Integrating Educational Resources and Curricula through Interactive Software: A Community-based Case Study

Shana Fliginger
Department of Mathematics and Computer Science
SDSMT
501 E. St. Joseph
Rapid City, SD 57701
sefliginger@rushmore.com

James Tang
Department of Mathematics and Computer Science
SDSMT
501 E. St. Joseph
Rapid City, SD 57701
jamest85@hotmail.com

Dr. Manuel Penaloza
Professor of Computer Science
SDSMT
501 E. St. Joseph
Rapid City, SD 57701
Manuel.Penaloza@sdsmt.edu

Abstract

In order to bridge the digital divide while enriching the education of children, parents and teachers, this project builds upon the conceptual uniqueness of the Children's Science Center (CSC) in Rapid City, South Dakota to bring an interactive learning opportunity to area residents and augment interest in local paleontology and geology via the World Wide Web.

This paper describes the integration process while focusing on the design and implementation of software developed using the Microsoft .NET framework for stand-alone workstations. Through the creation of these learning centers, the project combines educational resources, curricula and recent computer technology to provide a conducive venue for learning.

Introduction

The South Dakota School of Mines and Technology counts the Children's Science Center as one of its most important community outreach programs. The CSC strives to foster positive attitudes about mathematics and the sciences by providing informal learning experiences and serving as a resource center for parents, educators, children and other members of the general public. This concept matches the philosophy of the Rapid City Area Schools (RCAS) as stated in the *K-5 Science Curriculum*, released November 2001, "In the elementary grades, students will experience hands-on opportunities that encourage the active construction of scientific ideas, knowledge, and explanations."[1]. Therefore, a project benefiting both organizations needs to be scientifically sound, fit in with curriculum goals, and be informal. Even though the RCAS and the CSC work well together, a lack of programs and funding persists. Indeed, the enrollment as of September 2002 reveals that ten out of sixteen of the public elementary schools and one middle school in the district qualify for U.S. Government Title I Assistance, according to Rapid City Area School District Enrollment [2] and Title I Part A – Selection of the Eligible School Attendance Areas [3] reports. One school shows that 90.4% of the students come from low-income families and 59.9% of the student body is Native American. Clearly, the majority of schools qualify for government aid, especially in funding for materials and programs in mathematics and science.

This project directly addresses this need by integrating resources from the South Dakota School of Mines and Technology, the Museum of Geology, the Children's Science Center, the Rapid City Area School District and the Badlands National Park whose educational staff provided additional curricula suggestions [4] through its "Badlands in Your Classroom" (NPS, 2002). The software resulting from this project creates an interactive educational experience and provides an impetus to learn. Since the software also contains web-based components, it can serve as a web portal to link to additional resources, for example, at the Museum of Geology located on the SDSMT campus or the website sponsored by the Badlands National Park. Additionally, educators can access suggested lesson plans that are age and topic appropriate and further points of contact to help fill gaps in current teaching materials. Currently, plans include software installation on two stand-alone workstations. Each station thus becomes a locus of information and resources about pre-historic animals as well as the Children's Science Center. The project currently has two entertaining, informative, interactive programs to educate children and adults about pre-historic animals, especially from the Oligocene period. However, this paper deals primarily with the puzzle games and briefly describes the other game. As a draw for visitors, the website also show-cases these puzzle games offered on three distinct experience levels, covering a pre-kindergarten to adult interest span.

To ensure scientific accuracy, Dr. Gale Bishop, Director of the Museum of Geology and paleontology graduate students have agreed to verify data accuracy.

This project breaks down into four phases: project definition, software design and analysis, implementation and testing, and installation at the CSC. This paper briefly handles the project definition phase. Focus then shifts to implementation by providing an overview of the software components and their basic mechanics.

