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

J. Parham[1], L. Gugerty[2], & D.E. Stevenson[1]

Clemson University
[1]School of Computing
[2]School of Psychology

March 2010

Outline

Outline

Schemata

Schemata (noun, plural of schema): A collection of mental structures.

Schemata

Schemata (noun, plural of schema): A collection of mental structures.

- Used to organize current knowledge and provide a framework for future understanding.

Schemata

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

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

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

Parham et. al., 2009

- 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.

Parham et. al., 2009

- Solving problems in Computer Science is a dynamic process.
- There are significant differences in the cognitive processes of each student.
- A domain-specific vocabular needs to be developed to determine specific cognitive processes influencing probelm 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

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

Participants

Students enrolled in either:

- Computer Science II
- Data Structures and Algorithms

- Volunteer-basis
- One hour
- $ 10

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

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

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

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

"Talk-Aloud Protocol".

The study took place in a room with two subjects:

- The student subject
- An interviewer

"Talk-Aloud Protocol".

The study took place in a room with two subjects:

- The student subject
- An interviewer

Verbalizations were recorded of the participants:

"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,

"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.

"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.

An interview was conducted after the student solved the problem.

Outline

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

Metacongitive 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

Summary of Results

56% of time spent in Schemata.

- 27% in *Write Code*

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* $\overset{12\%}{\rightarrow}$ *Start, Revisit Goals*

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

Summary of Interactions

$$\textit{Inspect}/\textit{Compare} \xrightarrow{12\%} \textit{Start, Revisit Goals}$$

$$\textit{Start, Revisit Goals} \xrightarrow{12\%} \textit{Inspect}/\textit{Compare}$$

$$\textit{Start, Revisit Goals} \xrightarrow{0\%} \textit{Compile}/\textit{Execute}$$

$$\textit{Compile}/\textit{Execute} \xrightarrow{12\%} \textit{Start, Revisit Goals}$$

Summary of Interactions

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

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

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

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

Write Code $\overset{33\%}{\rightarrow}$ *Inspect*/*Compare*

Inspect/*Compare*/*Understand Problem, Plan* $\overset{26\%}{\rightarrow}$ *Write Code*

Summary of Interactions

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

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

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

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

Write Code $\overset{33\%}{\rightarrow}$ *Inspect*/*Compare*

Inspect/*Compare*/*Understand Problem, Plan* $\overset{26\%}{\rightarrow}$ *Write Code*

Start, Revisit Goals $\overset{47\%}{\rightarrow}$ *Write Code*

Summary of Interactions

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

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

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

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

Write Code $\overset{33\%}{\rightarrow}$ *Inspect*/*Compare*

Inspect/*Compare*/*Understand Problem, Plan* $\overset{26\%}{\rightarrow}$ *Write Code*

Start, Revisit Goals $\overset{47\%}{\rightarrow}$ *Write Code*

Diagnose $\overset{30\%}{\rightarrow}$ *Start, Revisit Goals*

Understand Problem, Plan $\overset{30\%}{\rightarrow}$ *Start, Revisit Goals*

Summary of Interactions

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

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

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

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

Write Code $\overset{33\%}{\rightarrow}$ *Inspect/Compare*

Inspect/Compare/Understand Problem, Plan $\overset{26\%}{\rightarrow}$ *Write Code*

Start, Revisit Goals $\overset{47\%}{\rightarrow}$ *Write Code*

Diagnose $\overset{30\%}{\rightarrow}$ *Start, Revisit Goals*

Understand Problem, Plan $\overset{30\%}{\rightarrow}$ *Start, Revisit Goals*

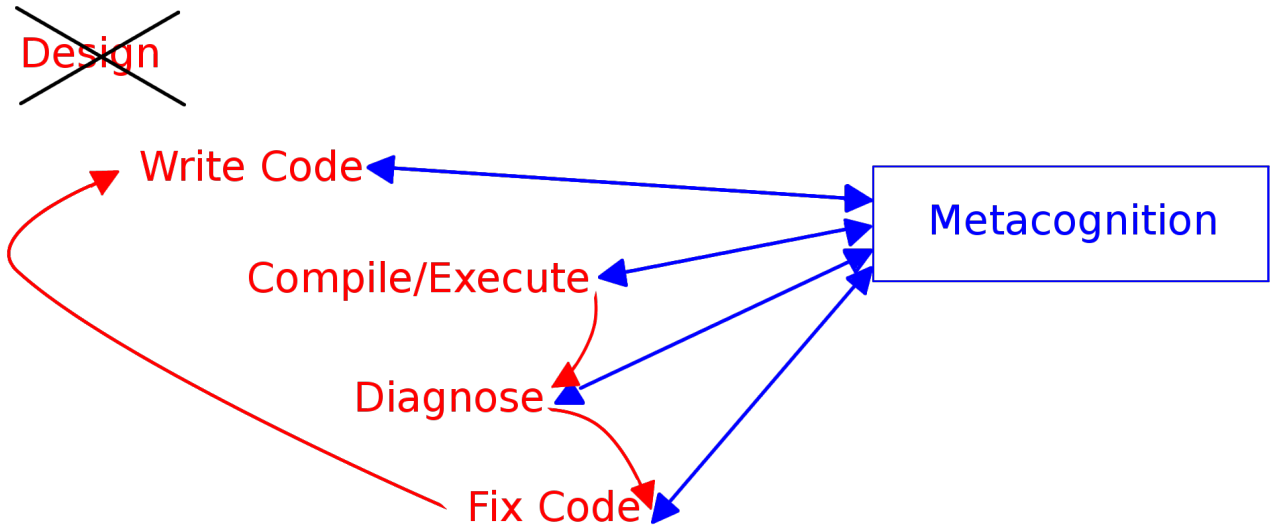
Fix Code $\overset{33\%}{\rightarrow}$ *Inspect/Compare*

Fix Code $\overset{30\%}{\rightarrow}$ *Compile/Execute*

Inspect/Compare $\overset{29\%}{\rightarrow}$ *Diagnose*

Overview of Interactions

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



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

References II


Wikipedia.
Think-Aloud Protocol
http://en.wikipedia.org/wiki/Think_aloud_protocol