INFORMATION ABOUT PRINCIPAL INVESTIGATORS/PROJECT DIRECTORS(PI/PD) and co-PRINCIPAL INVESTIGATORS/co-PROJECT DIRECTORS

Submit only ONE copy of this form **for each PI/PD and co-PI/PD** identified on the proposal. The form(s) should be attached to the original proposal as specified in GPG Section II.B. ***DO NOT INCLUDE THIS FORM WITH ANY OF THE OTHER COPIES OF YOUR PROPOSAL AS THIS MAY COMPRISE THE CONFIDENTIALITY OF THE INFORMATION.***

PI/PD Name: Brian M Slator

Gender: ☒ Male ☐ Female

Ethnicity: (Choose one response) ☐ Hispanic or Latino ☒ Not Hispanic or Latino

Race:
(Select one or more)
- ☐ American Indian or Alaska Native
- ☐ Asian
- ☐ Black or African American
- ☐ Native Hawaiian or Other Pacific Islander
- ☒ White

Disability Status:
(Select one or more)
- ☐ Hearing Impairment
- ☐ Visual Impairment
- ☐ Mobility/Orthopedic Impairment
- ☐ Other ________________
- ☒ None

Citizenship: (Choose one) ☐ U.S. Citizen ☒ Permanent Resident ☐ Other non-U.S. Citizen

Check here if you do not wish to provide any or all of the above information (excluding PI/PD name): ☒

REQUIRED: Check here if you are currently serving (or have previously served) as a PI, co-PI or PD on any federally funded project ☒

Ethnicity Definition:
Hispanic or Latino. A person of Mexican, Puerto Rican, Cuban, South or Central American, or other Spanish culture or origin, regardless of race.
Race Definitions:
American Indian or Alaska Native. A person having origins in any of the original peoples of North and South America (including Central America), and who maintains tribal affiliation or community attachment.
Asian. A person having origins in any of the original peoples of the Far East, Southeast Asia, or the Indian subcontinent including, for example, Cambodia, China, India, Japan, Korea, Malaysia, Pakistan, the Philippine Islands, Thailand, and Vietnam.
Black or African American. A person having origins in any of the black racial groups of Africa.
Native Hawaiian or Other Pacific Islander. A person having origins in any of the original peoples of Hawaii, Guam, Samoa, or other Pacific Islands.
White. A person having origins in any of the original peoples of Europe, the Middle East, or North Africa.

WHY THIS INFORMATION IS BEING REQUESTED:

The Federal Government has a continuing commitment to monitor the operation of its review and award processes to identify and address any inequities based on gender, race, ethnicity, or disability of its proposed PIs/PDs. To gather information needed for this important tasks, the proposer should submit a single copy of this form for each identified PI/PD with each proposal. Submission of the requested information is voluntary and is not a precondition of award. However, information not submitted will seriously undermine the statistical validity, and therefore the usefulness, of information recieved from others. Any individual not wishing to submit some or all the information should check the box provided for this purpose. (The exceptions are the PI/PD name and the information about prior Federal support, the last question above.)

Collection of this information is authorized by the NSF Act of 1950, as amended, 42 U.S.C. 1861, et seq. Demographic data allows NSF to gauge whether our programs and other opportunities in science and technology are fairly reaching and benefiting everyone regardless of demographic category; to ensure that those in under-represendted groups have the same knowledge of and access to programs and other research and educational oppurtunities; and to assess involvement of international investigators in work supported by NSF. The information may be disclosed to government contractors, experts, volunteers and researchers to complete assigned work; and to other government agencies in order to coordinate and assess programs. The information may be added to the Reviewer file and used to select potential candidates to serve as peer reviewers or advisory committee members. See Systems of Records, NSF-50, "Principal Investigator/Proposal File and Associated Records", 63 Federal Register 267 (January 5, 1998), and NSF-51, "Reviewer/Proposal File and Associated Records", 63 Federal Register 268 (January 5, 1998).

COVER SHEET FOR PROPOSAL TO THE NATIONAL SCIENCE FOUNDATION

PROGRAM ANNOUNCEMENT/SOLICITATION NO./CLOSING DATE/if not in response to a program announcement/solicitation enter NSF 00-2	FOR NSF USE ONLY
NSF 99-167 **02/16/00**	**NSF PROPOSAL NUMBER**
FOR CONSIDERATION BY NSF ORGANIZATION UNIT(S) (Indicate the most specific unit known, i.e. program, division, etc.) **INFORMATION TECHNOLOGY RESEARC**	

DATE RECEIVED	NUMBER OF COPIES	DIVISION ASSIGNED	FUND CODE	DUNS# (Data Universal Numbering System)	FILE LOCATION
				803882299	

EMPLOYER IDENTIFICATION NUMBER (EIN) OR TAXPAYER IDENTIFICATION NUMBER (TIN)	SHOW PREVIOUS AWARD NO. IF THIS IS	IS THIS PROPOSAL BEING SUBMITTED TO ANOTHER FEDERAL AGENCY? YES ☐ NO ☒ IF YES, LIST ACRONYMS(S)
456002439	☐ A RENEWAL ☐ AN ACCOMPLISHMENT-BASED RENEWAL	

NAME OF ORGANIZATION TO WHICH AWARD SHOULD BE MADE	ADDRESS OF AWARDEE ORGANIZATION, INCLUDING 9 DIGIT ZIP CODE
North Dakota State University Fargo	**Post Office Box 5405** **University Station** **Fargo, ND. 581055405**
AWARDEE ORGANIZATION CODE (IF KNOWN) **0029975000**	
NAME OF PERFORMING ORGANIZATION, IF DIFFERENT FROM ABOVE	ADDRESS OF PERFORMING ORGANIZATION, IF DIFFERENT, INCLUDING 9 DIGIT ZIP CODE
PERFORMING ORGANIZATION CODE (IF KNOWN)	

IS AWARDEE ORGANIZATION (Check All That Apply)
(See GPG II.D.1 For Definitions) ☐ FOR-PROFIT ORGANIZATION ☐ SMALL BUSINESS ☐ MINORITY BUSINESS ☐ WOMAN-OWNED BUSINESS

TITLE OF PROPOSED PROJECT **ITR/SOC: Learning Computer Science in Virtual Worlds**

REQUESTED AMOUNT	PROPOSED DURATION (1-60 MONTHS)	REQUESTED STARTING DATE	SHOW RELATED PREPROPOSAL NO., IF APPLICABLE
$ **332,903**	**36** months	**08/01/00**	

CHECK APPROPRIATE BOX(ES) IF THIS PROPOSAL INCLUDES ANY OF THE ITEMS LISTED BELOW

- ☐ BEGINNING INVESTIGATOR (GPG 1.A.3)
- ☐ DISCLOSURE OF LOBBYING ACTIVITIES (GPG II.D.1)
- ☐ PROPRIETARY & PRIVILEGED INFORMATION (GPG II.D.10)
- ☐ NATIONAL ENVIRONMENTAL POLICY ACT (GPG II.D.10)
- ☐ HISTORIC PLACES (GPG II.D.10)
- ☐ SMALL GRANT FOR EXPLOR. RESEARCH (SGER) (GPG II.D.12)
- ☐ GROUP PROPOSAL (GPG II.D.12)
- ☐ VERTEBRATE ANIMALS (GPG II.D.12) IACUC App. Date ____
- ☐ HUMAN SUBJECTS (GPG II.D.12) Exemption Subsection ____ or IRB App. Date ____
- ☐ INTERNATIONAL COOPERATIVE ACTIVITIES: COUNTRY/COUNTRIES ____
- ☐ FACILITATION FOR SCIENTISTS/ENGINEERS WITH DISABILITIES (GPG V.G.)
- ☐ RESEARCH OPPORTUNITY AWARD (GPG V.H)

PI/PD DEPARTMENT	PI/PD POSTAL ADDRESS
Department of Computer Science	**IACC Building, Room #258** **Fargo, ND 58105** **United States**
PI/PD FAX NUMBER **701-231-8255**	

NAMES (TYPED)	High Degree	Yr of Degree	Telephone Number	Electronic Mail Address
PI/PD NAME **Brian M Slator**	**Ph.D.**	**1988**	**701-231-6124**	**slator@badlands.nodak.edu**
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				

CERTIFICATION PAGE

Certification for Principal Investigators and Co-Principal Investigators:

I certify to the best of my knowledge that:

(1) the statements herein (excluding scientific hypotheses and scientific opinions) are true and complete, and
(2) the text and graphics herein as well as any accompanying publications or other documents, unless otherwise indicated, are the original work of the signatories or individuals working under their supervision. I agree to accept responsibility for the scientific conduct of the project and to provide the required progress reports if an award is made as a result of this application.

I understand that the willful provision of false information or concealing a material fact in this proposal or any other communication submitted to NSF is a criminal offense (U.S.Code, Title 18, Section 1001).

Name (Typed)	Signature	Social Security No.*	Date
PI/PD **Brian M Slator**		SSNs are confidential and are not displayed *ON FASTLANE SUBMISSIONS*	
Co-PI/PD			
Co-PI/PD			
Co-PI/PD			
Co-PI/PD			

Certification for Authorized Organizational Representative or Individual Applicant:

By signing and submitting this proposal, the individual applicant or the authorized official of the applicant institution is: (1) certifying that statements made herein are true and complete to the best of his/her knowledge; and (2) agreeing to accept the obligation to comply with NSF award terms and conditions if an award is made as a result of this application. Further, the applicant is hereby providing certifications regarding Federal debt status, debarment and suspension, drug-free workplace, and lobbying activities (see below), as set forth in Grant Proposal Guide (GPG), NSF 00-2. Willful provision of false information in this application and its supporting documents or in reports required under an ensuring award is a criminal offense (U. S. Code, Title 18, Section 1001).

In addition, if the applicant institution employs more than fifty persons, the authorized official of the applicant institution is certifying that the institution has implemented a written and enforced conflict of interest policy that is consistent with the provisions of Grant Policy Manual Section 510; that to the best of his/her knowledge, all financial disclosures required by that conflict of interest policy have been made; and that all identified conflicts of interest will have been satisfactorily managed, reduced or eliminated prior to the institution's expenditure of any funds under the award, in accordance with the institution's conflict of interest policy. Conflict which cannot be satisfactorily managed, reduced or eliminated must be disclosed to NSF.

Debt and Debarment Certifications

(If answer "yes" to either, please provide explanation.)

Is the organization delinquent on any Federal debt?	Yes ☐	No ☒
Is the organization or its principals presently debarred, suspended, proposed for debarment, declared ineligible, or voluntarily excluded from covered transactions by any Federal department or agency?	Yes ☐	No ☒

Certification Regarding Lobbying

This certification is required for an award of a Federal contract, grant, or cooperative agreement exceeding $100,000 and for an award of a Federal loan or a commitment providing for the United States to insure or guarantee a loan exceeding $150,000.

Certification for Contracts, Grants, Loans and Cooperative Agreements

The undersigned certifies, to the best of his or her knowledge and belief, that:

(1) No federal appropriated funds have been paid or will be paid, by or on behalf of the undersigned, to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with the awarding of any federal contract, the making of any Federal grant, the making of any Federal loan, the entering into of any cooperative agreement, and the extension, continuation, renewal, amendment, or modification of any Federal contract, grant, loan, or cooperative agreement.