Project Scope and Definition

Early in the definition phase, the team determined the financial, educational and time constraints for the project. The financial aspects of the project remained the straight forward. Team members requested funding from several national and local organizations. Rushmore State Bank, now Great Western Bank, donated a sum of money. Raytheon provided two Pentium III computer workstations. Since South Dakota prides itself on having one of the most wired school systems in the nation, team members wanted to design web components for this project to ensure a larger outreach. One of the possible ways to achieve both windows and web-based software was to code the software using the Microsoft .NET platform. After submitting a proposal, the team received approval from Microsoft. In addition to donating software and materials, Microsoft representatives provided assistance with the transition to .NET coding.

Based on an assessment of the science curriculum provided by the RCAS and interviewing CSC personnel, team members elected to aim their software at the early learner. Although this software specifically targets the need for pre-kindergarten through early reader material, it expands to interest older students and has provision for further augmentation. Team members chose paleontology not only because it is currently a hot topic in media, but also because the area offers substantial paleontological resources. In addition to the Museum of Geology at SDSMT, the Badlands National Park hosts one of the world’s largest Oligocene period fossil sites. Since games can provide a learning experience in a relaxed and conducive atmosphere, the team devised two games to provide these interactive experiences. One game was based on the puzzle concept and the other on a more complex game, similar to those played on boards. Because puzzles are less complex and appeal to most younger children, the team chose to implement puzzle pictures of pre-historic animals as venue for the early learner. Additionally, puzzles cover a wide variety of topics. Therefore, the thematic is adaptable from geology or paleontology to another science. One goal of the project was to design components with flexibility to accommodate the addition of other games or alterations in topics.

Furthermore, one aspect of the project serves as source of professional development for instructors. Within the resources, there are curriculum suggestions and lists of additional information sources. Of course, other users can access this data. Eventually, this repository could become a database.

Another key item is the provision for feedback to the CSC by community members through e-mail contact. The web-based nature of the project lends allows users not only the opportunity to kept abreast of the programs and special events at the CSC, but also provides an avenue to make request or express concerns.

Because two of the team members are graduate students at SDSMT, this project also meets the requirements for the Non-thesis Option for Masters of Science degree.

Therefore, it is supervised academically and under specific time constraints that coincide with CSC season schedule.

Component Overview

The software both on the workstation at the CSC and the web pages have similar major components.

Opening Interface Window

The student designed software presents the user with a full screen, rectangular opening window consisting of central oval shaped logo unique to the software and divided into three access button areas. Each access button area will have the letter signifying the area into which the user will branch. ‘I’ in the upper left hand corner will represent the portal into informational sub windows containing facts about the CSC including any on-going special events, SDSMT, The Museum of Geology, and the Visitor’s Center at the Badlands National Park. A ‘G’ in the upper right hand corner will lead the user to two or more educational games. Each game subsequently open to windows describing the respective game and the player levels for that game. The ‘R’ in the lower right hand corner will give educators and other interested users access to supplemental curricular suggestions, links to additional information, and mediotech resources as they become available. Each section will have child-friendly, intuitive icons. Across the top of the screen the user will see an inviting message identifying the Children’s Science Center. Sponsor information will be aesthetically placed along the bottom. Since CSC already has a website, visitors to the website will access this page from a link on the CSC home page, currently located at http://www.hpcnet.org/sdsmt/csc.

Information Component

The Information component gives more details about the CSC including: schedules, Enrichment Program highlights and future events, guest speakers, etc. In addition, the user can access concise information about the Museum of Geology and its programs. Finally, the information portal mentions information about the Badlands National Park and NPS Visitor Programs.

Resource Component

In order to fill a demand by educators for more quality material, which they can incorporate into classroom or home instruction, the resource portal provides a list of additional literature, texts, audio-visual matter, and places to contact pertaining to paleontology and geology. The resource component also includes suggested lessons plans based on grade and subject matter.

Game Component (Interactive Learning Opportunities)

This portal leads to a window with a list and a brief of educational games available to the user. For a brief description of a particular game, the user can click once on the name of the game. Currently, the project features two games. Since the games appear in list form, team members could devise a different game and the name of this game could easily fit into the list. One game, a puzzle game for ages 4 – 8, features a frame into which the child places pieces of a picture of a pre-historic animal. The user will be visually rewarded for the successful completion of the animal and hear a short narration about that animal, if available. Since this program is designed for children with little or no reading skills, audio instructions and hints will be given to the child. The second game features a scenario similar to the popular survival board games in which the user aids a pre-historic infant or juvenile animal from a user-selected epoch in becoming an adult.

