Ginger2: An Environment for Computer-Aided Empirical Software Engineering

Koji Torii, Ken-ichi Masumoto, Kumiyo Nakakoji, Yoshihiro Takada, Shingo Takada, Kazuyuki Shima

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING 1999

2006-08-16

Kim, Su-hyun
### Empirical software engineering

#### Introduction (1/2)

**In vivo (naturalistic)**

The study of software related artifacts for the purpose of characterization, understanding, evaluation, prediction, control, management, or improvement through qualitative or quantitative analysis [Harrison and Basili]

- Actual software development project
- Relevance of real, ill-defined problems that emerge in actual S/W development
- Make “knowledge” available and reusable for other projects and members

**In vitro (laboratory)**

A series of actions to obtain knowledge and better understanding about some aspects of software development given a set of problem statements in the form of issues, questions or hypotheses. [Koji Torii et al.]

- Consist of controlled Experiments
- Develop and test hypothesis about particular aspect of human behavior
- Develop generalized model of behavior
- Proposed techniques and methods

*These two approaches complement each other*
Computer-Aided Empirical Software Engineering (CAESE)

- Supporting *in vitro* study
  - Involving in empirical activities
  - Dealing with many types of data

- Feature
  - Empirical study process model
  - Data collection model
  - Empirical study tools
Architecture of GINGER [IST’96]

GINGER automatically collects software process and product data, and performs analysis to provide feedback to developers.
History of GINGER [IST’96] (since 1985)

- **Quality**
  - Evaluation of Quality (automated testing)
  - Estimation residual faults based on change
  - Estimation residual faults based on petrinet model

- **Performance**
  - Programmer Performance Model
  - Evaluation of Team Performance
  - Evaluation of Programmer debugging Performance

- **Software artifacts**
  - Off-line feedback (size, change)
  - On-line feedback (size, change)
  - Evaluation of Data Flow Diagram (from CASE tool)
  - Evaluation of process and product base on change history
Illustration of CAESE with CASE

(Hypothesis, Issues, Questions)

CASE
- Development Tools
  - Development Process Model
  - Product Model

Software Artifacts

Knowledge Concerning Software Development

CAESE
- Empirical Study Tools
  - Empirical Study Process Model
  - Data Collection Model

Software Developer

Customer Requirements

Experimenter

Problem Statements

Software Engineering Lab, KAIST
Empirical study process model

- **Needs Analysis**
  - Identifying problem, specifying goal, formulate hypothesis

- **Experiment Design**
  - Detailed specifications for the experiment. Experimental tools, types of subjects, analysis methods and tools

- **Experimentation**
  - Experiments are monitored and data are collected

- **Data Analysis**
  - Experimenters analyze data collected

- **Packaging**
  - Experimenters compile “knowledge” into a “package” available as an output from the study.
Data collection model

- **Three entities & Five viewpoints**
  - Same collection tool for different aspect of data
  - Data analysis needs to be conducted on the five types of data
Empirical study tool

CAESE Framework (4/4)

- Needs Analysis Tool
- Experiment Design Tool
- Data Collection Tools
- Data Analysis Tools
- Packaging Tools
- Data Integration Tools
- Control Integration Tools
- Empirical Study Management Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Integration Tools</td>
<td>Integrating the various types of data</td>
</tr>
<tr>
<td>Control Integration Tools</td>
<td>Synchronizing the use of different tools</td>
</tr>
<tr>
<td>Empirical Study Management Tools</td>
<td>Managing the empirical study itself</td>
</tr>
<tr>
<td></td>
<td>(Same way as project management tools in CASE)</td>
</tr>
</tbody>
</table>
Considerations

- Developers are engaged in many types of activities simultaneously
- CAESE should support continuous automatic collection of data
  - Fine-grained data
  - The load on the subject is lessened
  - Necessary data will not be left out by mistake
  - The chances of mis-informing would be minimized
## Implementations

### Data Collection Tools

<table>
<thead>
<tr>
<th>Data Collection Tools</th>
<th>Data Type</th>
<th>Data collection model category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>subjects</td>
</tr>
<tr>
<td>Audio &amp; Video Recorder</td>
<td>Video data</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Audio data</td>
<td>X</td>
</tr>
<tr>
<td><strong>Mon, Kterm-mon, Emacs-mon</strong></td>
<td>Key stroke</td>
<td>X</td>
</tr>
<tr>
<td>(window monitoring tools)</td>
<td>Mouse</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Window operations</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Changes in window</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td><em>(displaying on monitor)</em></td>
<td></td>
</tr>
<tr>
<td>Eyetrackd</td>
<td>Eye-trace</td>
<td>X</td>
</tr>
<tr>
<td>3D Motion Measurement System</td>
<td>3D motion data</td>
<td>X</td>
</tr>
<tr>
<td>Skin Resistance level Measurement</td>
<td>Skin resistance level</td>
<td>X</td>
</tr>
<tr>
<td>Ginger1 System</td>
<td>Tool usage log</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Program changes</td>
<td></td>
</tr>
</tbody>
</table>
Data & Control Integration

