CHALLENGES OF DISTANT LEARNING IMPLEMENTATION OF A PROGRAMMING CLASS

IARIA SOFTNET 2020 KEYNOTE

Evgeny Pyshkin
University of Aizu
Abstract

In the situation of societal lockdowns of 2020, educational institutions faced new challenges in organizing teaching and learning processes so that to adopt them to extensive use of remote teaching models. Currently we observe an increased academic discourse on efficient distant learning approaches. Though various practical solutions, successful practices, and supporting computer technology seem to be already in place, many additional efforts are required from academicians and tutors to resolve significant technical, managerial, methodological, and psychological issues of distant learning.

In this talk, we discuss our approach to programming class organization and workflow with a particular focus on its adoption to current situation requiring extensive (and even exclusive) use of distant learning tools, which have both great advantages and considerable limitations. We share a number of methodological and organizational solutions that may be used to improve software development instruction, where the suggested methods are not only focused on remote teaching tools, but may serve traditional face-to-face classes as well.

We particularly address such aspects as forms of teacher/learner collaboration and the project roles that teachers and students can perform during class sessions, interactivity issues, incremental design applied to the classroom needs, importance of visual models, integration of academic workflow with software testing, project management, code review and source control tools.
Speaker

Senior Associate Professor, Ph.D., Doctoral Maru-Go University of Aizu

Career: Software Engineer; Assistant Professor; Associate Professor; Senior Associate Professor Peter the Great St. Petersburg Polytechnic University

- Aizu-Wakamatsu – city in Tohoku Region, Japan
  - Population around 150000
  - Rice growing and sake production
  - Samurai city
  - University of Aizu – international university (40% of staff are foreigners)
  - University focused on computer technology and its applications
University of Aizu: To Advanced Knowledge for Humanity

- Since 1993: First contemporary university in the region
- International Outlook 70.6 (1st in Japan)
- Citations 58.6 (8th in Japan)
- The only university in Japan offering bilingual programs in the undergraduate school.
- Graduate school is completely in English

https://u-aizu.ac.jp/
### University of Aizu 会津大学

#### Japan National Ranking by Nikkei

<table>
<thead>
<tr>
<th>11 ~ 25位</th>
<th>学生数約1000人の会津大学が北大に迫る</th>
</tr>
</thead>
<tbody>
<tr>
<td>北海道国 ホイド大学</td>
<td>23位</td>
</tr>
<tr>
<td>福島県会津大学</td>
<td>199位</td>
</tr>
<tr>
<td>広島県広島大学</td>
<td>30位</td>
</tr>
<tr>
<td>兵庫県神戸大学</td>
<td>12位</td>
</tr>
<tr>
<td>千葉県千葉大学</td>
<td>21位</td>
</tr>
<tr>
<td>長野県信州大学</td>
<td>46位</td>
</tr>
<tr>
<td>鳥取県岡山大学</td>
<td>28位</td>
</tr>
<tr>
<td>片山県立教大学</td>
<td>93位</td>
</tr>
<tr>
<td>新潟県新潟大学</td>
<td>58位</td>
</tr>
<tr>
<td>愛知県慶應義塾大学</td>
<td>3位</td>
</tr>
<tr>
<td>大阪府立命館アジア太平洋大学</td>
<td>160位</td>
</tr>
<tr>
<td>福岡県九州大学</td>
<td>36位</td>
</tr>
<tr>
<td>東京都金沢大学</td>
<td>47位</td>
</tr>
<tr>
<td>長崎県長崎大学</td>
<td>47位</td>
</tr>
<tr>
<td>大阪府立教大</td>
<td>5位</td>
</tr>
<tr>
<td>福岡県熊本大学</td>
<td>49位</td>
</tr>
</tbody>
</table>

© 2019 Center for Globalization, the University of Aizu
Charming Aizu
Speaker’s Professional Background

- Human-centric computing
  - Rapid transformation from HCI to its own distinctive research agenda
  - Multiple links to digital transformation concepts
  - Affecting society and individuals
  - Inter- and even transdisciplinary nature of HCC
  - Towards more personalization and user collaboration