Puzzle Game Premise

The main object of this interactive learning experience is to teach children who have little or no reading skills about pre-historic animals and pique their curiosity about geology and paleontology. After selecting the puzzle game component, the child chooses the level at which he wishes to play. From a series of pictures of pre-historic animals from various epochs, the child chooses one puzzle to put to together. Upon completion of the puzzle picture of the animal, the program congratulates the child. The program also provides information about the animal, e.g., life style of the animal, habitat, diet, size, modern relatives, etc. If scientists have determined possible locomotion for that animal and the team has been given permission to use a particular source, the child can watch the pre-historic animal move.

Puzzle Game Mechanics

The learning experience begins with an audio-visual introduction informing the child that he or she is going to have fun putting an animal together and learn how that animal lived. The game then tells the child that he can play the game as a beginner, intermediate or as an advanced player. A switchboard appears with three options: Beginner, Intermediate, and Advanced. The child then receives the option to play as a beginner if no option is selected. The game assumes that child knows how to operate a mouse or that there is a caregiver with him to show him. Since putting together puzzles is a common skill, the program omits the explanation of how puzzles work. If the child chooses the intermediate or advanced option, the program divides the picture into sixteen or twenty-five pieces. In the advanced version, those pieces lie in random orientation and the player must move them into proper orientation before he drags and drops them into place. To select an animal, the child can scroll through the pictures by using forward and back arrows, located immediately below the frame.

The opening game screen features a rectangular frame into which the user places the puzzle pieces of the pre-historic animal. After choosing an animal, the picture breaks into the appropriate number of pieces based the experience level. The pieces of each puzzle are randomized before the start of the game. In the beginner and intermediate version, the program displays the rectangle representing the playing area on the left and with the puzzle pieces appearing to the immediate right. The portion where the pieces lie takes up approximately one third of the screen. The remaining two thirds of the screen is devoted to the animal, which is presented as series of nine or sixteen appropriately shaped puzzle pieces in the beginning or intermediate versions. The advanced version has twenty-five pieces, which lie to the right and the left of picture frames. Figure 1: Game Layout at the Beginner Level, taken after children tested the game, shows the layout of the beginner's playing field [5].

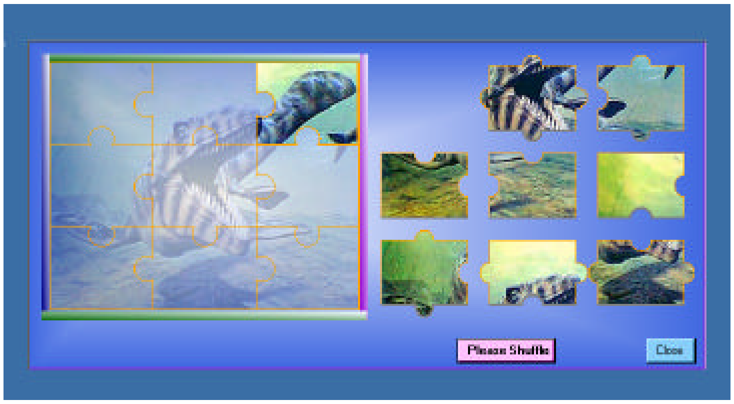

Figure 1: Game Layout at the Beginner Level

In the beginner version, a watermark of the picture, as seen in figure 1, serves as a guide to placing the nine pieces in the correct spot. Additionally, the program informs the child that if he needs a helpful hint, he can click on the help button. This button is a picture of a hand and the word 'help' under it, located in the upper right hand corner beyond the borders in figure 1. In the advanced version, the help version flashes the watermark since this level features an open frame.