- System architecture

Data Collection Tools
- Audio & Video Recorder
- Other Data Collection Tools

Data Integration Tools
- AV-Monitor
- Monitord
- AV-Database
- Database

MON language

Data Analysis Tools
- Audio & Video Player
- Other Data Analysis Tools

Control Integration Tools
- Data Recorder
- Data Player

Software Engineering Lab, KAIST
Two analysis activities

- Observational analysis (Replaying)
  - During and after experiments
  - Data display and visualization tool
- Computational analysis
  - Heavily depends on computational means
Data Analysis tools

- Experimenter may make while observing and experiment
- Data Screen, 3D Motion, and Event
- Measuring “Stress”
- Keystroke and Gaze point

- Statistical Analysis
- Annotator Tool
- Video Search Tool
- SRL Display
- Movement Similarity Analyzer
- Multidata Display
- Audio & Video Player
- GINGER2

Software Engineering Lab, KAIST
Case Study (1/3)

Modeling Debugging Process

- **Need Analysis**
  - Experts vs. novice users

- **Experiment Design**
  - Identifying which of the 15 modules contained a bug

- **Collecting Data**
  - Eyetrackd
  - Audio & Video Recorder: (To record interviews)

- **Data Analysis**
  - Analyzed how often gaze points shifted among modules
  - How long the gaze point stayed within a single module
  - Visualized the trace of gaze points
Modeling Debugging Process (cont’d)

- Analysis results
  - Difference in the frequency of shifts of gaze points
    - Experts gradually focused on one or two modules
    - Novice subjects kept shifting their gaze points
Two-person Debugging process

**Experiment Design**
- Two subjects find single bug

**Collecting Data**
- Kterm-mon
- Audio & Video Recorder: (communications)

**Need Analysis**
- How effective multiple programmer can work together in debugging

**Analysis results**
Debugging was more effective
- with a “shared buffer”:
  - communicated asynchronously
- separating role:
  - understanding the program / identifying the location of bug

**Data Analysis**
- How knowledge is communicated between the subjects
- How strict roles between the two subjects
Computer-Aided Empirical Software Engineering

- Benefits
  - Integration
    - Synchronization of simultaneously occurring data
    - Analyzing data from multiple perspectives
  - Sharing experiment designs
    - Allowing to develop “knowledge base”
      (Experiment design requires lots of experties and experience)
  - Extensibility
    - Reducing the cost of introducing new data collection equipments and new analysis methods and tools
      (Just make an addition to MON language)
Future Work

- Support for Needs Analysis
  - Applying GQM [Basili, TSE’88]

- Support for Experiment Design
  - After accumulating “know-how”
    - Focus on analysis by simulation

- Support for Packaging
  - Developing representation standard
  - Publishing data on the web

- Distributed CAESE
  - Extending transmission from LAN to Internet
Critiques
- Weak support for computational analysis

Lessons learned
- Data and control integration mechanism
  - MON Language
  - Synchronized collection of simultaneously occurring data
My approach
Thank you!!!
Data Collection of GINGER2

Audio & Video Recorder

Mon, Kterm-mon, Emacs-mon (window monitoring tools)

SRL Measurement System

Eye Tracking System

Colored Balls for 3D Motion Measurement System

Special Eyeglasses for Eye Tracking System
## MON language

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h hawk</td>
<td># The following event occurred on a computer called hawk.</td>
</tr>
<tr>
<td>:739713873</td>
<td># The following event occurred at 739713873 seconds.</td>
</tr>
<tr>
<td>w 0x380000c</td>
<td># The following event occurred on a window called 0x380000c.</td>
</tr>
<tr>
<td>Wo 400x200+100+100</td>
<td># A window was opened.</td>
</tr>
<tr>
<td>Ts 20 14</td>
<td># The height of a text line is 20 dots and the character width 14 dots.</td>
</tr>
<tr>
<td>T hawk%_ 0 0</td>
<td># The prompt “hawk%” was displayed.</td>
</tr>
<tr>
<td>C 6 0</td>
<td># A text cursor appeared.</td>
</tr>
<tr>
<td>:1.5</td>
<td># 1.5 seconds has passed.</td>
</tr>
<tr>
<td>K cd_/etc\r</td>
<td># The command “cd /etc” was input.</td>
</tr>
<tr>
<td>T cd_/etc</td>
<td># “cd /etc” was echoed at the location of the cursor.</td>
</tr>
<tr>
<td>T hawk%_ 0 1</td>
<td># The next prompt is displayed.</td>
</tr>
<tr>
<td>C 6 1</td>
<td># The cursor has moved.</td>
</tr>
<tr>
<td>:3</td>
<td># 3 seconds has passed.</td>
</tr>
</tbody>
</table>
Data Displays

Data Screen Display

3D Motion Display

Event Display
Multidata Display
Movement Similarity Analyzer

Base movement

Distance of Vectors vs. Time
Skin Resister Level Display

[Diagram showing changes and stress levels over time with annotations for DO, D1, T1, D2, T2, More stress, Low Load Level, High Load Level, and SRL Value]
Skin Resister Level Display

More stress