Informatics

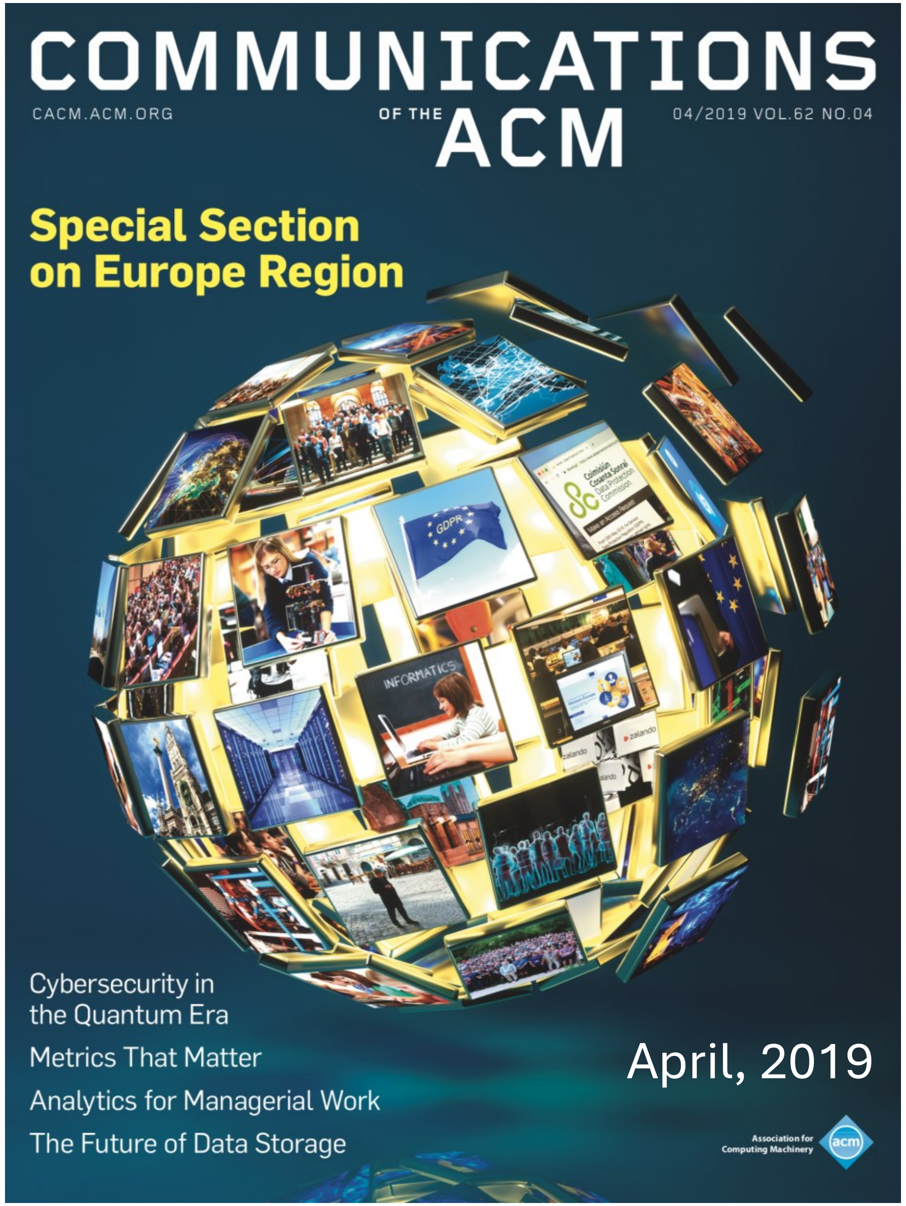
– a fundamental discipline in the 21st century and a driver for disciplinary renewal

