Empirical Evidence for the Existence and Uses of Metacognition in Computer Science Problem Solving

J. Parham\textsuperscript{1}, L. Gugerty\textsuperscript{2}, & D.E. Stevenson\textsuperscript{1}

Clemson University
\textsuperscript{1}School of Computing
\textsuperscript{2}School of Psychology

March 2010
Outline

Background
  Definitions
  Prior Work

Method
  Environment
  Data Collected

Observations
  Taxonomy
  Write Now, Plan Never
Outline

**Background**
Definitions
Prior Work

**Method**
Environment
Data Collected

**Observations**
Taxonomy
Write Now, Plan Never
Schemata (noun, plural of schema): A collection of mental structures.
Schemata (noun, plural of schema): A collection of mental structures.

- Used to organize current knowledge and provide a framework for future understanding.
Schemata (noun, plural of schema): A collection of mental structures.

- Used to organize current knowledge and provide a framework for future understanding.
- Influence attention.
Schemata (noun, plural of schema): A collection of mental structures.

- Used to organize current knowledge and provide a framework for future understanding.
- Influence attention.
- One can quickly classify new perceptions into schemata and act without effort.
Metacognition

Metacognition (noun): Cognizant of one’s cognition.
Metacognition

Metacognition (noun): Cognizant of one’s cognition.
Competing Definitions:

➤ The control one has over their own cognition and learning.
Metacognition

Metacognition (noun): Cognizant of one’s cognition. Competing Definitions:

- The control one has over their own cognition and learning.
- The processes that allow one to observe, reflect on, and to experience one’s own cognitive processes.
Metacognition (noun): Cognizant of one’s cognition. Competing Definitions:

▶ The control one has over their own cognition and learning.
▶ The processes that allow one to observe, reflect on, and to experience one’s own cognitive processes.

Unified Definition:

▶ Executive processes that are used to:
  ▶ direct and monitor cognition,
  ▶ monitor and evaluate what is being done, and
  ▶ interact with performance and learning components while learning a new task.
Outline

Background
  Definitions
  Prior Work

Method
  Environment
  Data Collected

Observations
  Taxonomy
  Write Now, Plan Never
Solving problems in Computer Science is a dynamic process.
Parham et. al., 2009

- Solving problems in Computer Science is a dynamic process.
- There are significant differences in the cognitive processes of each student.
Solving problems in Computer Science is a dynamic process.

There are significant differences in the cognitive processes of each student.

A domain-specific vocabulary needs to be developed to determine specific cognitive processes influencing problem solving in CS.
Moores et al. 2006

- Measured the relationship between metacognition and student test performance.
Moores et al. 2006

- Measured the relationship between metacognition and student test performance.
- Found metacognition to be related to procedural knowledge.
Moores et al. 2006

- Measured the relationship between metacognition and student test performance.
- Found metacognition to be related to procedural knowledge.
- This relationship is based on problem complexity.
Moores et al. 2006

- Measured the relationship between metacognition and student test performance.
- Found metacognition to be related to procedural knowledge.
- This relationship is based on problem complexity.
- Future studies need to clarify this relationship.
Outline

Background
  Definitions
  Prior Work

Method
  Environment
  Data Collected

Observations
  Taxonomy
  Write Now, Plan Never
Participants

Students enrolled in either:

- Computer Science II
- Data Structures and Algorithms
Participants

Students enrolled in either:

- Computer Science II
- Data Structures and Algorithms
- Volunteer-basis
Participants

Students enrolled in either:

- Computer Science II
- Data Structures and Algorithms
- Volunteer-basis
- One hour

$n = 11$
Participants

Students enrolled in either:

- Computer Science II
- Data Structures and Algorithms
- Volunteer-basis
- One hour
- $10

$n = 11$
Participants

Students enrolled in either:

- Computer Science II
- Data Structures and Algorithms
- Volunteer-basis
- One hour
- $10
- $n = 11$
The Room

Participants were given access to:

- A computer with:
  - internet access,
  - old programs, and
  - the ability to write new programs
- Blank paper, pencils, pens
- Textbooks on Java, C, C++
- A paper copy of the problem
Participants were given access to:

- A computer with:
  - internet access,
  - old programs, and
  - the ability to write new programs

- Blank paper, pencils, pens
The Room

Participants were given access to:

- A computer with:
  - internet access,
  - old programs, and
  - the ability to write new programs
- Blank paper, pencils, pens
- Textbooks on Java, C, C++
The Room

Participants were given access to:

- A computer with:
  - internet access,
  - old programs, and
  - the ability to write new programs
- Blank paper, pencils, pens
- Textbooks on Java, C, C++
- A paper copy of the problem
Outline