(2) If any funds other than Federal appropriated funds have been paid or will be paid to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with this Federal contract, grant, loan, or cooperative agreement, the undersigned shall complete and submit Standard Form-LLL, ''Disclosure Form to Report Lobbying,'' in accordance with its instructions.

(3) The undersigned shall require that the language of this certification be included in the award documents for all subawards at all tiers including subcontracts, subgrants, and contracts under grants, loans, and cooperative agreements and that all subrecipients shall certify and disclose accordingly.

This certification is a material representation of fact upon which reliance was placed when this transaction was made or entered into. Submission of this certification is a prerequisite for making or entering into this transaction imposed by section 1352, title 31, U.S. Code. Any person who fails to file the required certification shall be subject to a civil penalty of not less than $10,000 and not more than $100,000 for each such failure.

AUTHORIZED ORGANIZATIONAL REPRESENTATIVE	SIGNATURE	DATE
NAME/TITLE (TYPED) **Dr. Edna Holm, Interim Dean of Research**		**02/10/00**

TELEPHONE NUMBER	ELECTRONIC MAIL ADDRESS	FAX NUMBER
701-231-7131	**ndsu-research@plains.nodak.edu**	**701-231-1013**

*SUBMISSION OF SOCIAL SECURITY NUMBERS IS VOLUNTARY AND WILL NOT AFFECT THE ORGANIZATION'S ELIGIBILITY FOR AN AWARD. HOWEVER, THEY ARE AN INTEGRAL PART OF THE INFORMATION SYSTEM AND ASSIST IN PROCESSING THE PROPOSAL. SSN SOLICITED UNDER NSF ACT OF 1950, AS AMENDED.

PROJECT SUMMARY

This project seeks to study Computer Science education and learning in the context of an immersive virtual world. In particular, we propose to develop and test a multi-user virtual domain built on the model of a virtual Exploratorium of Computing. The structure of the virtual world will conform to the pedagogical hyper-structure of a computing curriculum. However, the material will be orgnized in an organic fashion, held together by a role-based theory of immersive education. Hence, the normal course boundaries and sequences will be abandoned in favor of a pragmatic, goal-oriented structure.

We propose to develop a two-level course combining the study of programming languages with the craft of writing programs. This will entail the implementation of an interactive module for assessing student assignments, and a graphical user interface for algorithmic visualizations.

We plan to study student learning in this virtual context. A formal comparison of this approach will be conducted, contrasting classroom instruction with our anyplace/anywhere, self-paced, and software tutored paradigm. Once completed, we hope to "publish" this solution for the benefit of a wide educational audience.

This project is a collaboration between two institutions in an EPSCoR state, North Dakota. This is a high-risk project, but with a high potential pay-off in terms of influencing the way Computer Science is taught.

TABLE OF CONTENTS

For font size and page formatting specifications, see GPG section II.C.

*Proposers may select any numbering mechanism for the proposal, however, the entire proposal must be paginated. Complete both columns only if the proposal is numbered consecutively.

1. Project Description

The ProgrammingLand MOOseum (Hill and Slator, 1998; Slator and Hill, 1999), implements an Exploratorium-style museum metaphor to create a hyper-course in computer programming principles aimed at structuring the curriculum as a tour through a virtual museum. Student visitors participate in a self-paced exploration of the exhibit space where they are introduced to the concepts of computer programming, are given demonstrations of these concepts in action, and are encouraged to manipulate the interactive exhibits as a way of experiencing the principles being taught, and to construct their own understanding of them (Duffy, Lowyck, Jonassen, 1983; Duffy and Jonassen, 1992).

ProgrammingLand has been developed and used in conjunction with traditional classroom instruction. However, the goal is for distance learning and non-traditional classes. It is intended to deliver the content that would normally be obtained from a lecture or textbook, yet also have many of the attractive qualities of games and other learner centered activities.

This proposal describes a project that has been ongoing for three years, involving Computer Science faculty at North Dakota State University (NDSU) and Valley City State University (VCSU; a four year college). Both institutions are located in North Dakota, an EPSCoR state, and Valley City is notable for being of America's first "laptop colleges". In particular, we describe a system being used in two ways in our curriculum. First, as a source of projects in the NDSU Comparative Languages course, where junior-level students study programming languages and implement virtual artifacts as a way of gaining a deeper understanding of the principles being covered. And second, as a virtual space for introductory programming courses (CS1 and CS2) at VCSU, where freshman-level students are engaged in the process of exploration and discovery as they learn how to program in C++.

The current collaboration has been successful, as we describe below, and is now ready for the next level. To do this, we propose a 3-year project composed of the following elements:

- Extending and finalizing the C++ wing of the MOOseum in terms of topics and subject matter, with particular attention paid to hyper-linking the space to reinforce the common principles shared across programming languages;
- Imposing a site-map (conceptual index) on the virtual space, to facilitate searching;
- Implementing a system for assessing student code, modeled on the Ceilidh system (Foubister, Michaelson, and Tomes, 1997);
- Further developing content areas related to algorithms in support of the Ceilidh module;
- Implementing a graphical user interface for algorithmic visualization of program compilation and execution;
- Controlled, comparative studies for assessing student learning in virtual worlds;
- Controlled evaluation of course content.

2. Visiting ProgrammingLand

A typical session for a student starts with a execution of the client program, that connects to the MOOseum. ProgrammingLand itself is structured into rooms, with exits being the path from one room to another. The motif of ProgrammingLand is that of an Exploratorium style museum. Therefore the term exhibit is usually used instead of room. Whenever an exhibit or room is entered, its description is displayed. Typically this is a paragraph or two of instruction on some topic. The exhibit also indicates what exits exist and where they lead. In the spirit of user-centered control, the student is always free to choose which path to take and what to do next. Although an exit is a one way path from one exhibit to another, they often come in pairs so that a student may return conveniently (the exception being the quiz rooms, described below).

The ProgrammingLand MOOseum is both a virtual environment and a web server. Anyone can visit and traverse the geography by pointing their browser to http:// newton.vcsu.nodak.edu: 7000/. This method of visiting the MOOseum is not fully interactive, which requires a login, but nonetheless provides a convenient way for touring the environment.

3. Local Context

The NDSU World Wide Web Instructional Committee (WWWIC; Slator et al, 1999) is currently engaged in several virtual/visual educational projects: The Geology Explorer (Slator et al., 1998; Saini-Eidukat et al. 1999; Schwert et al., 1999), The Virtual Cell (White, et al, 1999), and The Visual Computer Program (Juell, 1999). These have shared and individual goals. Shared goals include the mission to teach Science structure and process: the Scientific Method, scientific problem solving, diagnosis, hypothesis formation and testing, and experimental design. The individual goals are to teach the content of individual scientific disciplines: Geology, Cell Biology, and Computer Science.

These projects are designed to capitalize on the affordances provided by virtual environments. For example, to

- control virtual time and collapse virtual distance,
- create shared spaces that are physical or practical impossibilities,
- promote shared experiences for participants in different physical locations,
- implement shared agents and artifacts according to specific pedagogical goals,
- support multi-user collaborations and competitive play.

More information is available at http://www.ndsu.nodak.edu/wwwic/.

4. Fundamental Objects

ProgrammingLand is hosted on a MOO ("MUD, Object-Oriented", where MUD stands for "Multi-User Domain"). MUDs are typically text-based electronic meeting places where players build societies and fantasy environments, and interact with each other (Curtis 1992). Technically, a MUD is a multi-user database and messaging system. The basic components are "rooms" with "exits", "containers" and "players". MUDs support the object management and inter-player messaging required for multi-player games, and at the same time provide a programming language for writing the simulation and customizing the MUD.

In a MOO every thing is an object. These objects have properties that carry information, and methods that perform useful functions. When a student logs into the MOOseum, their character is activated. This character is the object the server uses to manage everything known about the player. There are several important properties attached to the player that have an impact on the educational use of the MOO. First is a list of every room they have visited. Second is a list of event and award tokens the student has earned. Third is a goal. Every student has a goal at all times. Typically an award token denotes the accomplishment of a goal, at which point a new goal is instantly assigned.

There are several types of room or exhibit in the MOOseum. The first type is called the lecture room, since it delivers a paragraph or so of expository text. A special type of lecture room is called a signpost or menu room, which serves as sort of a crossroad in the MOOseum. Lecture rooms also contain a property that holds quiz questions, discussed below. A second type is called a lesson room, since it controls access to a group of exhibits that comprise a lesson. The lesson room is a specialization of a lecture room and is often a signpost or menu room as well. The third type is called a workroom, which is used to house some type of interactive object. Currently there are two such objects in frequent use: the code machine and the workbench. A fourth type is called a quiz room, where students take tests. These are always connected to a lesson room.

5. Interactive Objects

The ProgrammingLand MOOseum is composed of "wings" dedicated to different programming languages: including C++, Lisp, Basic, and Java. Most wings have several interactive objects, usually code machines and workbenches. A code machine gives an explanation and trace of a piece of example code. A workbench allows a student to construct a small piece of code and then parses it. However, since it is problematic to install a full compiler into the MOO, it is assumed the student has access to a compiler or interpreter for the language they are studying (this proposal aims to solve that problem). Therefore assignments to be done outside of the MOO are currently required. This is accomplished by an interaction between several objects including a "lesson dispatcher" and a "roving goalie" agent.

```
A short program with assignments
This code machine is named simple, with an alias of s. It demonstrates a short program with assignments
and outputs. Use
   help simple
to get help on using code machines.
It would be a good exercise to look at the code in simple and try to compute manually what values will be
left in the variables a, b and c. Then use the trace feature to determine how close you were.
You see simple here.
Obvious exits: [exit] to Practice with the assignment statement
```

Figure 1: An exhibit with a code machine

5.1 Code Machines

The code machine contains a piece of programming code, which it will display, explain or trace (see Figures 1-4). The code machine may display the code with or without line numbers. The line numbers are important for the explanation and trace, but suppressing line numbers allows the student to copy the code from the MOO client and paste it into an edit window and actually compile the code.

```
=>show simple
 1: #include <iostream.h>
 2: #include <iomanip.h>
 3: int main () {
 4:  int a = 3, b = 5,c = -7;
 5:  a = b+c;
 6:  b = c + b * a;
```

Figure 2: Displaying Contents of a Code Machine

The explanation of the code works on a line by line basis. The numbered line is displayed with the explanation of just that line. The student then requests the next line. The trace of the code is a simulated execution.

```
=>explain simple
1: #include <iostream.h>
The first include obtains access to the I/O stream objects of cin and cout as well as the put-to and get-from
operators.
…
=>next simple
4:  int a = 3, b = 5,c = -7;
Declare and initialize the three integer variables. The initialization uses literals, rather than computed
expressions.
=>n s
5:  a = b+c;
Store the sum of b and c into a. Whatever value a had before is now lost. Thus the initialization of a was
not needed.
```

Figure 3: Receiving the Explanation of a Code Machine

The code machine displays the lines that are executed along with a description of what is happening at run time, including the new contents of any variables that are changed. When a student completes either the explanation or trace of a code machine, an "event" token is recorded, giving the student credit for their actions.

```
=>trace simple
 4:  int a = 3, b = 5,c = -7;
The program begins with the initialization of the three variables.
 Variable a receives 3
 Variable b receives 5
 Variable c receives -7
=>n s
 5:  a = b+c;
The first assignment computes the value of b + c, which results in -2, which is assigned to a.
 Variable a receives -2
=>n s
 6:  b = c + b * a;
The precedence of multiplication is higher so the multiply is done first and yields -10. This is then added to
c to give -17.
 Variable b receives -17
```

Figure 4: Tracing the Execution of a Code Machine

5.2 Workbenches

A workbench is built around a table driven parser, with enough of the tokens of a program fragment to allow the student to test whether they have the syntax of a construct correct. The student builds a statement or program fragment from the pieces that are inserted into the workbench. When complete, the student can ask the workbench to determine if the fragment is syntactically correct or not. In very simple instances the workbench may also interpret the code and run the program fragment. Like the code machine a successful parse of the code records an event token for the student.