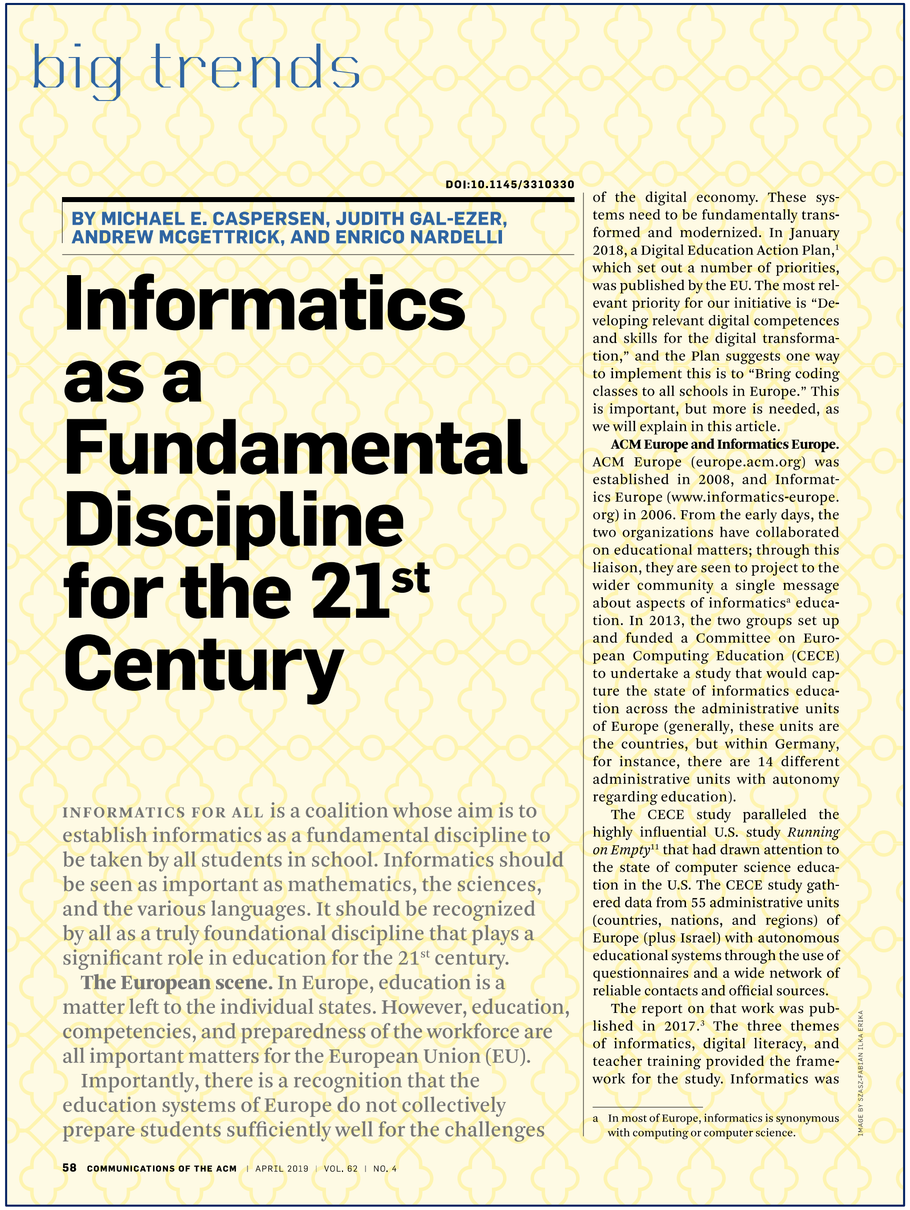
Michael E. Caspersen
It-vest – networking universities
Denmark
mec@it-vest.dk

Line Have Musaeus
Department of Computer Science
Aarhus University, Denmark
lh@cs.au.dk

It-vest
networking universities

Center for Computational Thinking and Design





big trends

DOI:10.1145/3310330

BY MICHAEL E. CASPERSEN, JUDITH GAL-EZER, ANDREW MCGETTRICK, AND ENRICO NARDELLI

Informatics as a Fundamental Discipline for the 21st Century

INFORMATICS FOR ALL is a coalition whose aim is to establish informatics as a fundamental discipline to be taken by all students in school. Informatics should be seen as important as mathematics, the sciences, and the various languages. It should be recognized by all as a truly foundational discipline that plays a significant role in education for the 21st century.

