Computational Thinking and COOL-Informatics in Primary and Secondary Schools
Andreas Bollin
April 28, 2016, OCG, Wollzeile 1, 1010 Vienna, Austria
Pondering about our Future

• In only 11 out 29 countries people believe that science and technological innovation will have a positive impact on job creation! [European Commission. SPECIAL EUROBAROMETER 419, Public Perceptions of Science, Research and Innovation, page 40, Oct. 2014]
Content

• Part (I) – Motivation
  • Computer Science
  • Cultural Technology

• Part (II) – The WHAT
  • Computational Thinking, Definition
  • Curricula, Limitations

• Part (III) – The HOW
  • COOL – Informatics
  • Principles

• Part (IV) Examples

• Part (V) Conclusion/Outlook
Motivation (1/3)

• What is it all about?
  • Technology is **everywhere**
  • Important **economic** factor
  • There is also some **hype** and **mystery** behind

• **Complexity** is constantly raising
  • Voyager … 3 KLOC (1977),
  • Mars Rover 160 KLOC (2003),
  • ISS … 5 MLOC (2009),
  • Boing 787 … 6.4 MLOC (2011)
  • GMC … 100 MLOC (2012)

⇒ **What** can we do in order to prepare society/people for the actual and future demands, problems and opportunities?
Motivation (2/3)

- **Informatics** ([Computer | Computing] Science): Part of **STEM** (Science, Technology, Engineering and Mathematics)

  (1) **Informatics**: Science of systematic **information** processing – especially the science of automated **information** processing with the help of a **computer** system

  (2) **Information**: (a) Static/Object-related: **knowledge** brought into some suitable form (b) Process-oriented: process/result of the transfer/gain of **knowledge**

  (3) **Knowledge**: (a) yields at knowledge enhancement, (b) is bound to a **language** (c) is bound to a physical carrier

  (4) **Computer**: helps in mechanically processing information (that is stored in suitable form) and makes use of detailed instructions (the program) which are interpreted automatically

  ➔ A COMPUTER is a LANGUAGE-INTERPRETATION MACHINE

  ➔ CS is more than just programming and more than the use of new technologies
Motivation (3/3)

• It is all about languages, the use of languages, information and problem solving. One issue is detecting problem similarities, limitations and solution techniques. With that we can
  • understand problems and patterns
  • learn to work out solutions
  • produce/create something new and useful

4th Cultural-Technology
(J. Wing [1])

http://blogfarm15.mb.uni-flensburg.de/lisasophie/wp-content/uploads/sites/71/2016/01/Kulturtechniken.jpg

Computational Thinking (The What)  (1/4)

• There is “one definition” when going back to the roots:

Following J. Wing, Computational Thinking (CT) is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent. [2]

• But … there are several “schools”, looking at CT in a
  • broader sense (set of problem solving skills [3]),
  • narrow sense (linked to coding skills [4]), or taking a
  • position in-between (coding after plentiful experience of CT through play [5])

Computational Thinking (The What) (2/4)

• Duncan and Bell analyzed several curricula [6]:

<table>
<thead>
<tr>
<th>Themes</th>
<th>England Key Stage 1 (students from 5-7 years old)</th>
<th>Australia F-2 (students from 5-8 years old)</th>
<th>CSTA L1:3 (students from 5 to 8 years old)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td>Understand what algorithms are</td>
<td>Sorting and patterns</td>
<td>Logical problems</td>
</tr>
<tr>
<td>Programming</td>
<td>Create and debug simple programs; relationship with algorithms</td>
<td>Step-by-step procedures; sequences; instructions, robotic toys</td>
<td>Purpose of software; turtle instructions</td>
</tr>
<tr>
<td>Data representation</td>
<td></td>
<td>Patterns and symbols; pixels and file size</td>
<td>How 0s and 1s represent information</td>
</tr>
<tr>
<td>Digital devices &amp; infrastructure</td>
<td></td>
<td>HW &amp; SW components, features of a device, data transfer</td>
<td>Use input/output devices</td>
</tr>
</tbody>
</table>