6. The Structure of ProgrammingLand

It will be helpful to consider the structure of the MOO from two different perspectives: the logical structure and the pedagogical structure. The logical structure is shared with every other MOO and the pedagogical structure is superimposed on this logical structure and distinguishes ProgrammingLand from other MOOs.

6.1 The Logical Structure

The logical structure of any MOO is that of a directed graph with nodes, usually called rooms in a MOO, and arcs, which are called exits. A room or exhibit is just a MOO object that may contain players or other objects and various properties. Each MOO object, including rooms, has a name and a description, both of which are displayed to a player when they enter the room. Most of the subject content of ProgrammingLand is in the descriptions of the exhibits.

When a student enters a room, the server displays the name and description of the room, plus the names of any objects contained by the room, which may include other players. Each room has a property that determines whether to display the exits or not. Usually this property is set to cause the server to display the names of the exits and their destinations as the last part of the description when a player enters the room. However, signpost rooms have a menu of possible exits in their descriptions and the exit display is suppressed as redundant. The server notifies relevant players when someone enters or leaves an exhibit.

Exits are the directed graph arcs that connect nodes or rooms. An exit has a name and possibly some aliases, like any other object in a MOO, but the name or alias of an exit is a command to use the exit to move to another node. In many MOOs the exits are named spatially, such as up, down, North, East, but in ProgrammingLand they usually identify topics or menu entries.

In Figure 1, the first line is the name of the room. This is followed by the room description, which is several lines. The next to the last line shows that there is an object in this room named simple. The final line indicates that "exit" is the name of an exit that leads to a room named “Practice with the assignment statement.” Should the student type “help simple” or “look simple” the server displays help on how the simple object works. Simple is an instance of a code machine, described above, which is used to show, trace and explain a small portion of programming language code.

> The Three Characteristics of Variables
> Gaining a mastery of programming requires that you be able to declare and use variables. This exhibit looks at the three characteristics in more depth. Since each one of these takes at least one exhibit, choose from the following menu:
> a) What is a legal variable name?
> b) Variable types in C++
> c) The idea of a value, or
> x) Return to the Variable exhibit

Figure 5: a Signpost Room

Figure 5 is an example of a signpost room. This room displays a menu of exit choices; its description indicates the exits, so it has no list of obvious exits from the server. It has no object other than the player in it, so there is no mention of things being present either.

6.2 The Pedagogical Structure

Embedded into the room, exit and object structure of ProgrammingLand are a number of items that work towards the education of students. The basic unit is called a lesson. A lesson covers one distinct topic of the material. It does not have to be exhaustive on the topic, but does need to be self contained. Lessons in ProgrammingLand are usually hierarchical, that is a lesson may contain smaller lessons within it. A lesson may have many of the following parts: a) an introduction that motivates the students or demonstrates the need for the topic, b) the content material, c) some kind of exercise that causes the student to use the new knowledge, and d) some type of assessment of the students grasp of the material. Although there is no attempt at a formal mapping, and while there are many exceptions, the reader can usefully think of a lesson as akin to a chapter in an imaginary textbook.

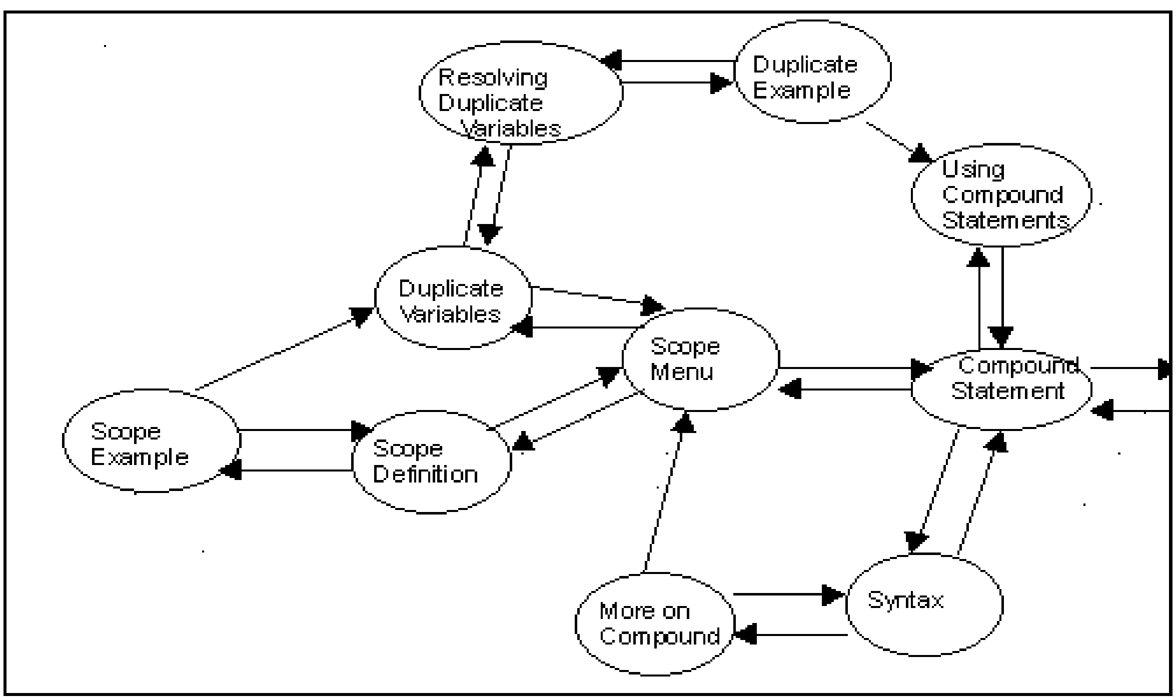

Figure 6: The Exhibits in the Compound Statement Lesson

A lesson in ProgrammingLand usually consists of several exhibits as well as several specialized objects. Typically there is an entryway that is the only way into or out of the lesson. The entryway is often a signpost/menu room, suggesting an order of perusing the material; but the student ultimately decides how to take the lesson. A sample lesson's rooms and exits are diagrammed in figure 6. The Compound Statement exhibit shown towards the right hand side of Figure 6, is the entrance to a lesson. A student entering into the lesson would see the display in figure 7.

> The Compound Statement
> The compound statement is not a flow of control statement, however it is used in most flow of control statements and is essential to the Structured Programming model.
> In the Structured Programming model there is the notion of a block. The block in C++ is the compound statement. It is a wrapper that binds several statements into one. It is also the block that greatly affects the scope of variables.
> You may choose any of the following exhibits to consider next.
> a) The syntax of the compound statement
> b) Scope of variables in compound statement
> c) The compound statement and other statements, or
> x) Return to the main exhibit on flow of control

Figure 7: The Compound Statement Signpost Room

6.2.1 Goals and Assessment

In order for students to learn from the MOO they will have to visit certain rooms that have the needed content and interact with certain objects while there. When a student visits an exhibit that fact is recorded. Likewise when they complete an exercise with an interactive object that fact is also recorded. The lesson dispatcher, described below, checks for these accomplishments.

The Compound Statement room in figure 7 is a lesson room which contains the requirements of the lesson. These requirements are organized as a list of lists. If the student has satisfied any of the sub-lists then they have satisfied the lesson. The requirements on any of these sub-lists may be that a) a certain room has been visited, b) an object has been exercised, or c) another lesson has been completed. However, the student must satisfy every requirement of one sub-list to satisfy the lesson and receive their award.

The Compound Statement exhibit shown in figure 7 allows for two methods of satisfying requirements; either 1) the student visits eight specified rooms within the lesson, or 2) they execute the trace of a code machine found within the lesson's rooms.

When a student leaves the Compound Statement exhibit, a lesson dispatcher checks their completion of the lesson requirements. If the student has satisfied any of them they are given credit for the lesson. Otherwise they are told some or all of the requirements they still need to accomplish, depending on how many there are. Next they are given a choice of either to continue on their way, assuming they will finish the lesson later, or taking a quiz to show their mastery of the material, which is another way to satisfy the lesson requirements.

The lesson, lesson-exit, quiz room, lesson dispatcher, and roving goalie work together to record event tokens that measure student's progress. When a student has visited the important rooms of a lesson, or exercised the machines that exist there, this credit is recorded. When they have enough credit, students are given an "out of MOO" programming assignment by a roving goalie, to finalize their learning.

6.2.2 Lesson Dispatcher

Either the lesson-exit or the quiz room may give the student credit for completing the lesson, and also notify the lesson dispatcher object of the student and the event. Certain lessons are allowed to change the goal of a student which causes the matching roving goalie to be activated which tells them about their new goal.

The lesson-exit is a descendent of the generic exit but behaves in a rather different way. When a student chooses an exit that will leave the lesson, their progress towards satisfaction of the requirements is checked. If the student has previously met the requirements the exit moves the student to their intended destination, tells the student they have completed the lesson, and posts a "completion of lesson" token to their event list.

If they have not completed the lesson, students are asked if they want to continue to their destination and finish the lesson later, or if they want to prove their mastery with a quiz. If they opt for a quiz they are transported to a quiz room (described below).

6.2.3 Goalies

A roving goalie is an object with several important properties for interaction with the student. Generally, students get a separate, personalized assignment, usually a programming assignment from a list of equivalent assignments. The roving goalie may have as few as one, or as many as a hundred, different assignments. When an assignment is given, an index is circularly incremented. If there are more assignments than students each student will get a different assignment. If there are more students then some may receive the same assignment. However, the intention is that students will each get a different assignment, insofar as possible. If the student ever wants to reread their individualized assignment, they use the **Showgoal** command.

The roving goalie records which student received which assignment. In this way, a student may only receive one such goal from a particular lesson, since the lesson dispatcher first checks that they do not have the lesson completed.

6.4 Quiz Rooms

When a student elects to take a mastery quiz, they are transported to a quiz room. A quiz room cannot be reached except by accepting the challenge of a quiz when leaving a lesson. There is one quiz room attached to each lesson, where the student takes a multiple choice quiz. If they pass, they are given full credit for the lesson.

The quiz room randomly generates multiple choice questions which cover lecture material the student missed. Attached to each lecture room is a series of quiz questions on the material covered. Each question consists of three parts, the question, one or more right answers and one or more wrong answers. The quiz generator looks at the player and determines which rooms they did not visit. Then it gathers the questions from these rooms. It reduces the quiz to five questions. If there are fewer than five questions, it accesses some general questions from the lesson room to bring it up to five. It then presents the questions to the student. If they answer incorrectly, the correct answer is given. If they answer four of the five correctly they pass the quiz and receive credit for the lesson. If they miss a second question, the quiz is terminated and they are instructed to resume the lesson to receive credit. If they attempt a second quiz, they get different questions.

