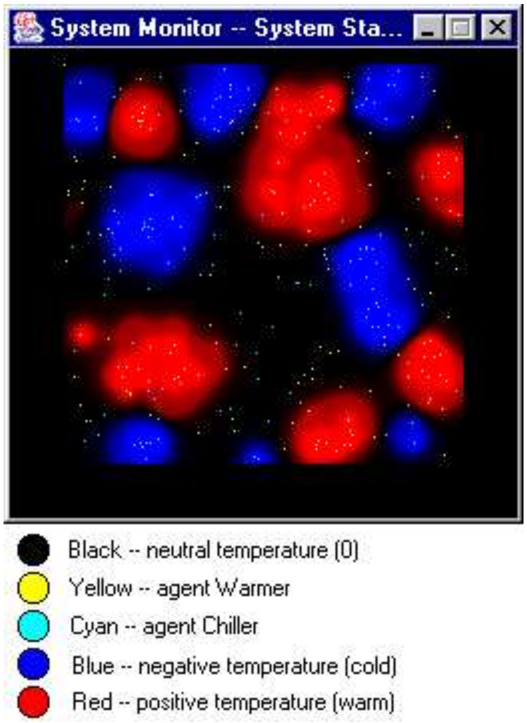

System Monitor -- System Sta...
● Black -- neutral temperature (0)
● Yellow -- agent Warmer
● Cyan -- agent Chiller
● Blue -- negative temperature (cold)
● Red -- positive temperature (warm)

Figure 2. System Monitor and legend

3.4 Experiments

Once we built this simple multiagent system, we were able to experiment with it to see if it was capable of exhibiting behavior that could be considered emergent. We established the following test scenarios:

Size in units	Number of Agents	Type of agents and probability of appearing	#
100 by 100	50	Warmers 1.0	1
		Chillers 1.0	2
		Warmers & Chillers .5/.5	3
400 by 400	50	Warmers 1.0	4
		Chillers 1.0	5
		Warmers & Chillers .5/.5	6
	1000	Warmers 1.0	7
		Chillers 1.0	8
		Warmers & Chillers .5/.5	9
		Warmers & Chillers .875/.125	10

For every test scenario we populated the environment with a random number of agents according to the probability ratio indicated in the table. The starting locations of agents were also randomly chosen, which resulted in fairly even scattering.

We want to point out that there are many factors that influence the behavior of the agents. Some of the most important ones are listed below:

- **Temperature propagation** – how far and how much does the temperature propagate from its source
- **Temperature intake** – how much environmental temperature affects an agent and vice versa
- **Normalization** – how soon a given space unit changes back to its normal temperature after being affected by agents

During this set of experiments all varying factors were kept the same.

4. RESULTS AND ANALYSIS

The simulations turned out to be rather computationaly intensive and, as a result, slow! We had to do a lot of optimizations of the system before we were able to achieve tolerable performance. Of course the intensity of computation increased with the size of the environment and the number of agents in it, since every agent is a separate thread. Another issue that arose was the length of the runtime for each simulation. Since we did not define an end for the world of Warmers and Chillers, at which point do we terminate a simulation? The latter issue was resolved as a very obvious and steady pattern appeared in agent behavior. But before we make any claims of emergent behavior we want to precisely state what Warmers and Chillers were programmed to do.

4.1 In Agent's Shoes

To understand what exactly an agent is capable of, imagine for a moment that you are a Chiller. The next few sentences describe the cycle of your actions. First you point your temperature detector at random direction and scan as far as it currently allows you. The hotter you are the further your detector reaches. The detector indicates in which direction it located the coldest spot. You attempt to move in that direction. If something is blocking your way you try to get around it. As soon as you enter the new space unit, it warms you up or cools you down more. Your detector adjusts its range accordingly. And lastly you send out whatever chill you have into the environment. At this point the cycle of your actions restarts. You are alone in this world and your only ambition is to be as cold as you can get.

That was a view of an agent (similar, but warmer rules apply for Warmers). Next is the description of what we get to see as observers.