Duncan and Bell analyzed several curricula [6] (contd.):

<table>
<thead>
<tr>
<th>Themes</th>
<th>England Key Stage 1 (students from 5-7 years old)</th>
<th>Australia F-2 (students from 5-8 years old)</th>
<th>CSTA L1:3 (students from 5 to 8 years old)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital devices and infrastructure</td>
<td>Create, organize, store, manipulate and retrieve content</td>
<td>Capture and manipulate data e.g. photo; downloading information; collect, organize, present data; use common Software</td>
<td>Writing tools; Organizing information (e.g. sorting); gather and communicate information; age appropriate research; use and create multimedia; concept mapping</td>
</tr>
<tr>
<td>Humans and computer</td>
<td>Recognize use beyond school; use safely and respect</td>
<td>Ergonomics, digital devices in everyday life; collaboration; ethical and safe use</td>
<td>Work collaboratively; careers that use computing; legal and ethical behavior</td>
</tr>
</tbody>
</table>
Computational Thinking (The What) (4/4)

- The curricula are mainly in agreement on what can and should be taught at each age level
- But, to make the most out of CT [1, p. 35], the following characteristics are, among others, important, too:
  - Conceptualization, not programming
    → **Modelling and Abstraction** at different levels
  - A way, human, not computers think
    → **Problem solving**, with/without a computer
  - Complements/combines mathematical/engineering thinking
    → **Language** skills are not to be neglected
  - For everyone, everywhere
    → **Integration** in different school subjects, using it in a variety of topics so that it is integral to human endeavor

COOL Informatics (The How) (1/4)

• Projects like “Informatik – ein Kinderspiel”, “Informatik erLeben” [7], “Computer Science Unplugged” [8] motivated us to
  • start our own “Informatics-Lab” in 2014,
  and
  • to define our own teaching approach, called “COOL Informatics” [9]

COOL Informatics (The How) (2/4)

- “COOL Informatics” is an acronym for:
  1. COoperative Open Learning [10]
  2. COmputer Supported Open Learning
  3. Simply “cool”

COOL Informatics (The How) (3/4)

- “COOL Informatics” is based on 4 principles [9, p.26]

<table>
<thead>
<tr>
<th>Principle</th>
<th>Teaching &amp; Learning Method</th>
<th>Neurodidactical Basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>Solution-based learning (worked examples); Step-by-step instructions + tasks</td>
<td>Pattern recognition Mirror neurons</td>
</tr>
<tr>
<td></td>
<td>Observational learning; Video tutorials; Hands-on, Minds-on Learning with all senses</td>
<td>Individual learning rhythm modality / multimedia effect</td>
</tr>
<tr>
<td>Cooperation</td>
<td>Team and group work; Peer tutoring and –teaching; Pair programming; Cross-curricular</td>
<td>“A joy (=knowledge) shared is a joy (=knowledge) doubled.” Recall = re-storage in</td>
</tr>
<tr>
<td></td>
<td>learning; Project-based learning</td>
<td>long-term memory Integrating individual needs, talents and competences as well as</td>
</tr>
<tr>
<td></td>
<td></td>
<td>practical relevance</td>
</tr>
</tbody>
</table>

• “COOL Informatics” is based on 4 principles [9, p.26] (contd.)

<table>
<thead>
<tr>
<th>Principle</th>
<th>Teaching &amp; Learning Method</th>
<th>Neurodidactical Basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individuality</td>
<td>Competence-based learning; Questioning; Self-organized learning with compulsory and optional tasks</td>
<td>Connecting new information to previous knowledge, Considering individual interests, needs, tasks, methods and learning rhythm</td>
</tr>
<tr>
<td>Activity</td>
<td>Hands-on, Minds-on; Learning by doing; Learning by animation, simulation and playing; Learning by playing and designing games (creative learning)</td>
<td>Knowledge must be newly created (constructed) by each learner (= constructivism) Learning is an active process (=progressive education)</td>
</tr>
</tbody>
</table>

Examples (1/5)