The quiz room is used to verify student mastery of material in the absence of the usual evidence: completing the goals as assigned. It is also a way for an expert to short-circuit the lesson structure, if they choose. A quiz room has ***no*** entrances. The only way to enter a quiz room is to take the quiz option when using a lesson-exit.

7. Toys in the Attic

In the spirit of the Exploratorium, the ProgrammingLand MOOseum is populated with a range of demonstrations, toys, robots, and interactive exhibits. These artifacts are intended to engage a visitor in the exploration of the content stored in the museum such that these playful, interactive objects will serve to both entertain and teach. Please note that all of the exhibits and machines in this section were implemented BY STUDENTS.

7.1 Demonstration Machines and Checker Machines

Demonstration machines were built for Lisp functions as an in-class project in the summer of 1998. A class of 50+ students were each assigned a unique Lisp function, and instructed to create a machine with a 'demo' function that would illustrate the operation of the function. These machines are accessed in rooms made specially for them. For example, the Lisp 'cons function' room, implemented by a student, is displayed in figure 8.

```
The cons Cave
You see a white room with writing on the wall in big red letters.  In the center of the room is a cons
machine.
In the far corner is a cons checker machine with six options hanging on the wall behind it.
To operate the machine just type
        plug <number> into cons checker machine
Message on wall: cons [Function] cons x y
        cons is the primitive function to create a new cons cell whose car is x and whose cdr is y. cons
may be thought of as creating a cons, or as adding a new element to the front of a list.
(Information is taken from Common Lisp the Language, 2nd Edition)
You see cons machine, cons checker machine,
#2278)(cons 'a '())=>(a ()),              #2270)(cons nil '(b c))=>(nil b c) or (() b c),
#2273)(cons a (b c))=>(a b c),            #2277)(cons '(a b) '(c d))=>(a b c d),
#2258)(cons 'a '(b c))=>(a b c), and      #2267)(cons '(a b) '(c d))=>((a b) c d) here.
Obvious exits: [exit] to List functions, [lists] to List Manipulation
=>look cons machine
You are looking at a large blue box with a shiny red button.  To see a demonstration type: demo.
=>demo
(cons 'a (cons 'b (cons 'c '())))=>(a b c)
(cons 'a '(b c d))=>(a b c d)
```

Figure 8: the cons machine

The 'cons checker machine' is designed to test a student's application of the information stored in the cons room and demonstrated by the cons machine. To do this, the student must choose from the sample executions that are hanging on the wall, and plug the correct ones into the cons checker machine. The cons checker machine looks and acts as shown in figure 9 (note that cons options are displayed in Figure 8, above, and that object numbers are used as shorthand identifiers):

```
=>look cons checker machine
You see a cons checker machine in the far corner.  The options to check are hanging on the wall around the
machine.  To test an option just type plug <number> into cons checker machine.
=>plug #2270 into cons checker machine
Way to go! The cons checker machine has accepted this as a correct answer.
=>plug #2277 into cons checker machine
According to the cons checker machine this is an incorrect answer.
The following statement is the correct answer:
(cons '(a b) '(c d))=>((a b) c d)
```

Figure 9: the cons checker machine

When the student plugs a correct value into the machine a congratulation message is returned and an award token is added to the player's history. When the player makes a mistake, the feedback includes the correct answer.

7.2 The Recursive Leprechaun

Since recursion is one of the most difficult concepts for students to master, it is important to expose the students to recursion as often as possible. One approach is to implement a recursive leprechaun, which resides in the Realm of Recursion (one of several exhibits implemented by students). The recursion leprechaun demonstrates a recursive counting function in a visually descriptive manner:

```
Realm of Recursion
On the wall you see a poster that reads:
(defunc leprechaun( stuff )
 (cond
    ((nil stuff) nil)
   ((t leprechaun(cdr (stuff)) + 1)))
The above function defines the recursive behavior of this leprechaun.
His syntax is 'count <list> with leprechaun'.
This is still a 'simple' leprechaun so please use the form
( ab cd ef ) or (a b c) for your list; () or ( ) for an empty list.
This will get him to recursively determine the length of a list.
If you receive an error, you most likely failed to match the form shown
You see leprechaun here.
Obvious exits: [exit] to The null function
```

Figure 10: the Realm of Recursion

The Leprechaun gives demonstrations of recursive counting.

```
=>look leprechaun
You see a small green leprechaun carrying a sack.
He tells you that he is a LISP list length leprechaun, and that you can see a demonstration of recursion by following the instructions on the wall.
=>count (a b c) with leprechaun
leprechaun #1 takes a new leprechaun from its sack, keeps one element from the list, and hands the rest to the new leprechaun asking him to count it.
leprechaun #2 takes a new leprechaun from its sack, keeps one element from the list, and hands the rest to the new leprechaun asking him to count it.
leprechaun #3 takes a new leprechaun from its sack, keeps one element from the list, and hands the
rest to the new leprechaun asking him to count it.
leprechaun #4 receives an empty list and counts to 0
He shouts "0!" and leaps into the sack from which he came.
leprechaun #3 adds the one item he still has, shouts "1!", eats the list element, and leaps into the sack from which he came.
leprechaun #2 adds the one item he still has, shouts "2!", eats the list element, and leaps into the sack from which he came.
The leprechaun scratches his head for a moment and then proudly tells you that your list contained 3 objects before he destroyed it.
He then pops the remaining element into his mouth, and goes back about his business, mumbling
something about being used.
```

Figure 11: the Leprechaun Counts

7.3 The Ring Toss Game

The Ring Toss game is intended to provide an amusing challenge in the area of associating programming languages with their historical antecedents.

```
Game room
Welcome to 'Rings and Pegs Game Room'. Here you can play an exciting game of rings and pegs.
You see four rings and twelve pegs here. These rings and pegs are related to each other by some properties.
Try to match a ring with a peg. For that you have to TOSS a RING at the appropriate PEG. If you get it
right, you will be rewarded.  Enjoy!!!!
To play: toss <ring_name> at <peg_name>
You see lisp_ring, c_ring, smalltalk_ring, fortran_ring,
Dennis_Ritchie_peg, John_McCarthy_peg, Alan_Kay_peg, Laning_and_Zierler_peg,
System_Programming_peg Object_Oriented_peg, Scientific_Computing_peg, AI_peg, Year_1957_peg,
Year_1959_peg, Year_1971_peg, and Year_1980_peg here.
Obvious exits: [back] to The remprop function room
```

Figure 12: the ring toss room

The goal of the ring toss game is to associate languages with people and other concepts. More than one ring can be tossed on a single peg (as noted: these are all student projects).

```
=>toss lisp_ring at John_McCarthy_peg
Your ring is still flying in the  air....ooo!!! It has just touched the right  peg...
Yeeeeees!! You got it....You WON
=>toss lisp_ring at Year_1957_peg
Oh! Your ring just MISSED the target by  a whisker....
Sorry!!....You lost...try again.
=>toss lisp_ring at Year_1959_peg
Your ring is still flying in the  air....ooo!!! It has just touched the right  peg...
Yeeeeees!! You got it....You WON
```

Figure 13: playing the ring toss game

7.4 The History Jukebox

The History Jukebox is a device for summarizing programming language history in an entertaining and on-demand fashion.

CS History Jukebox Room

You have entered a grand hall with wonderful furnishings. Lovely shafts of light pour in through skylights in the ceiling. There is a delicately painted sign hanging on the wall.

You see History Jukebox here.

Obvious exits: [trig] to Trigonometric Functions Room, [disco] to Disco Room

=>look History Jukebox

You see a Jukebox. To operate it, type: 'press <song number> on <jukebox name>'

Note: the material for this jukebox was principally borrowed from Sebesta (1999).

The History Jukebox's song list contains

1945 Plankalkul	1967 Simula 67
1953 Laning and Zierler System	1968 Algol 68
1957 FORTRAN I	1971 Pascal
1958 Algol 58	1972 C
1959 Lisp	1973 Prolog
1960 COBOL	1990 Smalltalk
1962 APL	1983 Ada
1964 Basic	1985 C++
1965 PL/1	1995 Java

Figure 14: the History Jukebox

The Jukebox plays historical "tunes".

=>press 1959 on History Jukebox

Lisp : Lisp was designed at MIT by John McCarthy. Modeled after the Lambda Calculus, it was intended to process lists, rather than arrays, for symbolic computation, rather than numbers. Lisp has only two native data types: atoms and lists, and program control is via recursion and conditional statements. It pioneered functional programming and is still the dominant language for AI, although it has largely been replaced by Common Lisp and Scheme, the contemporary dialects (Sebesta, 1996).

Figure 15: playing the History Jukebox

7.5 Tutor Robots

Tutor Robots were implemented to make the function rooms in the LambdaMOO wing more active and engaging. They are created from a prototype Turing Robot provided with the EnCore Moo (Haynes and Hulmevik, 1987), based on the Eliza model (Weizenbaum, 1966) which was inspired by Turing (1950). The Robots are user programmable and capable of matching key words and sentence patterns, and can be implemented with random responses and question responses.

8. ProgrammingLand in the Classroom

We now describe a curricular experience that combined virtual lecture with virtual laboratory to produce a virtual course. In particular, we combined a Virtual Lecture, using the Interactive Video Network (IVN), with a Virtual Laboratory and Museum of Computer Science, the ProgrammingLand MOO (Hill and Slator, 1998), to deliver both lecture-based and hands on instruction to students in remote locations. In doing so we pursued a particular theoretical approach to this new pedagogy - an approach that stresses the importance of virtual environments, authentic experiences, and active learning. We developed a relatively standard IVN course, but then augmented it with networked, multi-player, simulation-based, interactive multi-media - an educational environment that is both immersive and highly interactive (Reid, 1994).

8.1 Comparative Programming Languages: Course Components

NDSU COMP372: Comparative Programming Languages, was offered in the Summer of 1998 and 1999 during the 4-week session, and was comprised of the following elements.

8.1.1 IVN, The North Dakota Interactive Video Network

The North Dakota Interactive Video Network (ND IVN) is a two-way interactive telecommunications system located at many sites throughout the state. Any combination of two to fourteen sites may be connected together for a single event and several events may occur at the same time. Over 25 specially equipped telecommunications classrooms and conference rooms link the 11 North Dakota State University System campuses, the state capitol, 5 tribal colleges in ND, and 25 high schools in the state. In addition, ND IVN has the ability to connect to sites world-wide. ND IVN participants can hear all sites at all times but see only one other site. The Network automatically switches the video to the site that is currently speaking. For the automatic switching to occur, a sound must last about two seconds.

An IVN room is designed to as closely resemble a traditional classroom as practical. Each room had approximately 25 seats. There are two television monitors, one displays the current image and the other the image being transmitted from this location. Each student has a microphone on their table. When a student speaks, then the image from that location is broadcast to the other locations. Thus a reasonable conversation can be carried out; however, the originating site can not tell if their image is being transmitted or not. An instructor can also transmit computer images or the display of a paper that then functions like a blackboard. In this particular course, there were four such sites: two on the NDSU campus, one on the UND campus and one on the VCSU campus.

8.1.2 WWW Syllabus, Assignments, Exams, and E-mail

All pertinent documents, such as the syllabus and assignments, were posted on web pages. In most classes this is a courtesy to students. In this course it was a requirement since none of the locations were within 50 miles of each other.