4.2 Agent Clustering

Every single simulation resulted in some form of agent clustering. It appears as if agents like to be in groups with the same kin. However, we know that agents are not aware of each other except when they block each other‟s way. All they can sense is the temperature around them and they seek the place where it is most suitable for them. Figure 3 depicts the evolution of the system from experiment #3 (numbers represent system states).

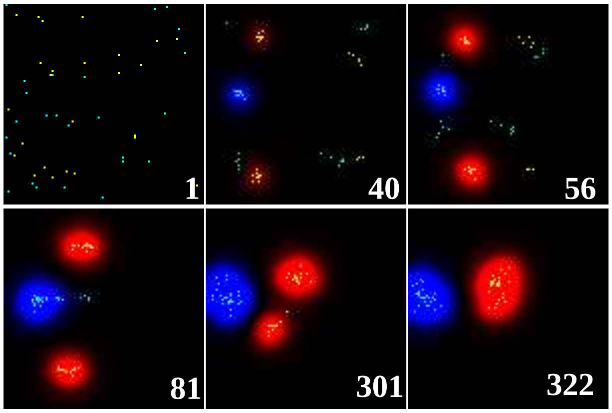


Figure 3. Data from experiment #3

During this experiment Warmers and Chillers formed two clusters. Such behavior proved to be the typical outcome in small environments. Experiments #1 and #2 resulted in complete clustering of Warmers and Chillers respectively.

Things turned out a little different for the same number and ratio of agents in a larger environment, as experiments 4-6 showed. Complete clustering never took place. Small clusters that formed and the temperature that they emitted took circular shapes. Some clustered agents were not even able to build up enough temperature to show up on display, and some were left singled out.

In the experiments 7 through 10 we increased a number of agents, and things took another turn. Clustering still occurred, but at a much steadier pace. The table below shows the progress of clustering in experiment #8.

System State Number	Number of Clusters	System State Number	Number of Clusters
301	26	7,101	10
1,201	20	12,001	9
4,001	11	15,001	8

The above results showed that clustering was gradually occurring. At first we thought that if we gave the system enough time, complete clustering would take place. To our surprise we observed another interesting global outcome of agent behavior.

4.3 Cluster Splitting

Clusters of agents merged and grew bigger. The bigger they got, the less circular their appearance became. From time to time larger clusters would stretch, bend and take irregular shapes. Soon the way they moved through the environment became amoeba-like.

Then we noticed that these amoeba-shaped clusters would sometimes stretch so much that it then resulted in splitting. Figure 4 shows amoeba-shaped clusters and their merging and splitting.

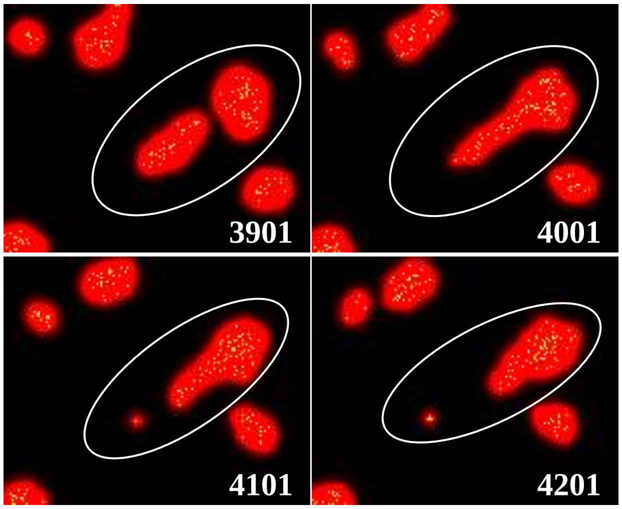


Figure 4. Scaled down fragments of experiment #7

The rest of the planned experiments did not result in anything new.

Even though we were not originally intending to do this, we decided to conduct an additional experiment related to cluster splitting. Instead of randomly populating the environment we now placed 1000 homogeneous agents very close to each other expecting them to immediately form one large cluster, which we would be able to observe. Figure 5 shows exactly what happened.