Background
Definitions
Prior Work

Method
Environment
Data Collected

Observations
Taxonomy
Write Now, Plan Never
Confounding Variables

Accounted for:
- Years of professional experience vs.
- Years university experience
Confounding Variables

Accounted for:
- Years of professional experience vs.
- Years university experience

Unexplored:
- Caffeine
- Sleep deprivation
- Empty stomach
- Stress
The study took place in a room with two subjects:

- The student subject
- An interviewer
The study took place in a room with two subjects:

- The student subject
- An interviewer

Verbalizations were recorded of the participants:
The study took place in a room with two subjects:
- The student subject
- An interviewer

Verbalizations were recorded of the participants:
- As they thought aloud while solving the problem,
"Talk-Aloud Protocol".

The study took place in a room with two subjects:

- The student subject
- An interviewer

Verbalizations were recorded of the participants:

- As they thought aloud while solving the problem,
- When they asked the interviewer for information,
"Talk-Aloud Protocol".

The study took place in a room with two subjects:
- The student subject
- An interviewer

Verbalizations were recorded of the participants:
- As they thought aloud while solving the problem,
- When they asked the interviewer for information,
- When prompted by the interviewer when silent for more than 10 seconds.
The study took place in a room with two subjects:
- The student subject
- An interviewer

Verbalizations were recorded of the participants:
- As they thought aloud while solving the problem,
- When they asked the interviewer for information,
- When prompted by the interviewer when silent for more than 10 seconds.

An interview was conducted after the student solved the problem.
Outline

Background
Definitions
Prior Work

Method
Environment
Data Collected

Observations
Taxonomy
Write Now, Plan Never
Metacognitive and Schematic Processes

Metacongitive Processes
Metacognitive and Schematic Processes

Metacongitive Processes

- Start, Revisit Goals
Metacognitive and Schematic Processes

Metacongitive Processes

- Start, Revisit Goals
- Understand Problem, Plan
Metacognitive and Schematic Processes

Metacongitative Processes

- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
Metacognitive and Schematic Processes

Metacongitive Processes
- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
- Verbalize Low Prior Knowledge
Metacognitive and Schematic Processes

Metacongitive Processes

➤ Start, Revisit Goals
➤ Understand Problem, Plan
➤ Read, Consider Design
➤ Verbalize Low Prior Knowledge
➤ Inspect
Metacognitive and Schematic Processes

Metacongitive Processes

- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
- Verbalize Low Prior Knowledge
- Inspect
- Compare
Metacognitive and Schematic Processes

Metacongitive Processes

- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
- Verbalize Low Prior Knowledge
- Inspect
- Compare

Schemata
Metacognitive and Schematic Processes

Metacongitive Processes

- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
- Verbalize Low Prior Knowledge
- Inspect
- Compare

Schemata

- Design
Metacognitive and Schematic Processes

Metacongitive Processes

- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
- Verbalize Low Prior Knowledge
- Inspect
- Compare

Schemata

- Design
- Write Code
Metacognitive and Schematic Processes

Metacongitive Processes

- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
- Verbalize Low Prior Knowledge
- Inspect
- Compare

Schemata

- Design
- Write Code
- Compile
Metacognitive and Schematic Processes

Metacongitive Processes
- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
- Verbalize Low Prior Knowledge
- Inspect
- Compare

Schemata
- Design
- Write Code
- Compile
- Execute
Metacognitive and Schematic Processes

Metacongitive Processes

- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
- Verbalize Low Prior Knowledge
- Inspect
- Compare

Schemata

- Design
- Write Code
- Compile
- Execute
- Diagnose
Metacognitive and Schematic Processes

Metacongitive Processes
- Start, Revisit Goals
- Understand Problem, Plan
- Read, Consider Design
- Verbalize Low Prior Knowledge
- Inspect
- Compare

Schemata
- Design
- Write Code
- Compile
- Execute
- Diagnose
- Fix Code
Outline

**Background**
- Definitions
- Prior Work

**Method**
- Environment
- Data Collected

**Observations**
- Taxonomy
  - *Write Now, Plan Never*
Summary of Results

56% of time spent in Schemata.
- 27% in Write Code
- 17% in Inspect
- 13% in Start, Revisit Goals
Summary of Results

56% of time spent in *Schemata*.
  - 27% in *Write Code*

44% of time spent in *Metacongition*.
  - 17% in *Inspect*
  - 13% in *Start, Revisit Goals*
Summary of Interactions
Summary of Interactions