In such a situation e-mail becomes a critical communication form, since distance keeps face to face conversations at a minimum. E-mail was used for a variety of situations in this course. Assignments that were not MOO-based were handed in through e-mail, with the program and other documentation as an attachment. The time stamp of the e-mail determined whether the item is on-time or late. MOO assignments were handed in by e-mail that announced its completion and specified the object numbers of the finished products. Many of the office visit situations were also handled with e-mail, sometimes more easily than a real visit; questions could be answered and programs could be examined. For example, it is often easier to attach an example program to an e-mail than it is to put it in the students hands in a visit, moreover it is much easier for them to run it later. The time delays of this approach leave something to be desired, but this is remedied by virtual office hours as discussed below.

The importance of e-mail made it crucial that both authors processed all e-mail several times a day. The course had a very short time duration, just four weeks. It was imperative for students to receive quick response to e-mail; and three exchanges with a single student in a single day was not uncommon.

8.2 Course Details

This course was a typical Junior level Programming Languages course for Computer Science majors and minors, the text was Sebesta (1996). The course was completed over the course of four weeks (actually, 18 class meetings). Lectures were multicast daily from an instrumented IVN classroom on the NDSU campus. There were several students in that IVN room, several more in an IVN classroom across campus, and a handful in another IVN classroom on a campus 70 miles to the north. There were 50 students altogether in 1998, and 80 in 1999.

The course syllabus was posted at a website online, and reading assignments (both from the text and from online sources), and homework assignments were posted on that site too. In addition, the details of the homework assignments, as well as information on how to negotiate the pitfalls of electronically submitting homework.

Although hosting the course on the Interactive Video Network, and posting the syllabus on the WorldWide Web were steps in the virtual direction, the technologies of the Internet were employed in even more interesting ways than that. In particular, exams were held outside of class and administered over the Web..

The pedagogical difficulties with online quizzes were a little more profound, because of the offline, self-paced nature of the Web. In this class, tests were posted on Friday and students were given until Monday evening to complete them. This unstructured, unproctored protocol meant students could take quizzes at their leisure with their textbooks open on their lap. As a consequence of all these factors, quizzes were painful to implement and were relatively weak indicators of student progress. Hence, quiz grades were quite high on average.

There was no attempt made to conceal that this approach was as new to the authors as to the students. It was perceived that they responded well and entered into the adventure. This did require flexibility when various technical problems occurred. In the end, the students rated the class highly.

The combination of IVN, WWW, and MOO proved to be quite effective and stronger than any of these alone. The use of the MOO greatly enhanced the effectiveness of the course by reducing the perceived separation of student and instructor. Student evaluations of the course were quite high, with 92% of the students responding rating the quality of the course as either above or much above average. Similarly, 88% of the students believed their understanding of the course content was either good or very good.

8.3. Introductory Programming: Course Components

The ProgrammingLand MOO has been used by a variety of students over a three year period at Valley City State University. It was first used as a supplement in an introductory C++ course in the fall of 1997. It has been used three times in the first C++ course and once in the follow-up C++ course. These courses are the institutions equivalent of CS 1 and CS 2 and are the required introduction to programming for those seeking a degree in Computer Science. However, it has been used once in a BASIC programming language course, and twice for a Java class whose prerequisite is the C++ courses.

In all the uses prior to the 1999-2000 academic year it has been a supplement to the textbook. In the Fall of 1999 the C++ course this was changed. Unlike previous editions of the class the course had no other textbook, using only the MOO. Furthermore MOO use was made mandatory by placing some of the assignments in the MOO. The only way a student could get several of their assignments was to satisfy the requirements of a lesson as described above.

When the lesson requirements are satisfied a MOO entity visits the student and gives them their new assignment. This assignment is unique to them though covering the same material as any other assignment in that lesson. In the fall of 1999 about half of the assignments were delivered through the MOO and about half given in class or on a web page. The spring 2000 CS 2 course has followed that trend. Where there is sufficient material in the MOO for a lesson, the assignment is given by the MOO.

9. Plan of Work

ProgrammingLand is not currently complete in terms of content, nor is it self-contained. The student still needs access to a programming language system for compiling, software development, and testing. Thus the goals assigned by the roving goalie are externally satisfied. One important goal of this project is the automatic evaluation of student programs, such as has been done by the Ceilidh system (Foubister, Michaelson, and Tomes, 1997). Ceilidh works by having students complete a large number of small coding assignments where the input-output behavior is prescribed in advance, and data sets are provided. The system scores student work by matching against a set of idealized solutions. This avoids the general problem of evaluating programs using automatic means, and allows the system to make assessments on both correctness and style.

Our version of the Ceilidh system will be implemented as Java applets interfacing with the network connection objects built into the MOOseum. The student will create and edit their code within the MOO, then give a command to submit the code. The MOOseum will transfer the code to the applet, which processes it and sends back the results. This approach has the advantage of off-loading the processing, making load balancing easier. In addition, the interfacing on both machines is relatively simple; the tools to compile, execute, analyze the style and test results may be written in any language for any OS or existing code used. This would cause the MOOseum to be a substantially more effective self-contained educational system.

The Ceilidh module will be the major development thrust of the project, and will occupy most of the faculty and graduate student effort throughout the life of the project.

The MOOseum currently contains slightly more than 1000 exhibits, which we judge to represent about 70% of the necessary C++ material, and while the museum metaphor is very apt for the introductory learning of material, it lacks a very nice feature that a lowly textbook always provides, namely an index. Students report that finding a particular topic, once they have studied it, can be somewhat tedious. We propose to implement a site map, or conceptual index, that will allow quick access to the material. We anticipate that an automated and extensible index will require the efforts of an undergraduate student for roughly the first year of the project.

Introductory programming is mainly text based so the MOOseum, as a text-only medium, has not been as restrictive as in other areas. However, certain machines, such as the workbench, are clumsy using just text and could be much easier to use with a graphical interface. Furthermore, ProgrammingLand is currently a first semester programming course aid. The emphasis is much more on syntax and semantics than on techniques. As the content expands to more advanced topics the use of graphics to demonstrate these topics will increase. For example the MOO currently has little content on sorting, dynamic data structures or graphical user interfaces. These would be much easier to handle with graphical displays or graphical machines. We anticipate this effort commencing in the second year and continuing through to the end, and requiring the efforts of one or two students full-time.

The rudiments for student progress and monitoring are in place in the lesson structure, awards and events properties. However, there is no convenient interface for either the instructor or the student. Students need to be partitioned into classes connected with instructors. Instructors need to be able to ask: “Who has not done lesson X?” Students need to be able to ask: “What am I lacking for the completion of my next lesson?” and “Where am I compared to the class average or norm?” and “What do I need to do to finish this course?” We anticipate an undergraduate student working half-time can complete this reporting module within a year.

Assessment of student learning, and course evaluation, will be an ongoing concern throughout the life of the project. The promise of this sort of project is a system that offers improved interactivity, better learner centered control, and increased access to content, while showing no significant change in student performance for either better or worse. Still, we cannot afford to be cavalier where student progress is at stake, and continual oversight is warranted, both in terms of the assessment of student achievement and the evaluation of course content. We anticipate employing a half-time graduate student to do this work, with frequent, regularly scheduled, data collection, analysis, and report writing throughout the course of the project.

10. Dissemination

At the end of this 3-year project, we propose to “publish” the ProgrammingLand MOOseum for the benefit of a wide educational audience. The form this will take is not entirely clear, but one obvious distribution mechanism is via CD-ROM. Since the MOOseum will be, by this time, almost entirely self-contained, we anticipate it will be easily incorporated into most undergraduate curriculums, with potential for distribution at the high school level as well. We would plan to write an instructors manual, and promote the use of the MOOseum through published descriptions in the Computer Science Education literature.

References

Curtis, Pavel (1992). Mudding: Social Phenomena in Text-Based Virtual Realities. Proceedings of the conference on Directions and Implications of Advanced Computing (sponsored by Computer Professionals for Social Responsibility)

Duffy, T.M. Lowyck, J. and Jonassen, D.H. (1983). Designing environments for constructive Learning. New York: Springer-Verlag

Duffy, T.M. and Jonassen, D.H. (1992). Constructivism: new implications for instructional technology. In Duffy and Jonassen (eds.), Constructivism and the Technology of Instruction. Hillsdale: Lawrence Erlbaum.

Foubister, S.P., G.J. Michaelson, and N. Tomes (1997) Automatic Assessment of Elementary Standard ML programs using Ceilidh. Journal of Computer Assisted Learning, 13(2), pp. 99-108.

Haynes, Cynthia , and Jan Rune Holmevik, eds (1997), High Wired: On the Design, Use and Theory of Educational MOOs. University of Michigan.

Hill, Curt and Brian M. Slator (1998) Virtual Lecture, Virtual Laboratory, or Virtual Lesson. In the *Proceedings of the Small College Computing Symposium (SCCS98)*. Fargo-Moorhead, April. pp. 159-173.

Juell, Paul (1999). Educational Opportunities using VRML in the Classroom. International Conference on Mathematics/Science Education and Technology, March 1-4, San Antonio, Texas.

Reid, T Alex (1994) Perspectives on computers in education: the promise, the pain, the prospect Active Learning. 1(1), Dec. CTI Support Service. Oxford, UK

Saini-Eidukat, Bernhardt, Don Schwert and Brian M. Slator (1999). Designing, Building, and Assessing a Virtual World for Science Education. Proceedings of the 14th International Conference on Computers and Their Applications (CATA-99), April 7-9, Cancun

Schwert, D.P., Brian M. Slator, B. Saini-Eidukat, (1999). A Virtual World For Earth Science Education In Secondary And Post-Secondary Environments: The Geology Explorer. International Conference on Mathematics/Science Education &Technology, March 1-4, 1999, San Antonio, TX.

Sebesta, Robert W. (1996). Concepts of Programming Languages (3rd Edition). New York: Addison Wesley, ISBN 0-8053-7133-8.

Slator, Brian M. and Harold "Cliff" Chaput (1996). Learning by Learning Roles: a virtual role-playing environment for tutoring. Proceedings of the Third International Conference on Intelligent Tutoring Systems (ITS'96). Montréal: Springer-Verlag, June 12-14, pp. 668-676. (Lecture Notes in Computer Science, edited by C. Frasson, G. Gauthier, A. Lesgold).

Slator, Brian M., D. Schwert, B. Saini-Eidukat, P. McClean, J. Abel, J. Bauer, B. Gietzen, N. Green, T. Kavli, L. Koehntop, B. Marthi, V. Nagareddy, A. Olson, Y. Jia, K. Peravali, D. Turany, B. Vender, J. Walsh (1998). Planet Oit: a Virtual Environment and Educational Role-playing Game to Teach the Geosciences. In the Proceedings of the Small College Computing Symposium (SCCS98). Fargo-Moorhead, April. pp. 378-392.

Slator, Brian M. and Curt Hill (1999). Mixing Media For Distance Learning: Using IVN And Moo In Comp372. World Conference on Educational Media, Hypermedia and Telecommunications (ED-MEDIA 99), June 19-24, Seattle, WA.

Slator, Brian M., P. Juell, P.E. McClean, B. Saini-Eidukat, D.P. Schwert, A. White, C. Hill (1999a). Virtual Environments for Education at NDSU. World Conference on Educational Media, Hypermedia and Telecommunications (ED-MEDIA 99), June 19-24, Seattle, WA.