The European scene. In Europe, education is a matter left to the individual states. However, education, competencies, and preparedness of the workforce are all important matters for the European Union (EU).

Importantly, there is a recognition that the education systems of Europe do not collectively prepare students sufficiently well for the challenges of the digital economy. These systems need to be fundamentally transformed and modernized. In January 2018, a Digital Education Action Plan,[1] which set out a number of priorities, was published by the EU. The most relevant priority for our initiative is "Developing relevant digital competences and skills for the digital transformation," and the Plan suggests one way to implement this is to "Bring coding classes to all schools in Europe." This is important, but more is needed, as we will explain in this article.

ACM Europe and Informatics Europe. ACM Europe (europe.acm.org) was established in 2008, and Informatics Europe (www.informatics-europe.org) in 2006. From the early days, the two organizations have collaborated on educational matters; through this liaison, they are seen as project to the wider community a single message about aspects of informatics[a] education. In 2013, the two groups set up and funded a Committee on European Computing Education (CECE) to undertake a study that would capture the state of informatics education across the administrative units of Europe (generally, these units are the countries, but within Germany, for instance, there are 14 different administrative units with autonomy regarding education).

The CECE study paralleled the highly influential U.S. study *Running on Empty*[11] that had drawn attention to the state of computer science education in the U.S. The CECE study gathered data from 55 administrative units (countries, nations, and regions) of Europe (plus Israel) with autonomous educational systems through the use of questionnaires and a wide network of reliable contacts and official sources.

The report on that work was published in 2017.[3] The three themes of informatics, digital literacy, and teacher training provided the framework for the study. Informatics was

a In most of Europe, informatics is synonymous with computing or computer science.

IMAGE BY SZASZ-FABIAN ILKA ERIKA

informatics for all

It-vest
networking universities

Center for Computational Thinking and Design

The Vision

Informatics is a new aspect of 'bildung' – a new fundamental competence for all

Art	German	Music
Economy	Social science	...
Geology	Marketing	Technology
History	Physics	Design
Classical history	Spanish	Geography
English	Psychology	Biology
Chemistry	Literature	Chinese
Biotechnology		