- Software Engineering
  - Methodology for software development education
  - Software testing
  - Software for computer-assisted language teaching

- Cross-cultural communication, technology disciplines in scope of humane sciences
Focus of this Talk

- Review of earlier works on programming teaching
  - Multi-aspect tasks in software education
  - Interdisciplinary aspects of software engineering
  - From engineering to liberal arts

- Programming class workflow
  - Why lecture and exercises are not enough
  - Network of connected activities and links to distant learning

- Affordances and constraints of distant learning

- Teaching and learning practices
  - Developing students’ abilities to think effectively
  - Developing practical and soft skills

Considering software disciplines within the context of liberal arts is connected to significant changes in the learning models.

We anticipate more than only professional developers’ skills from our students:
- They have to be able to work in a collaborative environment.
- Significance of organizational learning models favoring public display, teamwork and professional discussion significantly increases.

It is extremely important to find ways to create a collaboration environment where students can actively participate in the co-learning process together with their more experiences colleagues.

“Computer science draws upon perspectives from many disciplines and has a symbiotic relationship with the liberal arts disciplines, so it might be considered the ultimate of them” **

---

Bridge a Methodology Gap in Software Education

Attention to important particularities of software development process with respect to a software development course

- **Software changeability**: Much different from products of engineering

- **Software as a community product**: Contributing to open-source solutions requires specific skills and abilities

- **Many interdisciplinary activities**: Students have to get programming skills, but also to learn how to communicate with stakeholders, and how to cooperate in multidisciplinary teams

- **Programming is close to language study**: A software problem may have a variety of acceptable solutions
Use Case: Learning Activities in a Programming Course

- Practices and lessons learned after teaching the connected undergraduate courses “Introduction to Programming” and “C Programming” in the University of Aizu (128 class hours in total)

- Diversity of activity forms
  - How computer science can learn from teaching forms and practices which exists in fine arts
Refining Our Approach to Programming Class Organization and Workflow


Roles that Teachers and Students Perform


Use Case: Hands-On Session Example

Exercise: Let’s Extend Our Model

- Now our language supports three types of tokens:
  - Identifiers
  - Assignment =
  - End-of-expression ;
  - Binary values 10001101
- Input file contains expressions like the following:
  - value = 10001101;
- The problem:
  - Revise the state diagrams
  - Implement the revised finite state machines
    - Lexer recognizes tokens
    - Parser follows syntax rules
  - In process of program execution all the binary values should be printed as hexadecimal values
Bridging to More Complex Concepts:
Multi-Aspect Tasks *

Example:
The syntax of a parenthesis-free expression may be defined by a context-free grammar $E = (V_N, V_T, P, S)$ where $V_N$ is a finite set of nonterminal symbols, $V_T$ is a finite set of terminal symbols, $S$ is a start symbol, and $P$ is a finite set of the grammar production rules.

The first group of rules is to be used during the syntactic analysis stage

$$S ::= <expression> ";"$$
$$<expression> ::= <item>$$
$$<expression> ::= <item>{"+"|"-"}<expression>$$
$$<item> ::= <factor>$$
$$<factor> ::= <factor>{"*"|"/"}<item>$$
$$<factor> ::= <identifier>$$
$$<factor> ::= <dec-const>$$
$$<factor> ::= "("<expression>")"$$

The second group is to be used by a lexer:

$$<identifier> ::= <letter>[<letter>|<dec-digit>]...$$
$$<dec-const> ::= [<dec-digit>]...$$
$$<letter> ::= {'_'|'A'..'Z'|'a'..'z'}...$$
$$<dec-digit> ::= {'0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'}...$$