Slator, Brian M., Donald Schwert, Bernhardt Saini-Eidukat (1999b). Phased Development of a Multi-Modal Virtual Educational World. Proceedings of the International Conference on Computers and Advanced Technology in Education (CATE'99), Cherry Hill, NJ, May 6-8

Turing, Alan (1950). Computing Machinery and Intelligence. Mind, 65(236), pp. 433-460. Reprinted in ``Computers and Thought'' (1963). Edited by Feigenbaum and Feldman. New York: McGraw-Hill.

Unkel, Christopher (1997). WinMOO. http://www-personal.engin.umich.edu/~cunkel

Weizenbaum, Joseph (1966). ELIZA -- a Computer Program for the Study of Natural Language Communication between Man and Machine. Communications of the ACM, vol. 9, pp. 36-45

White, Alan R., Phillip E. McClean, and Brian M. Slator (1999). The Virtual Cell: An Interactive, Virtual Environment for Cell Biology. World Conference on Educational Media, Hypermedia and Telecommunications (ED-MEDIA 99), June 19-24, Seattle, WA.

Brian Michael Slator
Associate Professor
Computer Science Dept, North Dakota State University,
IACC Bldg, Rm. 258-A6, Fargo, ND 58105
slator@badlands.nodak.edu http://www.cs.ndsu.nodak.edu/~slator/
(office) 701-231-6124 (home) 701-271-8875

Education

Ph.D. 1988 Computer Science (Related Area: Linguistics), New Mexico State University, Las Cruces, NM; Dissertation: Lexical Semantics and Preference Semantics Analysis; Advisor: Dr. Yorick A. Wilks.

M.S. 1985 Computer Science (Minor: Linguistics), New Mexico State University, Las Cruces, NM; Thesis: Natural Language Interface: An Algorithm for Design.

B.S. 1983 Computer Science (with Honors; second major in English), University of Wisconsin - La Crosse.

Employment

1996-present Associate Professor, Computer Science Dept., North Dakota State University.

1992-1996 Assistant Professor (Research), The Institute for the Learning Sciences, Northwestern University.

1990-1992 Research Associate, The Institute for the Learning Sciences, Northwestern University.

1989-1992 Research Specialist, Natural Language Group, (Summer position), Computing Research Laboratory, New Mexico State University.

1988-1990 Assistant Professor, Department of Computer Science, North Dakota State University.

1985-1988 Graduate Research Fellow, Natural Language Group, Computing Research Laboratory, New Mexico State University.

Up to 5 Publications Most Closely Related

1. Slator, Brian M. (1999). Intelligent Tutors in Virtual Worlds. *8th International Conference on Intelligent Systems*. Denver, CO. June 24-26, pp. 124-127.
2. Slator, B.M., P. Juell, P.E. McClean, B. Saini-Eidukat, D.P. Schwert, A. White, C. Hill (1999). Virtual Environments for Education at NDSU. *World Conference on Educational Media, Hypermedia and Telecommunications (ED-MEDIA 99)*, June 19-24, Seattle, WA, pp. 875-880. (*Outstanding Paper Award*)
3. Schwert, D.P., B.M. Slator, B. Saini-Eidukat, (1999). A Virtual World For Earth Science Education In Secondary And Post-Secondary Environments: The Geology Explorer. *International Conference on Mathematics/Science Education &Technology (MSET-99),* March 1-4, San Antonio, TX, pp. 519-525.
4. Slator, Brian M., Donald Schwert, Bernhardt Saini-Eidukat (1999). Phased Development of a Multi-Modal Virtual Educational World. *Proceedings of the International Conference on Computers and Advanced Technology in Education (CATE'99),* Cherry Hill, NJ, May 6-8, pp. 92-96
5. Slator, Brian M. and Golam Farooque (1998). The Agents in an Agent-based Economic Simulation Model. *11th International Conference on Computer Applications in Industry And Engineering (CAINE-98)* November 11-13, 1998, Las Vegas, Nevada USA, pp. 175-179.

Up to 5 Other Significant Publications

1. Slator, Brian M. and Harold "Cliff" Chaput (1996). Learning by Learning Roles: a virtual role-playing environment for tutoring.*Third International Conference on*

Intelligent Tutoring Systems (ITS'96). Montreal: Springer-Verlag, June 12-14, pp. 668-676.
2. Manaris, Bill and Brian M. Slator (1996). Interactive Natural Language Processing: Building on Success. *IEEE Computer*. (Special Edition on Interactive Natural Language Processing, Edited by Manaris and Slator). July. pp. 28-32.
3. Guthrie, Louise, James Pustejovsky, Yorick A. Wilks and Brian M. Slator (1996). The Role of Lexicons in Natural Language Processing. *Communications of the ACM*. (Special Edition on Natural Language Processing, edited by Yorick A. Wilks), Vol. 39, No. 1, pp. 63-72.
4. Wilks, Yorick A., Brian M. Slator, and Louise Guthrie (1996). *Electric Words: Dictionaries, Computers and Meanings.* Cambridge, MA: MIT Press..
5. Slator, Brian M. and Christopher K. Riesbeck (1992). TaxOps: a case-based advisor. *International Journal of Expert Systems,* Vol. 4, No. 2, pp. 117-140 (Special issue on Case-based Reasoning, edited by Evangelous Simoudis), JAI Press: Greenwich, CT.

List of collaborators during the last 48 months

Roger Schank, Chris Riesbeck, Ray Bareiss, Alex Kass, Gregg Collins, Tom Hinrichs, Cliff Chaput, Bob Hooker, Scott MacQuarrie, Kerim Fidel: Institute for the Learning Sciences at Northwestern University

Kendall Nygard, Mark Pavicic, Ken Magel, Paul Juell, Bill Perrizo: Computer Science Dept. at NDSU

Phil McClean, Alan White, Don Schwert, Bernhardt Saini-Eidukat Joseph Latimer, Jeffrey Clark, Plant Science, Botany/Biology, Geosciences, Business, and Sociology/Anthropology Departments at NDSU

Richard Beckwith, Intel Corporation

Yorick Wilks: Sheffield University

Louise Guthrie: University of Texas, El Paso

Bill Manaris: University of Southwest Louisiana

Graduate and Postdoctoral Advisors

Dr. Yorick A. Wilks, Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield, UK, S14DP (ph: 011-44/742-825-571; email: yorick@dcs.sheffield.ac.uk)

Other Relevant Information

Dr. Brian M. Slator is Associate Professor of Computer Science at North Dakota State University. He has broad experience with the design, development and implementation of a number of MUD and MOO environments for learning. For six years as a research scientist at the Northwestern University Institute for the Learning Sciences (ILS), he designed and managed the development of a number of multimedia applications in educational technology, case-based reasoning for intelligent tutoring, and job-aid style performance support (e.g. Slator and Riesbeck, 1992; Slator and Chaput, 1996). Most relevant to this proposal, he was the architect of an interactive, multi-user retailing game, its economic simulation, software agent-based tutoring, and manager of the GAMES project. Since joining North Dakota State University, he is directly involved with no less than five graphically oriented educational media projects. He has taught courses in user interface design and human computer interaction, and he is experienced in dealing with the issues involved with both developing virtual worlds and designing graphical user interface.

Curtis Dean Hill
Assistant Professor
Mathematics Dept, Valley City State University,
Rhoades Science Center, Rm. 110, Valley City, ND 58072
Curt_Hill@mail.vcsu.nodak.edu
http://www.vcsu.nodak.edu/offices/MST/Faculty/Curt_Hill/curt.htm
(office) 701-845-7103 (home) 701-845-4584

Education

M.S. 1981 Computer Science,
University of Nebraska, Lincoln, NE;
Thesis: Concurrent Extensions to the Programming Language Pascal.
B.S. 1974 Mathematics (second major in General Science),
University of Iowa – Iowa City.

Employment

1995-present Assistant Professor, Mathematics Dept., Valley City State University.
1990-1995 Instructor, Computer Science Dept., Iowa State University.
1983-1990 Instructor, Des Moines Area Community College, Boone Campus, Iowa.
1978-1983 Academic Programmer/Analyst, University of Nebraska.
1976-1977 Programmer, First Data Resources, Omaha, NE.
1974-1976 Programmer, Mutual of Omaha, Omaha, NE.

Publications Most Closely Related

1. Hill, Curt , Slator, Brian M. (2000). Computer Science Instruction in a Virtual World. *World Conference on Educational Media, Hypermedia and Telecommunications (ED-MEDIA 2000)*, June 26-July 1 2000, Montreal, Quebec, Canada.
2. Slator, Brian M., C. Hill (1999). Mixing Media for Distance Learning. *World Conference on Educational Media, Hypermedia and Telecommunications (ED-MEDIA 99)*, June 19-24, Seattle, WA pp. 881-886.
3. Slator, Brian M., P. Juell, P.E. McClean, B. Saini-Eidukat, D.P. Schwert, A. White, Curt Hill (1999). Virtual Environments for Education at NDSU. *World Conference on Educational Media, Hypermedia and Telecommunications (ED-MEDIA 99)*, June 19-24, Seattle, WA, pp. 875-880. (*Outstanding Paper Award*).
4. Hill, Curt, Slator, Brian M. (1998). Virtual Lecture, Virtual Laboratory or Virtual Lesson. *Small College Computing Symposium (SCCS'98)*, April 1998, Fargo, ND and Moorhead, MN, pp. 159-173.
5. Hill, Curt (1999). Extracting Data from a MOO. *Small College Computing Symposium (SCCS'99)*. La Crosse, WI, April 15-17.

Other Significant Publications

1. Hill, Curt and Paul Juell (2000). Telecommuting Arrives in the Classroom. *Midwest Instructional Computing Symposium (MICS'2000)*. St. Paul, Minnesota: April 2000.
2. Slator, Brian M., Jeff Clark, Paul Juell, Joe Latimer, Phil McClean, Bernhardt Saini-Eidukat, Don Schwert, Alan White, with Curt Hill and others (1999). Research and Development of Virtual Worlds for Immersive Instruction. In the *Proceedings of the Small College Computing Symposium (SCCS99)*. La Crosse, WI, April 15-17.

3. Hill, Curt (1998). Teaching with the Builders. *Inprise Developer's Conference (ICON'99)*. Denver Colorado: August 1998.

List of collaborators during the last 48 months

Paul Juell, Brian Slator: Computer Science Dept. at NDSU
Phil McClean, Alan White, Don Schwert, Bernhardt Saini-Eidukat, Plant Science, Botany/Biology, Geosciences, Business, and Sociology/Anthropology Departments at NDSU
Richard Leyton of Sybase

Graduate and Postdoctoral Advisors

Dr. David W. Embley, Computer Science Department, Brigham Young University, Provo, Utah, 84602 (ph: 801-378-6470; email: david_embley@byu.edu)

Other Relevant Information

Curt Hill is an Assistant Professor in the Mathematics Department at Valley City State University. He has focused his attention on the issues of undergraduate education in Computer Science. His main research interest is currently the ProgrammingLand MOO for online instruction of introductory programming skills in an interactive and immersive environment. For the summer of 1999, and again in 2000, he was awarded an ND-EPSCoR FLARE (Faculty Laboratory and Research Experience) grant, for his project entitled, "The Interactions of the Virtual Museum".

SUMMARY PROPOSAL BUDGET

YEAR 1

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months)	
	Proposed	Granted
AWARD NO.		