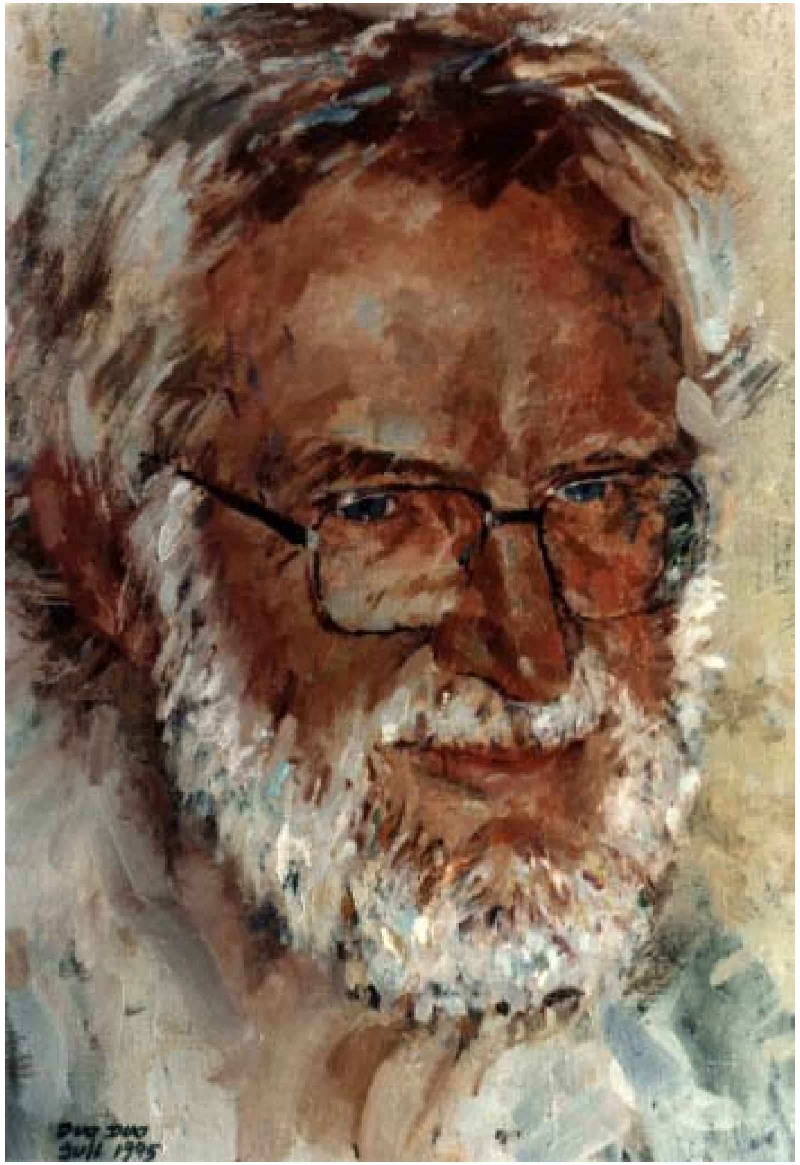
Informatics

Language **Informatics** Mathematics

Mathematics is (primarily) the language of the natural sciences

Informatics $\underbrace{\text{is a}}_{}$ language of all disciplines

becoming



Peter Naur (1928-2016)

Turing Laureate (2005)

1966-67 Danish Broadcasting Corporation's Rosenkjær Lectures

To conceive the proper place of informatics in the curriculum, it is natural to compare with subjects of similar character. One will then realise, that languages and mathematics are the closest analogies.

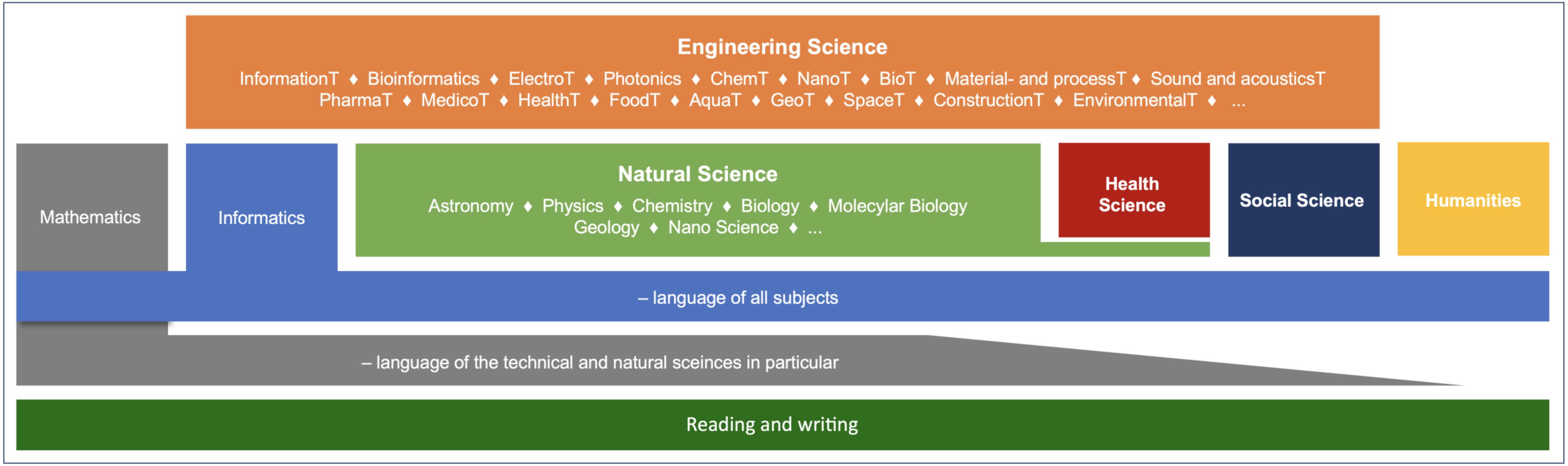
Common for the three is also their character as tools for many other subjects.

