Capturing and Characterising Notional Machines

Capturing and Characterising Notional Machines

Sally Fincher (co-leader)
School of Computing
University of Kent
Canterbury, Kent, UK
S.A.Fincher@kent.ac.uk

Johan Jeuring (co-leader)
Department of ICS
Utrecht University
The Netherlands
J.T.Jeuring@uu.nl

Craig S. Miller (co-leader)
School of Computing
DePaul University
Chicago, IL, USA
cmiller@cs.depaul.com

Peter Donaldson
Computing Science
University of Glasgow
Glasgow, Scotland, UK
peter.donaldson.2@glasgow.ac.uk

Benedict du Boulay
Department of Informatics
University of Sussex
Brighton, Sussex, UK
b.du-boulay@sussex.ac.uk

Matthias Hauswirth
Faculty of Informatics
USI Università della Svizzera italiana
Lugano, Switzerland
matthias.hauswirth@gmail.com

Arto Hellas
Department of Computer Science
Aalto University
Espoo, Finland
arto.hellas@aalto.fi

Felienne Hermans
LIACS
Leiden University
The Netherlands
f.f.j.hermans@liacs.leidenuniv.nl

Colleen Lewis
Department of Computer Science
Harvey Mudd College
Claremont, CA, USA
lewis@cs.hmc.edu

Andreas Mühling
Institute for Computer Science
Kiel University
Kiel, Germany
andreas.muehling@infomatik.uni-kiel.de

Janice L. Pearce
Computer Science Department
Berea College
Berea, KT,, USA
pearcej@berea.edu

Andrew Petersen
Department of ICS
University of Toronto
Toronto, Canada
andrew.petersen@utoronto.ca

ABSTRACT

A *notional machine* is a pedagogic device to assist the understanding of some aspect of programs or programming. It is typically used to support explaining a programming construct, or the user-understandable semantics of a program. For example, *a variable is like a box with a label*, and assignment copies or moves a value into that box.

This working group will capture examples of notional machines from actual pedagogical practice, as expressed in textbooks (or other teaching materials) or used in the classroom. We will interview at least 30 teachers about their experience with, and perceptions of, the use of notional machines in teaching. Using the interviews, we will work on devising and refining a form to characterise essential features of notional machines. We will also attempt to relate them to each other to describe potential learning sequences or progressions. The working group report will contain descriptions of notional machines used at different levels in education, in different countries, by many teachers.

The resulting catalogue of notional machines will allow a teacher to select a machine for a particular use, permit comparison between them, and provide a starting point for further categorization and analysis of notional machines.

Additionally, we will make more theoretical explorations. We will explore a variety of *presentational formats*, examining what is necessary and what superfluous; we will look for *dimensions of comparison* and will examine how notional machines are instantiated across the discipline. We argue that the creation and use of notional machines is potentially a *signature pedagogy* for computing [1] and that creating and using notional machines represents a certain level of pedagogic sophistication that might be an indicator of pedagogic content knowledge (PCK).

CCS CONCEPTS: Applied Computing. Education

KEYWORDS

Notional machines, pedagogical practice, signature pedagogies, PCK

Capturing and Characterising Notional Machines Sally Fincher, Johan Jeuring, Craig S Miller

ITiCSE '20, June 15–19, 2020, Trondheim, Norway

ACM ISBN 978-1-4503-6874-2/20/06.
https://doi.org/10.1145/3341525.3394988

1 What is a Notional Machine?

First identified by du Boulay [2,3,4] a notional machine is a pedagogic device to assist the understanding of some aspect of programs or programming. It is normally associated with teaching novices, but can also be applied in other teaching or training situations with more expert participants. It is called "notional" in the sense that what is being described is a simplified, partially true, version of the truth. It is called a "machine" because it describes both entities and their actions. For example, *a variable is like a box with a label*, and assignment copies/moves a value into that box.

A notional machine is not a mental representation in itself, although learners will develop an understanding of the computer and programming, and that understanding may well be influenced by any notional machines that they have been exposed to.

A crucial aspect of a notional machine is that it should simplify an actual concept or skill as an aid to understanding. It should also encompass the associated epiphenomena of programs and programming, such as the names of program constructs, the causes and content of error messages, the names on buttons in the environment (e.g. "save", "file" etc), the locations of files containing programs, the difference between editing and running a program and so on.

For a beginner faced with learning how to design, code, test and debug programs, there are three areas in which a notional machine may play a role.

- The programming environment in which the program-ming is to be undertaken. This includes the file system and tools for coding, editing and debugging.
- The programming language itself. This includes the basic constructs of the language as well as any built-in language features for aggregating small chunks of code into larger units.
- The tasks of programming, where the human, in a way, *acts* as a "machine" undertaking actions such as coding, debugging, editing and testing.

3. Preliminary work

The proposal for this working group originates from Dagstuhl seminar 19281, Notional Machines and Programming Language Semantics in Education, 7th-12th July 2019. At this seminar, a subgroup was formed to gather and examine practical examples of notional machines. The small corpus collated there provides preliminary work for the working group, and includes the following:

- An interview protocol for eliciting notional machines.
- A collection of more than 10 interview recordings and transcriptions.
- A rough working set of attributes to describe (and potentially help classify) notional machines.

2 Objectives of this Working Group

This group has four objectives:

(a) To conduct a literature review, to ground and inform the work;
(b) To capture examples of notional machines in use;
(c) To catalogue them in a common scheme; and,
(d) To arrange them in clusters or sequences.

The outputs of the working group will form the first contextualized collection of notional machines from multiple sources. We anticipate that it will not only provide a useful collection for teachers, but also a foundational resource for other researchers interested in this area [5] [6].

ACKNOWLEDGMENTS

This work builds on initial activity undertaken at Dagstuhl seminar 19281, *Notional Machines and Programming Language Semantics in Education*, 7th-12th July 2019. The three organizers of this working group participated in the Dagstuhl seminar.

REFERENCES

[1] Shulman, L. (2005) Signature Pedagogies in the Professions *Daedalus*, 134(3), 52-59

[2] du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57– 73.

[3] du Boulay, B., & O'Shea, T. (1981). Teaching novices programming. In M. J. Coombs & J. L. Alty (Eds.), *Computing Skills and the User Interface* (pp. 147–200). Cambridge, MA: Academic Press.

[4] du Boulay, B., O'Shea, T., & Monk, J. (1981). The black box inside the glass box: presenting computing concepts to novices. *International Journal of Man–Machine Studies*, 14(3), 237–249.

[5] Sorva. Juha (2013). Notional machines and introductory programming education. ACM Trans. Comput. Educ. 13, 2, Article 8 (July 2013), 31 pages. DOI:https://doi.org/10.1145/2483710.2483713

[6] Berry, Michael (2015) The Design and Implementation of a Notional Machine for teaching Introductory Programming, PhD thesis, University of Kent.