The child uses a drop and drag technique to place the chosen element in its corresponding spot within the puzzle frame. As the element approaches the target area, it snaps into place. The child cannot put an element in an inappropriate place. Whenever the puzzle is completed, the child receives an acoustic reward; "Good Job." Additionally, an information block slides down over the area where the pieces were originally situated. Typically, the child is rewarded for completion of the puzzle and given brief information about the pre-historical animal. If a picture of the skeleton of the pre-historical animal is available, the program also displays that picture, usually in thumbnail format.

A literate child can choose an animal from a word database and a child who is unable to read can click through a series of pictures. After the child selects the picture element, the program bases the animal to be constructed on that element. At all levels, the program also tracks which parts have been selected as the game progresses.

The game concludes when all of the pieces of the chosen animal are in place. For beginners, the game then rewards the child with an audiovisual message. Following the reward message, the child hears about the animal. Specifically, the child learns: when the animal lived, the presumed geographic areas in which the animal thrived, about its dietary habits, its life span, which animals were it predators –'enemies', which animals were its co-habitors – 'friends'. The detail of the information that the program provides depends on the information that is currently known about the animal and its availability. Furthermore, if there is an accepted theory concerning the locomotion of the animal, the animal moves within the rectangle representing approximately two thirds of the screen. Additionally, if an animated clip is available, the program will play the clip while the child is hearing about the animal. After the game tells the child about the animal, the program allows the child to choose to play again.

Premise of the Paleoland Game

The main object of this interactive learning experience, called Paleoland, is to teach children who have reading skills about pre-historic animals and peek their curiosity about geology and paleontology. The second game features a scenario similar to board games in which the user must aid a pre-historic infant or juvenile animal from a user-selected period along a path – "Path of Life" to reach maturity. The goal is to help the baby reach adulthood by achieving a certain number of points as he moves through the different climate areas until he reaches the proper habitat. When two players participate, the winner will be the first baby to reach maturity. Each time the child plays the game; the child will encounter a slightly different board background based on the random arrangement of one to six climate zones and the overlay of one of ten stepping stone paths. Even if the child selects the same epoch twice and chooses the same infant animal, the playing field will be different. Upon reaching the end of the game, the program congratulates the child and displays information about the environment and the life style of the animal. If scientists have determined possible locomotion for that animal, the child can watch it move. This game is designed as a windows application and currently there are no plans for web deployment.

Since this paper concentrates on the implementation of the puzzle game, the mechanics and implementation of the board game are left to another paper

Implementation

From the onset of the project, team members wanted to choose an efficient method of coding that allows windows applications to be able to be launched on the web without adding extensive code.

Selection of Programming Languages

Microsoft .NET accommodates both windows and web applications. Specifically, team members evaluated the latest version Microsoft's integrated development environment for suitability for this project. There are two programming languages that incorporate the use of windows forms as part of the .NET Framework, namely C++ and C#. According to Charles Petzold in *Programming Windows* (Microsoft Press, 2001), C# and the .NET Framework are “made for each other”. Therefore, C# seemed to be the most logical choice, especially in light of funding issues. Understanding the mechanics of C# and the .NET Framework is essential to the project.

C# is a new, objected oriented programming language belonging to the class of C languages. It is similar to C++ and C in appearance and structure, but uses forms like Visual Basic. C# even has a feature that allows the programmer to write comments with XML tags. The key to the .NET Framework appears to be its intermediate language, Microsoft Intermediate Language (MSIL). Previous versions of C and C++ compile directly to machine language. By contrast, C# compiles to MSIL by targeting a common language runtime (CLR). According Petzold [6], this makes C# executables platform independent. Although Petzold states that programmers can code directly to MSIL, it is the CLR that manages the code and provides the execution environment. In fact, code that does not target the CLR is called *unmanaged* code. The CLR assists with such elements as code verification, compilation, memory management aided by a garbage collection feature, and security management. In *Building Web Applications with C3 and .NET: A Complete Reference* (CRC Press, 2003), Dudley Gill adds that the “… CLR defines general characteristics such as visibility, naming (including casing), keywords and signatures, types, …, and type members, methods, properties and events.” (p. 4.) Basically, the CLR compiles code once and it runs on any platform supporting CLR (p. 5). The MSIL, which is CPU-independent, actually creates metadata. Metadata encompasses information about the types, members and references and describes the code. Both Petzold and Gill say this eliminates Type Libraries and Interface Definition Language. Next, the Just-In-Time (JIT) compiler must convert the code to machine or native language, which of course specific to the CPU. Adapted from Gill [7],
Figure 2: .NET Compilation Sequence neatly illustrates this process.