When informatics has become established in general education, the mystery surrounding computers will dissolve.

This is perhaps the most important reason for including informatics in general education. This will ensure that the ruling of computers and their use is not left to a small group of experts, but will become a usual political matter and thus, through the political system, comes to lie where it should – with all of us.

Foundations of competences

Informatics – the new kid on the block



Informatics is the language and the science behind the digital revolution.
Informatics is becoming a language of all disciplines.
Informatics – a fundamental discipline for the 21st century.

Informatics is now playing the role, which mathematics did from the 17th through the 20th centuries: Providing an orderly, formal framework and exploratory apparatus for other sciences.
George Djorgovski (2005)

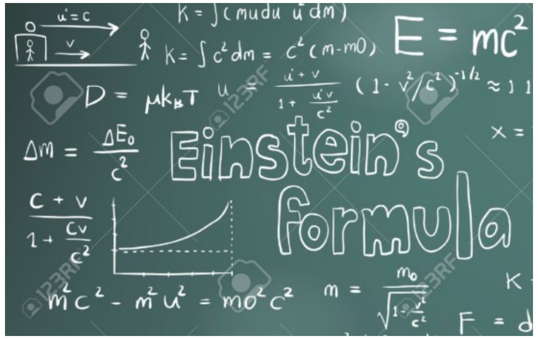
Computational language

– A new cultural technique

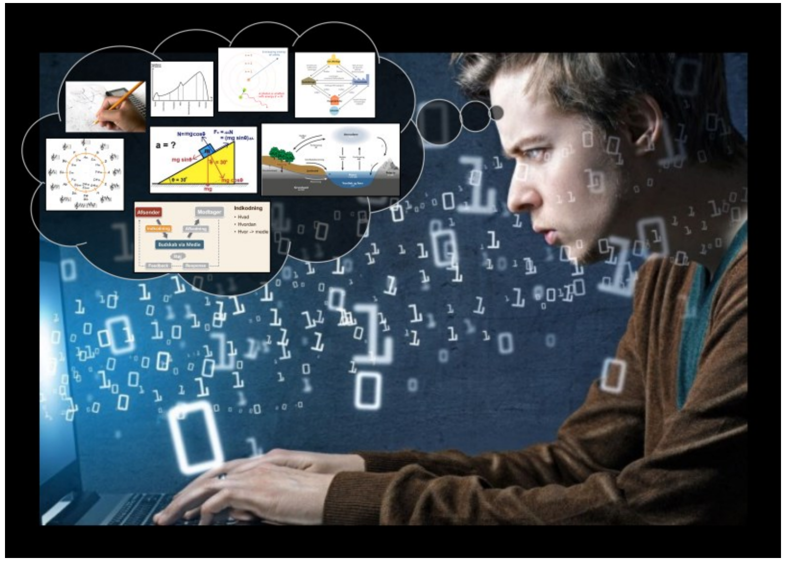
Spoken language

Written language

Mathematical language



Computational language



Computational literacy

Computational literacy/fluency

Two videos from the Lifelong Kindergarten group @ MIT Media Lab

CACM, November 2020

Coding is a new way for people to share ideas with the world
– an extended form of writing.

Like we learn to write and write to learn, we must learn to code and code to learn.

Natural vs. mathematical language

It is indeed
too odd
    for words
that half's
three quarters
    of two thirds.

    - Piet Hein

$$\frac{1}{2} = \frac{3}{4} \cdot \frac{2}{3}$$

Mathematical vs. computational language

The net vertical force from four sides is

$$\left(T\frac{\partial\zeta}{\partial x}\bigg|_{x+dx} - T\frac{\partial\zeta}{\partial x}\bigg|_{x}\right)dy + \left(T\frac{\partial\zeta}{\partial y}\bigg|_{y+dy} - T\frac{\partial\zeta}{\partial x}\bigg|_{y}\right)dx = T\left(\frac{\partial^2\zeta}{\partial x^2} + \frac{\partial^2\zeta}{\partial y^2}\right)dx\,dy$$