- [Mobile] Informatics-Lab at AAU-Klagenfurt
- > 1500 participants (of all ages) since 2014
- Covers “standard” topics like
  - Boolean Algebra / Logic
  - Binary Numbers
  - Coding and Encryption
  - Internet and Networks
  - Computer Systems, Hardware
  - Information Processing
Examples (2/5)

• The Informatics-Lab emphasizes:
  • Offering it **unplugged**
  • **Modelling**, Abstraction
  • **Communication, Teams**
  • **Cross-curricular** projects
    (e.g. cutting patterns, language teaching)
Examples (3/5)

- **CT_VS** – a project for the BMBF (Austrian Federal Ministry of Education and Women's Affairs) in 2015
- How to integrate CT in the curriculum of Austrian’s primary schools?
  - In-depth analysis of knowledge areas in computer science (Based on the ACM/IEEE CSC 2013)
  - In-depth analysis of curriculum → Guideline for teachers
Examples (4/5)

- **Software Engineering for Non-Informatics** [11]
  Implementing a language learning application

  - Setting: Vocational High-School (8 girls of 11th grade as "software engineers") and a lower secondary school (25 pupils of 6th grade as "customers").
  - Used MIT’s “App-Inventor” to overcome programming hurdles. But, the focus was on:
    - Modeling and Problem-Analysis
    - Communication Skills

**Examples (5/5)**

- Competences *(ACM/IEEE SE2004 Knowledge Areas)*

<table>
<thead>
<tr>
<th>CMP</th>
<th>Computing Essentials (172h): CS foundations, construction technologies &amp; tools, formal methods</th>
<th>VAV</th>
<th>Software V&amp;V (42h): reviews, testing, HCI testing &amp; evaluation, problem analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>FND</td>
<td>Mathematical &amp; Engineering Fundamentals (89h): logic, measures, discrete math, statistics, economics</td>
<td>MGT</td>
<td>Software Management (19h): planning, personnel, control &amp; configuration management</td>
</tr>
<tr>
<td>MAA</td>
<td>Modeling and Analysis (53h): modeling foundations, specification &amp; validation of requirements</td>
<td>PRO</td>
<td>Software Process (13h): process concepts and implementations</td>
</tr>
<tr>
<td>PRF</td>
<td>Professional Practice (35h): group dynamics, psychology, comm. skills, professionalism and ethics</td>
<td>QUA</td>
<td>Software Quality (16): quality concepts, culture, standards, and processes</td>
</tr>
<tr>
<td>DES</td>
<td>Software Design (45h): concepts, strategies, architectures, HCI design</td>
<td>EVL</td>
<td>Software Evolution (10h): evolution processes and activities</td>
</tr>
</tbody>
</table>
Summary and Outlook

• We are living in a **socio-economic world** and **change** is prevalent.
  • But, there is **a lot can we** do in order to prepare society/people for the actual and future demands, problems and opportunities.
  • Improving **CT skills** is one thing we can do.
  • The basis is conveying the skills in **COOL manner**.
  • With a lot of brain-friendly initiatives, we will not be living in the following future:

"In the future, airplanes will be flown by a dog and a pilot. And the dog's job will be to make sure that if the pilot tries to touch any of the buttons, the dog bites him."

(Scott Adams / Author of “Dilbert“)
Questions?

Contact:
Andreas.Bollin@aau.at
Motivation (2/3)

Computational Thinking and COOL in Primary and Secondary Schools - A. Bollin

Informatics (Computer Science): Part of STEM (Science, Technology, Engineering and Mathematics)

1) Informatics: Science of systematic information processing – especially the science of automated information processing with the help of a computer system


(2) Information: (a) Static/Object-related: knowledge brought into some suitable form (b) Process-oriented: process/result of the transfer/gain of knowledge

(3) Knowledge: (a) yields at knowledge enhancement, (b) is bound to a language (c) is bound to a physical carrier

(4) Computer: helps in mechanically processing information (that is stored in suitable form) and makes use of detailed instructions (the program) which are interpreted automatically

→ A COMPUTER is a LANGUAGE-INTERPRETATION MACHINE

→ CS is more than just programming and more than the use of new technologies