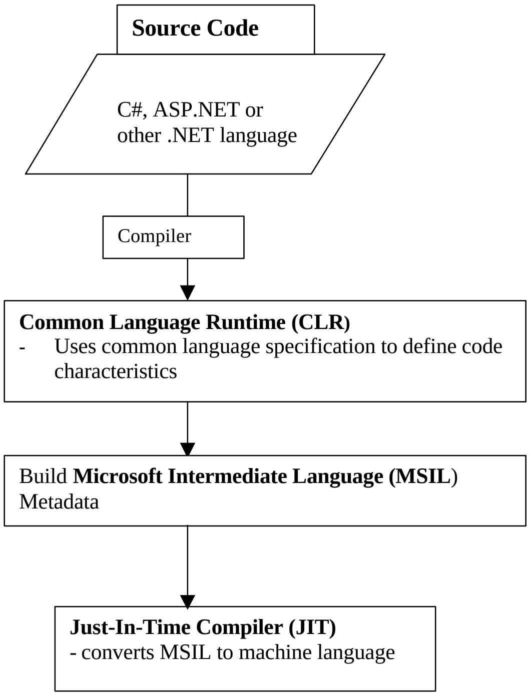
Source Code
C#, ASP.NET or other .NET language
Compiler
Common Language Runtime (CLR)
- Uses common language specification to define code characteristics
Build Microsoft Intermediate Language (MSIL) Metadata
Just-In-Time Compiler (JIT)
- converts MSIL to machine language

Figure 2: .NET Compilation Sequence

The main disadvantage to the .NET Framework is that this .NET runtime needs to be installed on other computers in order to run the C# program.

Nevertheless, ASP.NET operates on these same principles and has new features such as webservices, which make converting a windows based application to a web based application to a theoretically simple process. Additionally, ASP.NET has a number features which support OLE DB. Thus, ASP.NET works with any source that uses an OLE DB, including Microsoft Access and Microsoft SQL Server. To make handling data easier, ASP.NET also employs SQL and OLE data adapters. Based on this knowledge team members elected to implement the project using the .NET Framework with the majority of the programming being done in C#. To migrate the application to the web, programmers chose ASP.NET and HTML.

Coding Problems and Resolutions

Since C# is a relatively new programming language, team members familiarized themselves with C# and .NET coding techniques. In addition to Microsoft texts and documentation, team members took advantage of websites that allow users to post questions that are answered by other users and professional programmers.

Puzzle Piece Construction

Through trial and error, team members determined that for their programming environment .bmp, .jpg, .gif, and .png files provide the sharpest background pictures for the puzzle. The first major hurdle was creating the individual puzzle pieces. After consulting several websites [8] [9], the team decided to take this basic approach:

1) Find an image of the desired animal that has sufficient contrast to become a watermark
2) Cut a copy of the image into puzzle pieces by using path commands.
3) Highlight the jigsaw borders so the application generates guidelines for the beginning and advanced users and so that the pieces themselves are easier to see.

The team discovered that in order for the cuts to resemble puzzle pieces and fit together perfectly, the size of the image has to be divisible by the number of the pieces desired on a given side. For example, the beginner puzzle needs nine pieces, three on each side. Therefore, the size of the image had to be divisible by three. Square images were the easiest to manipulate and each of levels uses images that are almost perfect squares. This also complimented the rectangular shape of the forms that C# generates. The form had room for the square shape of the playing area as well as the corresponding area for the pieces, as figure 1 shows.

