# Experiment Design for AdvGeo Project

Noboru Matsuda
University of Pittsburgh
May 21, 2003

## 1. Background: Problem and Suppositions

Assume that the students can "understand" domain principles hence the instructional objectives are to make them enable to use domain principles in an actual problem-solving situation. We hope the students to achieve a mastery level.

Given that the students have no difficulty to "understand" the concept around the domain principles, what they need to be mastery is to gain a good activation of those principles in an actual problem-solving situation.

We claim that all what the students need is to get exposed to what they are supposed to do. We conjecture that learning is a function of exposure to the application of domain principles.

The AdvGeo tutor we would build for this study articulates everything that students need to know. The tutor breaks theorem proving steps into substeps that are normally not taught explicitly in classroom instruction. The tutor starts with providing strongest scaffolding on all those steps that basically corresponds with providing bottom-out hints in the model tracing tutors. This kind of scaffolding is called *spontaneous scaffolding* in this study.

Spontaneous scaffolding must fade along with learning. Based on the student performance on the target problem-solving step, the tutor gradually declines the degree of scaffolding (or the amount of information explicitly provided). The fading strategy must fit the hierarchical structure of the model of theorem proving.

## 2. Research Questions

Is spontaneous scaffolding better than model tracing tutor?

Would the students *really* understand the target problem-solving skills and concepts?

## 3. The Control Conditions

If the spontaneous tutor works, then it is probably due to …

- whole thorough scaffolding
- scaffolding on a particular step especially for operationalization
- reification on subgoaling (the goal stack)
- reification on rule application (the proof tree)
- …

Our main concern is on the thorough scaffolding, hence we design control conditions that have less (or too much) scaffolding.

- A tutor that does not fade
- A tutor that does not provide scaffolding at all, but does provide the same GUI learning environment with the goal stack and the proof tree.

Question: How should we handle false, illogical, and/or redundant input?