ORGANIZATION
North Dakota State University Fargo

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Brian M Slator

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Brian M Slator - none	0.00	0.00	2.00	$ 12,800	$
2.					
3.					
4.					
5.					
6. (0) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (1) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	12,800	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (2) GRADUATE STUDENTS				24,000	
4. (1) UNDERGRADUATE STUDENTS				8,064	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				44,864	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				4,161	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				49,025	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.) Workstation $ 3,000					
TOTAL EQUIPMENT				3,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				1,500	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ 0					
2. TRAVEL 0					
3. SUBSISTENCE 0					
4. OTHER 0					
(0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				500	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				0	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				0	
5. SUBAWARDS				30,598	
6. OTHER				0	
TOTAL OTHER DIRECT COSTS				31,098	
H. TOTAL DIRECT COSTS (A THROUGH G)				84,623	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) H-D (Rate: 41.0000, Base: 76025)					
TOTAL INDIRECT COSTS (F&A)				31,170	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				115,793	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 115,793	$

M. COST SHARING PROPOSED LEVEL $ 0 | AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY		
Brian M Slator		INDIRECT COST RATE VERIFICATION		
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked	Date Of Rate Sheet	Initials - ORG

SUMMARY PROPOSAL BUDGET

YEAR 2

ORGANIZATION **North Dakota State University Fargo**				FOR NSF USE ONLY PROPOSAL NO.	DURATION (months) Proposed	Granted
PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR **Brian M Slator**				AWARD NO.		
A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR		Funds Requested By proposer	Funds granted by NSF (if different)
1. Brian M Slator - none	0.00	0.00	2.00		$ 13,440	$
2.						
3.						
4.						
5.						
6. (0) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00		0	
7. (1) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00		13,440	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)						
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00		0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00		0	
3. (2) GRADUATE STUDENTS					24,000	
4. (1) UNDERGRADUATE STUDENTS					8,064	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)					0	
6. (0) OTHER					0	
TOTAL SALARIES AND WAGES (A + B)					45,504	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)					4,353	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)					49,857	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.) Workstation $ 3,000						
TOTAL EQUIPMENT					3,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)					1,500	
2. FOREIGN					0	
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 0 2. TRAVEL 0 3. SUBSISTENCE 0 4. OTHER 0						
(0) TOTAL PARTICIPANT COSTS					0	
G. OTHER DIRECT COSTS						
1. MATERIALS AND SUPPLIES					500	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION					0	
3. CONSULTANT SERVICES					0	
4. COMPUTER SERVICES					0	
5. SUBAWARDS					31,400	
6. OTHER					0	
TOTAL OTHER DIRECT COSTS					31,900	
H. TOTAL DIRECT COSTS (A THROUGH G)					86,257	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) H-D (Rate: 41.0000, Base: 51857)						
TOTAL INDIRECT COSTS (F&A)					21,261	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)					107,518	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)					0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)					$ 107,518	$
M. COST SHARING PROPOSED LEVEL $ 0	AGREED LEVEL IF DIFFERENT $					
PI / PD TYPED NAME & SIGNATURE* **Brian M Slator**	DATE			FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION		
ORG. REP. TYPED NAME & SIGNATURE*	DATE			Date Checked	Date Of Rate Sheet	Initials - ORG

 ***SIGNATURES REQUIRED ONLY FOR REVISED BUDGET (GPG III.B)**

SUMMARY PROPOSAL BUDGET

YEAR 3

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months)	
	Proposed	Granted
AWARD NO.		

ORGANIZATION
North Dakota State University Fargo

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Brian M Slator

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Brian M Slator - none	0.00	0.00	2.00	$ 14,112	$
2.					
3.					
4.					
5.					
6. (0) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (1) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	14,112	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (2) GRADUATE STUDENTS				24,000	
4. (1) UNDERGRADUATE STUDENTS				8,064	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				46,176	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				4,554	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				50,730	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.) Workstation $ 3,000					
TOTAL EQUIPMENT				3,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				1,500	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 0 2. TRAVEL 0 3. SUBSISTENCE 0 4. OTHER 0					
(0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				500	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				0	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				0	
5. SUBAWARDS				32,243	
6. OTHER				0	
TOTAL OTHER DIRECT COSTS				32,743	
H. TOTAL DIRECT COSTS (A THROUGH G)				87,973	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) H-D (Rate: 41.0000, Base: 52730)					
TOTAL INDIRECT COSTS (F&A)				21,619	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				109,592	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 109,592	$

M. COST SHARING PROPOSED LEVEL $ 0 AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY		
Brian M Slator		INDIRECT COST RATE VERIFICATION		
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked	Date Of Rate Sheet	Initials - ORG

SUMMARY PROPOSAL BUDGET

Cumulative

ORGANIZATION: North Dakota State University Fargo	FOR NSF USE ONLY — PROPOSAL NO. / DURATION (months): Proposed / Granted
PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR: Brian M Slator	AWARD NO.

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Brian M Slator - none	0.00	0.00	6.00	$ 40,352	$
2.					
3.					
4.					
5.					
6. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (1) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	6.00	40,352	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (6) GRADUATE STUDENTS				72,000	
4. (3) UNDERGRADUATE STUDENTS				24,192	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				136,544	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				13,068	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				149,612	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.) $ 9,000					
TOTAL EQUIPMENT				9,000	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				4,500	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 0 2. TRAVEL 0 3. SUBSISTENCE 0 4. OTHER 0					
(0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				1,500	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				0	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				0	
5. SUBAWARDS				94,241	
6. OTHER				0	
TOTAL OTHER DIRECT COSTS				95,741	
H. TOTAL DIRECT COSTS (A THROUGH G)				258,853	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE)					
TOTAL INDIRECT COSTS (F&A)				74,050	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				332,903	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 332,903	$

M. COST SHARING PROPOSED LEVEL $ 0	AGREED LEVEL IF DIFFERENT $	
PI / PD TYPED NAME & SIGNATURE*: Brian M Slator	DATE	FOR NSF USE ONLY — INDIRECT COST RATE VERIFICATION
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date Of Rate Sheet / Initials - ORG

Budget Justification Page

This project is a collaboration between North Dakota State University (NDSU) and Valley City State University (VCSU), both located in North Dakota, an EPSCoR state. The PI is on the Computer Science faculty at NDSU. The budget for VCSU is brought forward as a Subaward on the NDSU budget.

Summer support is requested for the PIs at both institutions.

Salary is requested to hire two full-time graduate students and one full-time undergraduate at NDSU, and one full-time undergraduate at VCSU. These students will work on software design and implementation, as well as assessment and evaluation studies.

Funds for three computer workstations are requested for students and faculty to use for software development and data analysis.

Travel funds are requested for faculty and students to attend regional and national conferences.

Funds are requested for materials and supplies as they are needed in the course of the project.

SUMMARY PROPOSAL BUDGET

YEAR 1

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months)	
	Proposed	Granted
AWARD NO.		

ORGANIZATION
Valley City State University

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Curtis Hill

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Curtis Hill	0.00	0.00	2.00	$ 8,400	$
2.					
3.					
4.					
5.					
6. (0) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (1) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	8,400	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (0) GRADUATE STUDENTS				0	
4. (1) UNDERGRADUATE STUDENTS				8,064	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				16,464	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				2,601	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				19,065	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				0	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				1,500	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ 0					
2. TRAVEL 0					
3. SUBSISTENCE 0					
4. OTHER 0					
(0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				250	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				0	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				0	
5. SUBAWARDS				0	
6. OTHER				0	
TOTAL OTHER DIRECT COSTS				250	
H. TOTAL DIRECT COSTS (A THROUGH G)				20,815	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) H-D (Rate: 47.0000, Base: 20815)					
TOTAL INDIRECT COSTS (F&A)				9,783	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				30,598	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 30,598	$

M. COST SHARING PROPOSED LEVEL $ 0 AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY		
Curtis Hill		INDIRECT COST RATE VERIFICATION		
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked	Date Of Rate Sheet	Initials - ORG

SUMMARY PROPOSAL BUDGET COMMENTS - Year 1

SUMMARY PROPOSAL BUDGET

YEAR 2

ORGANIZATION **Valley City State University**				FOR NSF USE ONLY PROPOSAL NO.	DURATION (months) Proposed / Granted
PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR **Curtis Hill**				AWARD NO.	

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Curtis Hill	0.00	0.00	2.00	$ 8,820	$
2.					
3.					
4.					
5.					
6. (0) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (1) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	8,820	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (0) GRADUATE STUDENTS				0	
4. (1) UNDERGRADUATE STUDENTS				8,064	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				16,884	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				2,727	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				19,611	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				0	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				1,500	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 0 2. TRAVEL 0 3. SUBSISTENCE 0 4. OTHER 0					
(0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				250	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				0	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				0	
5. SUBAWARDS				0	
6. OTHER				0	
TOTAL OTHER DIRECT COSTS				250	
H. TOTAL DIRECT COSTS (A THROUGH G)				21,361	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) H-D (Rate: 47.0000, Base: 21361)					
TOTAL INDIRECT COSTS (F&A)				10,039	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				31,400	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 31,400	$

M. COST SHARING PROPOSED LEVEL $ 0	AGREED LEVEL IF DIFFERENT $			
PI / PD TYPED NAME & SIGNATURE* Curtis Hill	DATE	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION		
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked	Date Of Rate Sheet	Initials - ORG

SUMMARY PROPOSAL BUDGET

YEAR 3

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months)	
	Proposed	Granted
AWARD NO.		

ORGANIZATION
Valley City State University

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Curtis Hill

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Curtis Hill	0.00	0.00	2.00	$ 9,261	$
2.					
3.					
4.					
5.					
6. (0) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (1) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	9,261	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (0) GRADUATE STUDENTS				0	
4. (1) UNDERGRADUATE STUDENTS				8,064	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				17,325	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				2,859	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				20,184	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				0	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				1,500	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ 0					
2. TRAVEL 0					
3. SUBSISTENCE 0					
4. OTHER 0					
(0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				250	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				0	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				0	
5. SUBAWARDS				0	
6. OTHER				0	
TOTAL OTHER DIRECT COSTS				250	
H. TOTAL DIRECT COSTS (A THROUGH G)				21,934	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) H-D (Rate: 47.0000, Base: 21934)					
TOTAL INDIRECT COSTS (F&A)				10,308	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				32,242	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 32,242	$

M. COST SHARING PROPOSED LEVEL $ 0 AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY		
Curtis Hill		INDIRECT COST RATE VERIFICATION		
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked	Date Of Rate Sheet	Initials - ORG

SUMMARY PROPOSAL BUDGET — Cumulative

ORGANIZATION: Valley City State University				FOR NSF USE ONLY		
				PROPOSAL NO.	DURATION (months) Proposed	Granted
PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR: Curtis Hill				AWARD NO.		
A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)	
1. Curtis Hill	0.00	0.00	6.00	$ 26,481	$	
2.						
3.						
4.						
5.						
6. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0		
7. (1) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	6.00	26,481		
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)						
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0		
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0		
3. (0) GRADUATE STUDENTS				0		
4. (3) UNDERGRADUATE STUDENTS				24,192		
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0		
6. (0) OTHER				0		
TOTAL SALARIES AND WAGES (A + B)				50,673		
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				8,187		
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				58,860		
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)						
TOTAL EQUIPMENT				0		
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				4,500		
2. FOREIGN				0		
F. PARTICIPANT SUPPORT COSTS						
1. STIPENDS $ 0						
2. TRAVEL 0						
3. SUBSISTENCE 0						
4. OTHER 0						
(0) TOTAL PARTICIPANT COSTS				0		
G. OTHER DIRECT COSTS						
1. MATERIALS AND SUPPLIES				750		
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				0		
3. CONSULTANT SERVICES				0		
4. COMPUTER SERVICES				0		
5. SUBAWARDS				0		
6. OTHER				0		
TOTAL OTHER DIRECT COSTS				750		
H. TOTAL DIRECT COSTS (A THROUGH G)				64,110		
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE)						
TOTAL INDIRECT COSTS (F&A)				30,131		
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				94,241		
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				0		
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 94,241	$	
M. COST SHARING PROPOSED LEVEL $ 0	AGREED LEVEL IF DIFFERENT $					
PI / PD TYPED NAME & SIGNATURE*: Curtis Hill	DATE			FOR NSF USE ONLY — INDIRECT COST RATE VERIFICATION		
ORG. REP. TYPED NAME & SIGNATURE*	DATE			Date Checked	Date Of Rate Sheet	Initials - ORG

Current and Pending Support

(See GPG Section II.D.8 for guidance on information to include on this form.)