Additionally, since C# uses forms, a form representing the individual piece had to be coded. The actual code involved repetitive computations, which need to be painstakingly adjusted to ensure the proper appearance and fit. Team members made cuts into the piece by redrawing only portions of a circle. The effect was similar to punching partial holes in a scrap of paper. Calculating arc and cords lengths determined how much or how little of the circle appeared. The program repeated this process for all sides of the piece using myGraphicsPath method. Edge pieces, of course, kept (redrew) the appropriate one or two straight lines. Figure 3: Cutting a Path for Each Piece illustrates the complex and repetitive nature of C# code to cut a puzzle piece.

```
// Add left side of puzzle
myGraphicsPath.AddLine(X_Start,Y_Start,X_Start,Y_WaterPiece+Y_Start);

// Add bottom side of puzzle
myGraphicsPath.AddLine(X_Start,Y_WaterPiece+Y_Start,X_Start+5,Y_WaterPiece+Y_Start);
myGraphicsPath.AddArc ((X_WaterPiece-ARC)/2+X_Start,Y_WaterPiece+Y_Start-
t,ARC,ARC,90+d,240);

myGraphicsPath.AddLine(X_WaterPiece+X_Start - 5, Y_WaterPiece+Y_Start,
                       X_WaterPiece+X_Start,Y_WaterPiece+Y_Start);

// Add right side of puzzle
myGraphicsPath.AddLine(X_WaterPiece+X_Start,Y_WaterPiece+Y_Start,X_WaterPiece+X_Star
t,Y_WaterPiece+Y_Start-5);

myGraphicsPath.AddArc ( X_WaterPiece+X_Start-t,(Y_WaterPiece-ARC)/2
+Y_Start,ARC,ARC,d,180+d);
 myGraphicsPath.AddLine( X_WaterPiece+X_Start, Y_Start,  X_WaterPiece+X_Start, Y_Start);

//top
myGraphicsPath.AddLine(X_WaterPiece+X_Start,Y_Start,X_Start,Y_Start);
```

Figure 3: Cutting a Path for Each Piece

In order to use code efficiently, programmers created a size-based muster for each level. Therefore, the size of the image was slightly modified. The cutting technique depended on the size and did not depend on the content. This allowed the selection of the images to be independent of the puzzler’s level. In other words, the same image can be used on different levels. However, for the sake of diversity and to make the game more challenging, the application defaults to different images at each level for the duration of the user’s puzzling session.

Rotation of the Pieces

Rotating the pieces proved to be another large challenge. To make the game more challenging, team members felt that the intermediate and advanced level players should view the pieces in random orientation to the puzzle image. Not only do these pieces appear in a different order each time the visitor plays the game, but also their placement is upside down, sideways or parallel to the image. The player then double clicks on the piece to rotate it clockwise by 90° for each double click set.

Rotating solid shapes was straightforward. However, applying this process to the puzzle pieces not only took a long time for the processor work through the code, but it caused even the simplest image on the piece to ghost. Team members implemented the rotation process by using vector arrays. In all puzzle piece calculations; determining the coordinates of the corners of the points of intersection was the most important. Team members used these coordinates to check for correct positioning of pieces, to begin and end color gradation, and above all the as point of rotation. The most crucial of these coordinates was the point of intersection for four pieces – the four-corner coordinate.

Actual vector manipulation occurred in similar manner to the pseudo-code found in figure 4.

```
private void myPicBox2_DoubleClick(object sender, System.EventArgs e)

int iNum;
for(int x = 0; x < 4; x++)
   for (int y = 0; y < 4; y++)
      if (sender.Equals(this.PicBox2[x,y]))
              {
                       iNum = 4*x + y;
                       iRotation[iNum] = (iRotation[iNum] + 1)%4;

                      this.PicBox2[x,y].Image.RotateFlip(RotateFlipType.Rotate90FlipNone);
 //rotate 90 degree each double click

                      this.PicBox2[x,y].Refresh();
                   x = 4;
                  y = 4;

private void myPicBox2_Paint(object sender, System.Windows.Forms.PaintEventArgs e)
int iNum, iRota;

for(int x = 0; x < 4; x++)
    for (int y = 0; y < 4; y++)
              if (sender.Equals(this.PicBox2[x,y]))
                            {
                                          iNum = 4*x + y;

              //get this picBox's rotation record,0,1,2,3 total
                                          iRota = iRotation[iNum];
              //Restrict drawing area to the region described above
                                          this.PicBox2[x,y].Region = new Region(myPicBox2Path[iNum,iRota]);

                                          Pen myPen = new Pen(Color.Orange,ORANGE_LINE);
                                          e.Graphics.DrawPath(myPen, myPicBox2Path[iNum,iRota]);
                                          y = 4;
                                          x = 4;
```