Initially one large cluster formed, it then soon started splitting and by the time we finished simulation we had two independent clusters.

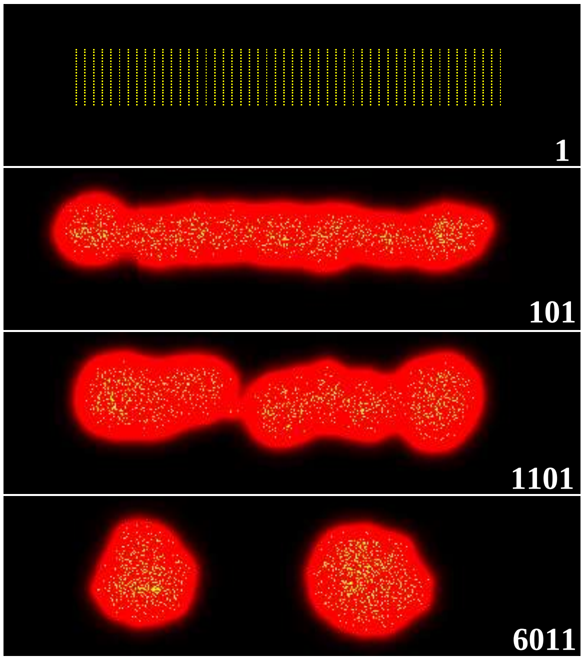


Figure 5. Scaled down fragments of splitting exp

5. CONCLUSIONS

5.1 Emergent Behavior Achieved

Recall a limited view of an agent (section 4.1) and juxtapose it against the results of the experiments we conducted. The only thing that Thermal Agents do is seeking more comfortable places in the environment. They are not even fully aware of each other. Circular or amoeba-shaped clustering formations and joint movement through the environment were not programmed into them. Cluster splitting came up as something rather unexpected. Therefore some of the behavior exhibited by this multiagent system should be considered emergent.

5.2 Creating Intelligence From Non-Intelligence

The multiagent system we created has no practical use, nor does it solve any real world problems. However, it is an example of how something that appears to be somewhat intelligent can be simply an outcome of its less intelligent parts. In the case of our system, very narrow-minded agents came together as more sophisticated entities without ever trying to do it or knowing about it. Analogous to that, are the cells comprising human bodies aware of what they are a part of? Or are they simply acting on their own behalf in their own world of cells and we are just an accidental occurrence transparent to them, similar to clusters being transparent to the agents in the system that we described?

We believe that the study of emergent behavior of multiagent systems will contribute significantly to the field of Artificial Intelligence. Undoubtedly, some day a truly intelligent artificial entity will emerge from many much less intelligent, but well coordinated elements; and hopefully such emergence will be planned, not accidental.

6. REFERENCES

[1] Huhns, Michael N., and Singh P. *Readings in Agents*. Morgan Kaufmann, San Francisco, CA., 1998.

[2] Lesser, Victor R. Multiagent systems: an emerging subdiscipline of AI. *ACM Computing Surveys v. 21, n. 3*, 340-342, 1995.

[3] Luger, George F. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Pearson Education Limited, Harlow, United Kingdom, 2002.

[4] Minsky, Marvin. *The Society of Mind*. Simon and Schuster, New York, NY., 1985.

[5] Sycara, Katia P. Multiagent systems. *AI Magazine* 1998 Summer; 79-92.

[6] Weiss, Gerhard. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA., 1999.

[7] Hooft, Gerard „t. *In Search of the Ultimate Building Blocks*. Cambridge University Press, Cambridge, United Kingdom, 1997.

[8] Hawking, Stephen. *A Brief History of Time*. Bantam Books, New York, NY., 1996.

[9] Purves, et al. *Life: The Science of Biology, Sixth Edition*. Sinauer Associates, Inc., 2001.

[10] Swarm Development Group. http://www.swarm.org