

Senior Project Proposals

Zhizhang Shen *
Dept. of Computer Science and Technology
Plymouth State University

June 9, 2006

1. *The following projects are relatively focused on programming.*

- **An improved *SUDOKU* solver**

SUDOKU has become a popular game. If you do a Google search, you can find about 20,000,000 pages devoted to this game. If you walk into the Border book store in Concord, you will immediately see a book stand with a bunch of SUDOKU books.

Essentially an extension of the *eight queens problem*, a SUDOKU puzzle typically consists of a nine by nine grid, and its unique solution can be summarized as follows [1]:

“Fill(ing) in the grid so that every row, every column, and every 3x3 box contains the digits 1 through 9. ”

For example, given the following SUDOKU puzzle [2, pp.106],

2	8						7	
				7	1			2
1	9				6		8	
3					7			
9		8				7		5
			6					8
	3		7				5	9
4			2	1				
	7						1	6

Given such a puzzle, some of the digits are easy to fill in. For example, if we designate an index for the rows and columns such that it starts from the left to the right for the columns and from bottom up for the rows, then the cell (3, 3) has to be a ‘1’, since it cannot be put any where in the first two rows as a ‘1’ already exists and it cannot be put in the cell (3, 1), either, as a 1 exists in column 1.

Following similar reasoning, we can easily fill in a ‘2’ in (7, 5), a ‘6’ in (3, 5), and the four missing ‘7’.

*Address correspondence to Dr. Zhizhang Shen, Dept. of Computer Science and Technology, Plymouth State University, Plymouth, NH 03264, USA. email address: zshen@mail.plymouth.edu Home page: oz.plymouth.edu/~zshen.

On the other hand, although we know for sure that a ‘1’ cannot be put either in the first or the second 3x3 boxes in the top three rows since both of them already contain a 1, we cannot determine where, between (9, 7) and (9, 9), we should put down a 1. When this situation happens, and it happens a lot, we might make a choice, and continue; and if it turns out to be a wrong choice, we then backtrack to this place and make a change. But, it turns out that this would lead to quite a chaos, partly because we don’t have a hierarchical memory as where we have gone to after making this choice.

Thus, I personally will not fill in a number unless I can 100% sure about where it should go to, which needs a lot of logic reasoning....

We eventually obtain the following solution:

2	8	4	9	5	3	6	7	1
6	5	3	8	7	1	9	4	2
1	9	7	4	2	6	5	8	3
3	2	6	5	8	7	1	9	4
9	4	8	1	3	2	7	6	5
7	1	5	6	4	9	3	2	8
8	3	1	7	6	4	2	5	9
4	6	9	2	1	5	8	3	7
5	7	2	3	9	8	4	1	6

Since I posted this proposal in Spring 2006, one student has come up with a complete solver which solves any given SUDOKU game, even only one cell is filled, by taking a brute-force depth first search. Another student came up with a reduction technique which greatly reduces the searching space a program has to go through, and came up with a partial solver, which only solves a subset of the games.

There are at least two ways to move forward.

- It is thought that the searching space is so big that a brute-force approach will take a long time to crack a “tough” game. This has proved to be false. Thus, an interesting idea is to add on a DFS layer on the partial solver to see how much time we actually gain by taking this reduction technique.
- The current version of the game is 9×9 . Another direction is certainly to generalize the game to $n^2 \times n^2, n \geq 4$.

When n gets larger, the searching space will go up exponentially. Thus, the aforementioned reduction technique will become significant.

- **A SUDOKU Creator**

Although a program does exist to create all the games, it is still interesting to come up with your own, particularly for the general case.

An obvious way is to randomly put down nine 1’s, nine 2’s, ..., and nine 9’s into the grid; then randomly take some of them off. The issue is how many numbers should you take off, and what numbers to take off, so that the original configuration can be reconstructed.

An interesting finding of the aforementioned complete solver is that even if you fill in just one cell, the whole board is fixed. This explains why all the games we have seen so far

has a unique solution. Another issue is thus if we can come up with a mathematical proof for this result. If this can be done, another issue is that if this stays to be the case for the more general case.

This part might not be as difficult as compared with the solving part in an algorithmic term, but should be (much) more difficult in a conceptual term. It is also open-ended.

The following two projects are so-sponsored with Dept. of Safety, State of New Hampshire. The student(s) who (is) are willing to do any of these two projects are expected to work closely with the staffs over there, while I play an advisory and monitory role on this side.

Below is to convey the intents and descriptions of these projects, but not a complete technical specification and should not be used as such. Further documentation and meetings may be set up to clarify any points described below. If you are interested in any of them, please come to see me for further details.

- **A server administration and monitoring Tool**

A student wishing to pursue this project should have a strong background in both Unix and Windows environments, basic SQL skills, and Java or C# programming knowledge. Familiarity with different Networking protocols and utilities such as: Telnet, SSL, SSH, FTP, and ping would be a great help.

The goal of this tool is to be able to monitor both Windows and Unix servers running a variety of programs and to notify the appropriate person in the event of a failure. Among others, this tool should be able to do basic checks such as a ping of a machine to make sure the physical box is up, as well as more specific tests which would send over a simple Oracle SQL statement to test if the database is running correctly.

We would like the tool broken down into two applications. The first would be an administration tool. The administration tool would allow a technically advanced user to setup (provide a login and password) and view all servers in a graphic manner. The administration tool must also be able to define an arbitrary amount of test cases for each server, and be able to configure them.