Visualizing the solution in the form of L-Net suggested by M. Lekarev**

# Case Study: Parenthesis-free Expression Parsing as a Multi-Aspect Task

<table>
<thead>
<tr>
<th>Solution stage</th>
<th>Underlying models</th>
<th>Programming language focus</th>
<th>Visual models</th>
<th>In-depth focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement definition</td>
<td>Recursive grammar</td>
<td>Unit tests planning</td>
<td>Syntactic diagrams</td>
<td>Left- and right- recursive grammars, grammar transformation</td>
</tr>
<tr>
<td>Symbol syntactic class definition</td>
<td>Sets, bitwise operations</td>
<td>Enumerations, symbol tables</td>
<td>Masked bitwise operations</td>
<td>Syntactic classes recognition in more complex languages</td>
</tr>
<tr>
<td>Lexical analysis</td>
<td>Finite state machines, hash search</td>
<td>Inheritance, dynamic binding, parametric polymorphism</td>
<td>Visual formalisms, state charts, class diagrams</td>
<td>Upcasting, downcasting, run time type information, regular expressions</td>
</tr>
<tr>
<td>Syntactic analysis</td>
<td>Recursive descent parser</td>
<td>Recursive functions</td>
<td>Syntax trees, class diagrams</td>
<td>Type switch, multiple dispatch</td>
</tr>
<tr>
<td>Code generation</td>
<td>Polish notation</td>
<td>Collections</td>
<td></td>
<td>Interfaces of collections</td>
</tr>
<tr>
<td>Computation (execution)</td>
<td>Stack</td>
<td>Stack operations</td>
<td></td>
<td>Stack implementations, transforming recursion to iteration</td>
</tr>
</tbody>
</table>
Code Review Practices in a Programming Class

- “Human-based” quality assurance practice
- Extends learning process and enforcing practicing “soft skills”
- Around 40% of useful review comments are related to evolvability defects

Readability Features: Example

Strong influence on readability
- Naming conventions
- Empty lines
- Number of identifiers and characters per line
- Specific indentation scheme

Moderate predictive readability power
- Long identifier names
  - They are not directly connected to a concept of self-documented code
- Comments
  - While comments can enhance readability, they are typically used in code segments that started out less readable

Incremental Design and Importance of Visual Models

- From the very first steps, it is important to introduce to students an approach to work on their practical assignments incrementally.
- Even classroom demos (should) be discussed in their possible evolvement.
- **Example with Rational fractions (C Programming class):** from imperative constructs to structural types and modularity.
Case Study and Discussion:
Distant Learning Affordances and Constrains

- Pre-recorded videos
- Online meeting tools
- Learning management systems
- Version control systems
- Project management tools
- Code review tools
- Hybrid educational environments specifically focused on programming teaching
- Mind maps
Challenges of Present Day Software in Frame of Education

- **“TRADITIONAL” SOFTWARE**
  - Logical and imperative
  - States and transitions
  - Limited parallelism
  - Data modeling and processing
  - Logic resolution

- **AI BASED SOFTWARE**
  - Associative
  - Neural networks
  - Inherently parallel
  - Machine learning
  - Inference (prediction)
Revisiting Metaphors in (Software) Education

- Importance of good metaphors in educations
  - For mapping domains
  - For reifying abstract ideas
  - For conveying complex concepts
  - For bridging communities

---

Software Metaphors: Examples

- Program objects
  - Scope
  - Assignment
  - Lifetime

- Control structures
  - Decision
  - Loop

- Modular structures
  - Library
  - Package

- Program workflow
  - Thread
  - Lazy computation

- Interface design
  - Menu
  - Palette
  - Files and folders

- Design patterns
  - Factory method
  - Delegation
  - Future

- Software analysis
  - Bad smell
  - Refactoring
  - Code mutant
Application of Metaphors from Fine Arts to Programming Teaching

L’oeil suit les chemins qui lui ont été ménagés dans l’oeuvre

Paul Klee, quoted in George Perec’s La Vie, mode d’emploi (1978)

Unless you write a Chinese character in the right order of brush strokes, it will never look beautiful!
Acknowledgements to Collaborators

Natalia Bogach
Peter the Great St. Petersburg Polytechnic University

Vlatko Davidovski
Cognizant Business Consulting Strategy & Transformations

Vitaly Klyuev
University of Aizu

John Blake
University of Aizu

Maxim Mozgovoy
University of Aizu

Takako Yasuta
Fukushima Medical University

Andrei Kuznetsov
JetBrains
CHALLENGES OF DISTANT LEARNING
IMPLEMENTATION OF A PROGRAMMING CLASS

Evgeny Pyshkin
University of Aizu