Figure 4: Functions Required to Rotate Pieces

Coding generally commenced in a straightforward manner. The team also resolved issues regarding the watermark of the image, color gradient of the form, randomizing the placement of the pieces on the board, and the use of a slide form bearing the congratulatory and informational messages.

Implementation Status

As of the submission date for this paper, team members continue add functionality and runtime efficiency to the code. After children test the improvements, the plans to make adjustments in the code and incorporate constructive suggestions. The process of migrating the window-based application to the web is underway. Members of the team also assist in the programming of the second component, which primarily is being carried out through the SDSMT .NET Users Group. The discussion of the implementation of that component remains a topic for another paper. Both projects rely on permission to use the artist rendered images of the pre-historical animals and to include published information on the animals. Team members still await responses from some artists, authors and publishers.

Conclusion

February 21, 2003 marked the first time the project was presented to third, fourth and fifth grade students as a hands-on presentation during Engineering Week at SDSMT. Although the software was designed for younger children, older children and the adults who accompanied them enjoyed the experience and asked for the website address. The enthusiasm with which visitors greeted the project indicates just how well the program fits into the CSC concept. Educators and parents were pleased to learn that skeletons of the pre-historic animals are available in the Rapid City Area. Some of the children have since visited the Museum of Geology and the Children's Science Center. At least in this first instance, the project met its goal of generating and perpetuating interest in the sciences.

Clearly, through creativity and persistent involvement individuals can help bridge the digital gap to bring interactive learning and technology into the community by integrating existing resources.

References

1. Smith Peel, P. (2001). Philosophy. *K-5 Science Curriculum* (2). Rapid City, SD.

2. Brenden, H. (2002). Rapid City Area School District 51-4: Enrollment Report. Retrieved January 4, 2003 from http://www.rcas.org/csac/second/.

3. Warton, P & Rose, R. (2002). Title I Part A – Selection of the Eligible School Attendance Areas (2-3). Rapid City, SD.

4. National Park Service. (2002). Badlands in Your Classroom. Part of *Badlands National Park: Earth Science Education Program* (CD-ROM format). NPS: U.S. Department of the Interior.

5. The painting of the Mosasaur used in this puzzle was done by Mark Hallett and is published on page 203 of *Dinosaurs - A Global View* by Sylvia J Czerkas and Stephen A. Czerkas. (Barnes and Nobel Books, 1990). Retrieved January 4, 2003 from http://www.oceansofkansas.com/images2.

6. Petzold, C. (2001). *Programming Windows*. Redmond, WA: Microsoft Press.

7. Gill, D. (2003). *Building Web Applications with C3 and .NET: A Complete Reference.* Boca Raton, FL: CRC Press.

8. Sanju. Events in C#. Retrieved January 12, 2003 from http://www.csharphelp.com/archives/archive253.html.

9. Thangaraju, S. Image Viewer in C#. Retrieved February 21, 2003 from http://www.c-sharpcorner.com/winforms/ImageViewerST.asp.

Acknowledgements

Team members gratefully acknowledge the assistance of Bob Rose, Director of Federal Programs/Grants at RCAS and Charles Beall, Education Specialist at Badlands National Park for providing curricula materials and suggestions. We also would like to thank Dr. Gale Bishop for his support by allowing us to use digital information and for his willingness to verify data accuracy.

Additionally, appreciate the efforts of representatives of Great Western Bank, Raytheon, and Microsoft for the donation of funding and materials.