Once the administrator has created the test plan, the next tool would come into play. This tool would run in the background on a non-technical user's computer. This tool will notify such a user in the event any test fails, when this notifying component should not give any technical information immediately but should give contact information based on which piece of the test plan failed. Once the technical person involved is on the phone, the non-technical user should be able to view the exact error message received, and provide such a message to the responsible individual. Moreover, this notifying application must have some way of getting any updates to the test plan within a 24 hour period.

The entire application must not require any installation on the machines that it is monitoring. You may assume that both the administrator and the non-technical user component are running on Windows 2000 or Windows XP boxes, with the latest JDK and .NET Framework installed. The finished product should be able to start automatically on the non-technical user's machine. The administration application can start via command-line, or executable (.bat, .exe, or .jar). An installation file is not required, but detailed instructions for installation are. If an installation file is provided, all that is needed is

the filename and location.

Choice of language is up to the student, although an Object Oriented solution is recommended as then the test case architecture is pretty straightforward, and the product would be easily extended. So far, in prototyping this project, such languages as Java, C#, and Python have all been tried. All three solutions have drawbacks and benefits. Java does not support the ping utility and the student should not be responsible for writing one. Instead the student must find another way to tell if the physical machine is up and running. Many open source options exist for using the telnet, SSH, and FTP protocols, and those solutions are encouraged. .NET offers many networking solutions pre-packaged, and python does as well.

The State is not necessarily expecting in a working solution, or even a completed project, as this is a lot of work for one semester. What they are looking for is something that can be easily maintained and extended. They are also looking for good design patterns and choices, as well as reuse of code. Some high level UML diagrams can be provided if the student wishes. They would like at least 10 testing classes created, which should cover as broad a spectrum as possible.

Finally, they would also like the ability to test the following specific technologies: Oracle Database, MySQL Database, MS SQL Server Database, Unix, Windows, and any standard FTP server.

- **J2EE Application Programming**

A student wishing to pursue this project should have a strong background in Java, as well as a good working knowledge of HTML and XML. Any knowledge of JSP, Javascript, or Server administration would be very helpful.

The purpose of this project is to configure an open-source or free J2EE web server, create a small sample Java application, and get it to work with struts (the open source solution from sun).

The application content is up to the student but should be similar in difficulty to an on-line pet store or library. The data source of the application is also up to the student, either XML or a database of their choice.

The Java part of this project should be easily separated into three tiers: *the Data tier*, *the Business tier*, and *the Presentation tier*. The business tier should contain all business logic, and perform all validations. The Data tier should have both data source editing capability, and data encapsulation components. The presentation tier should be strictly presenting data to the end users.

Choosing an application server solution should be done before coding begins, as certain application servers have tools that make building applications very easy. One example would be BEA's Weblogic, which comes with Workbench, a Graphical Application building environment. JBoss has similar tools available. There are other application servers out there with similar solutions as well.

The goal for this project is getting the whole system to work, and to have good design practices as well. In a well constructed system, you should be able to have all three tiers on different physical machines, although this is not a requirement of this project. We'd like to see some use of EJB (Enterprise Java Bean)s, and can provide samples of struts entries as well as EJB implementations.

This project will be difficult. The thing that makes it difficult is that all these technologies working together form a very complex system, although none of the single technology used here is difficult and all of the pieces themselves are simple to program. The key in this project will be in thorough researching, in fine tuning all of the pieces of the system, and getting them to work together. A list of web sites, books, and other resources can be provided to an interested student.

2. *For those research oriented:*

- **On numerical processing**

I have published some papers on applied mathematics, particularly network performance analysis. Lots of simulation can be done as follow-up pieces to some of the analytical results.

As an example, in a recent paper, “A Generalized Broadcasting Schema for the Mesh Structures”, I analyzed a generalized broadcasting pattern, consisting of two simple patterns. The results show that we can construct such a general pattern depending on the particular applications, either computation intensive or communication intensive, to achieve the relevant goal(s) of minimizing resource demanding or maximizing load balancing. It is really nice to come up with a data simulation model to test the correctness of the theoretical analysis. For this specific topic, a student is expected to have a basic understanding of computer network communication, which may be acquired by taking such course(s) as CS3400, and have some strong skills in object oriented programming, since all the nodes in such a network can be treated as uniform objects.

For other potential topics that I have certain knowledge, you might want to get to the “My publication” section of my home page to have a look. If you are interested in any of them, you are welcome to talk to me to explore relevant research topics.

- ***Anything else?***

If you happen to bump into any discrete mathematics related issues such as “how many ways are there to...”, or if you are interested in any database application related issues, or if you need a general guidelines on any other issues, please feel free to contact me with the information as provided within this document.

3. *For those who are thorough and good at technical writing.*

- **An experiment platform for computation theory**

For those who has taken the computation theory course, it is abundantly clear that is rather a dry course, involved with lots of mathematical notation and thinking. It will be really nice to have “labs” so that we can experiment with it.

There does exist such a lab component, but the accompanying manual is not very easy to read. Some one might want to dig in and give a detailed description of each and every feature as contained in this collection of simulators together with examples.

The student is expected to have a basic understanding of the relevant mathematical material.

References

[1] www.sudoku.com

[2] *Super SUDOKU*, Modern Publishing, New York, NY 2005.