*Inspect/Compare* $\rightarrow$ *Start, Revisit Goals*

$12\%$

*Start, Revisit Goals* $\rightarrow$ *Inspect/Compare*

$12\%$
Summary of Interactions

*Inspect/Compare* $\rightarrow$ *Start, Revisit Goals* $12\%$

*Start, Revisit Goals* $\rightarrow$ *Inspect/Compare* $12\%$

*Start, Revisit Goals* $\rightarrow$ *Compile/Execute* $0\%$

*Compile/Execute* $\rightarrow$ *Start, Revisit Goals* $12\%$

*Inspect/Compare* $\rightarrow$ *Diagnose* $29\%$

*Diagnose* $\rightarrow$ *Start, Revisit Goals* $30\%$

*Fix Code* $\rightarrow$ *Inspect/Compare* $33\%$

*Inspect/Compare* $\rightarrow$ *Compile/Execute* $12\%$
Summary of Interactions

*Inspect/Compare → Start, Revisit Goals*  
12%

*Start, Revisit Goals → Inspect/Compare*  
12%

*Start, Revisit Goals → Compile/Execute*  
0%

*Compile/Execute → Start, Revisit Goals*  
12%

*Write Code → Inspect/Compare*  
33%

*Inspect/Compare/Understand Problem, Plan → Write Code*  
26%
Summary of Interactions

Inspect/Compare \(\rightarrow\) Start, Revisit Goals \(12\%\)
Start, Revisit Goals \(\rightarrow\) Inspect/Compare \(12\%\)
Start, Revisit Goals \(\rightarrow\) Compile/Execute \(0\%\)
Compile/Execute \(\rightarrow\) Start, Revisit Goals \(12\%\)

Write Code \(33\%\) \(\rightarrow\) Inspect/Compare
Inspect/Compare/Understand Problem, Plan \(26\%\) \(\rightarrow\) Write Code

Start, Revisit Goals \(47\%\) \(\rightarrow\) Write Code
Summary of Interactions

- Inspect/Compare $\rightarrow$ Start, Revisit Goals (12%)
- Start, Revisit Goals $\rightarrow$ Inspect/Compare (12%)
- Start, Revisit Goals $\rightarrow$ Compile/Execute (0%)
- Compile/Execute $\rightarrow$ Start, Revisit Goals (12%)
- Write Code $\rightarrow$ Inspect/Compare (33%)
- Inspect/Compare/Understand Problem, Plan $\rightarrow$ Write Code (26%)
- Start, Revisit Goals $\rightarrow$ Write Code (47%)
- Diagnose $\rightarrow$ Start, Revisit Goals (30%)
- Understand Problem, Plan $\rightarrow$ Start, Revisit Goals (30%)
Summary of Interactions

*Inspect/Compare* $12\% \rightarrow$ Start, Revisit Goals

Start, Revisit Goals $12\% \rightarrow$ Inspect/Compare

Start, Revisit Goals $0\% \rightarrow$ Compile/Execute

Compile/Execute $12\% \rightarrow$ Start, Revisit Goals

Write Code $33\% \rightarrow$ Inspect/Compare

Inspect/Compare/Understand Problem, Plan $26\% \rightarrow$ Write Code

Start, Revisit Goals $47\% \rightarrow$ Write Code

Diagnose $30\% \rightarrow$ Start, Revisit Goals

Understand Problem, Plan $30\% \rightarrow$ Start, Revisit Goals

Fix Code $33\% \rightarrow$ Inspect/Compare

Fix Code $30\% \rightarrow$ Compile/Execute

Inspect/Compare $29\% \rightarrow$ Diagnose
Overview of Interactions

All interactions occurring with frequency greater than 3% are shown with an arrow:

- Design
- Write Code
- Compile/Execute
- Diagnose
- Fix Code

Metacognition
Summary

- Metacognition: significant role in CS problem solving
Summary

- Metacognition: significant role in CS problem solving
- Often (51%) code was inspected and fixed before checking for compile/execute errors.
Summary

- Metacognition: significant role in CS problem solving
- Often (51%) code was inspected and fixed before checking for compile/execute errors.
- Often (30%) diagnosed errors were outside the current goal, requiring a switch to a previous goal before continuing.
References I

- Wikipedia. Schema (Psychology)  

- Wikipedia. Metacognition  
  http://en.wikipedia.org/wiki/Metacognition

- Parham, J.; Chinn, D.; and Stevenson, D.E.  
  Using Bloom’s Taxonomy to Code Verbal Protocols of Students Solving a Data Structure Problem.  
  *Proceedings of the 47th Annual Southeast Regional Conference, March 2009*

- Moores, T.T.; Chang, J.C.; and Smith, D.K.  
  Clarifying the Role of self-efficacy and metacognition as predictors of performance: Construct Development and Test.  
  *SIGMIS Database*
Wikipedia.

Think-Aloud Protocol

http://en.wikipedia.org/wiki/Think_aloud_protocol