The following information should be provided for each investigator and other senior personnel. Failure to provide this information may delay consideration of this proposal.

Investigator: **Brian Slator**

Other agencies (including NSF) to which this proposal has been/will be submitted.

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **A Shared Development Environment for Science-based Courseware**

Source of Support: **NSF/DUE**
Total Award Amount: $ **155,000** Total Award Period Covered: **02/01/98 - 01/31/01**
Location of Project: **North Dakota State University**
Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **1.00** Sumr: **0.00**

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **Learning by Doing Physical Geology in a Virtual Laboratory/Virtual Field Trip Computer Environment**

Source of Support: **NSF/GEO**
Total Award Amount: $ **49,981** Total Award Period Covered: **10/01/99 - 09/30/01**
Location of Project: **North Dakota State University**
Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **1.00** Sumr: **0.00**

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **The Virtual Center for Plant Genomics**

Source of Support: **NSF-EPSCoR**
Total Award Amount: $ **403,475** Total Award Period Covered: **07/01/99 - 06/30/01**
Location of Project: **Montana State University, subaward to North Dakota State**
Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **0.50** Sumr: **0.00**

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **New Directions in Virtual Geoscience Education**

Source of Support: **NSF-CCLI**
Total Award Amount: $ **74,192** Total Award Period Covered: **05/01/00 - 04/30/01**
Location of Project: **North Dakota State University**
Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **1.00** Sumr: **1.00**

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **Intelligent Tutoring Using Case-based Reasoning**

Source of Support: **North Dakota EPSCoR**
Total Award Amount: $ **38,273** Total Award Period Covered: **05/01/00 - 04/30/02**
Location of Project: **North Dakota State University**
Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **1.00** Sumr: **1.00**

*If this project has previously been funded by another agency, please list and furnish information for immediately preceding funding period.

Current and Pending Support

(See GPG Section II.D.8 for guidance on information to include on this form.)

The following information should be provided for each investigator and other senior personnel. Failure to provide this information may delay consideration of this proposal.

Investigator: **Brian Slator**

Other agencies (including NSF) to which this proposal has been/will be submitted.

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title: **Learning Computer Science in Virtual Worlds**

Source of Support: **NSF-ITR**
Total Award Amount: $ **332,903** Total Award Period Covered: **08/01/00 - 07/31/03**
Location of Project: **North Dakota State University**
Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **1.00** Sumr: **2.00**

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title: **Real World Geology on a Virtual Planet**

Source of Support: **NSF-ITR**
Total Award Amount: $ **466,122** Total Award Period Covered: **09/01/00 - 08/31/03**
Location of Project: **North Dakota State University**
Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **1.00** Sumr: **2.00**

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title: **Learning in a Virtual, Interactive World**

Source of Support: **NSF-ITR**
Total Award Amount: $ **497,463** Total Award Period Covered: **08/01/00 - 07/31/03**
Location of Project: **North Dakota State University**
Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **1.00** Sumr: **1.00**

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title: **Training Future Scientists with a Virtual Cell**

Source of Support: **Dept. of Education - FIPSE Comprehensive Program**
Total Award Amount: $ **202,000** Total Award Period Covered: **10/01/00 - 09/30/03**
Location of Project: **North Dakota State University**
Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **1.00** Sumr: **2.00**

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title:

Source of Support:
Total Award Amount: $ Total Award Period Covered:
Location of Project:
Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

*If this project has previously been funded by another agency, please list and furnish information for immediately preceding funding period.

Current and Pending Support

(See GPG Section II.D.8 for guidance on information to include on this form.)

The following information should be provided for each investigator and other senior personnel. Failure to provide this information may delay consideration of this proposal.

Investigator: **Curtis Hill**

Other agencies (including NSF) to which this proposal has been/will be submitted.

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **Learning Computer Science in Virtual Worlds**

Source of Support: **NSF-ITR**

Total Award Amount: $ **332,903** Total Award Period Covered: **08/01/00 - 07/31/03**

Location of Project: **North Dakota State University**

Person-Months Per Year Committed to the Project. Cal: **0.00** Acad: **1.00** Sumr: **2.00**

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

---

*If this project has previously been funded by another agency, please list and furnish information for immediately preceding funding period.

FACILITIES, EQUIPMENT & OTHER RESOURCES

FACILITIES: Identify the facilities to be used at each performance site listed and, as appropriate, indicate their capacities, pertinent capabilities, relative proximity, and extent of availability to the project. Use "Other" to describe the facilities at any other performance sites listed and at sites for field studies. USE additional pages as necessary.

Laboratory:

Clinical:

Animal:

Computer: **Each PI has a personal workstation on their desk. In addition, Slator has a lab space with 4 workstations and 3 Unix servers, served by a 100 MBit switch.**

Office:

Other: ____________________

MAJOR EQUIPMENT: List the most important items available for this project and, as appropriate identifying the location and pertinent capabilities of each.

The NDSU Multimedia Center is located in the IACC Building at NDSU and is equipped with several modern workstations. The Center licences a wide variety of multi-media development software. The IACC building is also home to a set of student clusters where upwards of 200 workstations are available 24 hours a day.

OTHER RESOURCES: Provide any information describing the other resources available for the project. Identify support services such as consultant, secretarial, machine shop, and electronics shop, and the extent to to which they will be available for the project. Include an explanation of any consortium/contractual arrangements with other organizations.

CERTIFICATION PAGE

Certification for Principal Investigators and Co-Principal Investigators:

I certify to the best of my knowledge that:

(1) the statements herein (excluding scientific hypotheses and scientific opinions) are true and complete, and
(2) the text and graphics herein as well as any accompanying publications or other documents, unless otherwise indicated, are the original work of the signatories or individuals working under their supervision. I agree to accept responsibility for the scientific conduct of the project and to provide the required progress reports if an award is made as a result of this application.

I understand that the willful provision of false information or concealing a material fact in this proposal or any other communication submitted to NSF is a criminal offense (U.S.Code, Title 18, Section 1001).

Name (Typed)	Signature	Social Security No.*	Date
PI/PD Brian M Slator	Brian M. Slator	SSNs are confidential and are not displayed *ON FASTLANE SUBMISSIONS	2/10/00
Co-PI/PD			
Co-PI/PD			
Co-PI/PD			
Co-PI/PD			

Certification for Authorized Organizational Representative or Individual Applicant:

By signing and submitting this proposal, the individual applicant or the authorized official of the applicant institution is: (1) certifying that statements made herein are true and complete to the best of his/her knowledge; and (2) agreeing to accept the obligation to comply with NSF award terms and conditions if an award is made as a result of this application. Further, the applicant is hereby providing certifications regarding Federal debt status, debarment and suspension, drug-free workplace, and lobbying activities (see below), as set forth in Grant Proposal Guide (GPG), NSF 00-2. Willful provision of false information in this application and its supporting documents or in reports required under an ensuing award is a criminal offense (U. S. Code, Title 18, Section 1001).

In addition, if the applicant institution employs more than fifty persons, the authorized official of the applicant institution is certifying that the institution has implemented a written and enforced conflict of interest policy that is consistent with the provisions of Grant Policy Manual Section 510; that to the best of his/her knowledge, all financial disclosures required by that conflict of interest policy have been made; and that all identified conflicts of interest will have been satisfactorily managed, reduced or eliminated prior to the institution's expenditure of any funds under the award, in accordance with the institution's conflict of interest policy. Conflict which cannot be satisfactorily managed, reduced or eliminated must be disclosed to NSF.

Debt and Debarment Certifications

(If answer "yes" to either, please provide explanation.)

Is the organization delinquent on any Federal debt?	Yes ☐	No ☒
Is the organization or its principals presently debarred, suspended, proposed for debarment, declared ineligible, or voluntarily excluded from covered transactions by any Federal department or agency?	Yes ☐	No ☒

Certification Regarding Lobbying

This certification is required for an award of a Federal contract, grant, or cooperative agreement exceeding $100,000 and for an award of a Federal loan or a commitment providing for the United States to insure or guarantee a loan exceeding $150,000.

Certification for Contracts, Grants, Loans and Cooperative Agreements

The undersigned certifies, to the best of his or her knowledge and belief, that:

(1) No federal appropriated funds have been paid or will be paid, by or on behalf of the undersigned, to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with the awarding of any federal contract, the making of any Federal grant, the making of any Federal loan, the entering into of any cooperative agreement, and the extension, continuation, renewal, amendment, or modification of any Federal contract, grant, loan, or cooperative agreement.

(2) If any funds other than Federal appropriated funds have been paid or will be paid to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with this Federal contract, grant, loan, or cooperative agreement, the undersigned shall complete and submit Standard Form-LLL, "Disclosure Form to Report Lobbying," in accordance with its instructions.

(3) The undersigned shall require that the language of this certification be included in the award documents for all subawards at all tiers including subcontracts, subgrants, and contracts under grants, loans, and cooperative agreements and that all subrecipients shall certify and disclose accordingly.

This certification is a material representation of fact upon which reliance was placed when this transaction was made or entered into. Submission of this certification is a prerequisite for making or entering into this transaction imposed by section 1352, title 31, U.S. Code. Any person who fails to file the required certification shall be subject to a civil penalty of not less than $10,000 and not more than $100,000 for each such failure.

AUTHORIZED ORGANIZATIONAL REPRESENTATIVE	SIGNATURE	DATE
NAME/TITLE (TYPED) Dr. Edna Holm, Interim Dean of Research	Edna T. Holm	02/10/00

TELEPHONE NUMBER	ELECTRONIC MAIL ADDRESS	FAX NUMBER
701-231-7131	ndsu-research@plains.nodak.edu	701-231-1013

*SUBMISSION OF SOCIAL SECURITY NUMBERS IS VOLUNTARY AND WILL NOT AFFECT THE ORGANIZATION'S ELIGIBILITY FOR AN AWARD. HOWEVER, THEY ARE AN INTEGRAL PART OF THE INFORMATION SYSTEM AND ASSIST IN PROCESSING THE PROPOSAL. SSN SOLICITED UNDER NSF ACT OF 1950, AS AMENDED.