Continuity of vertical force on an unit area of the surface requires

$$p_o + p + T\left(\frac{\partial^2\zeta}{\partial x^2} + \frac{\partial^2\zeta}{\partial y^2}\right) = 0.$$

Hence

$$-\rho g\zeta - \rho\frac{\partial\Phi}{\partial t} + T\left(\frac{\partial^2\zeta}{\partial x^2} + \frac{\partial^2\zeta}{\partial y^2}\right) = 0, \quad z = 0. \tag{1.12}$$

which can be combined with the kinematic condition (1.3) to give

$$\frac{\partial^2\Phi}{\partial t^2} + g\frac{\partial\Phi}{\partial z} - \frac{T}{\rho}\nabla^2\frac{\partial\Phi}{\partial z} = 0, \quad z = 0 \tag{1.13}$$

When viscosity is neglected, the normal fluid velocity vanishes on the rigid seabed,

$$\mathbf{n}\cdot\nabla\Phi = \mathbf{0} \tag{1.14}$$

Let the sea bed be $z = -h(x, y)$ then the unit normal is

$$\mathbf{n} = \frac{(\mathbf{h_x}, \mathbf{h_y}, \mathbf{1})}{\sqrt{\mathbf{1} + \mathbf{h_x^2} + \mathbf{h_y^2}}} \tag{1.15}$$

Hence

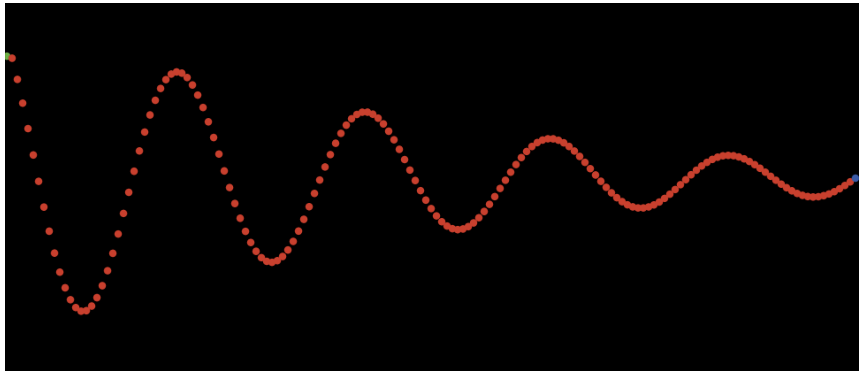
$$\frac{\partial\Phi}{\partial z} = -\frac{\partial h}{\partial x}\frac{\partial\Phi}{\partial x} - \frac{\partial h}{\partial y}\frac{\partial\Phi}{\partial y}, \quad z = -h(x, y) \tag{1.16}$$

Central

Stochastic methods, differential equations and lots of other “good stuff”

40 pages of non-trivial mathematics

Wave model



```
set yvel yvel + ((([ypos] of (turtle (who - 1))) - ypos) +
                 (([ypos] of (turtle (who + 1))) - ypos))
set yvel ((1000 - friction) / 1000) * yvel  ;; apply friction
```

Decentral

Simple local rule: (a+b) / 2
Emergence and dynamics "for free"

Agent-based modelling (ABM)

Mathematical vs. computational language (2)

$$S \to I \to R,$$

mens den totale populationsstørrelse $N = S + I + R$ er konstant.

Hvis antallet af modtagelige og smittende individer til tiden $t = t_0$ er $S(t_0)$ hhv. $I(t_0)$, så vil antallet af modtagelige til tiden $t = t_0 + \Delta t$ være

$$S(t_0 + \Delta t) = S(t_0) - c\frac{I(t_0)}{N}S(t_0)\Delta t, \tag{1}$$

idet hver af de modtagelige individer har $c\Delta t$ kontakter, og sandsynligheden for at en kontakt fører til smitte er I/N. Tilsvarende bliver antallet af smittende individer

$$I(t_0 + \Delta t) = I(t_0) + \frac{c}{N}I(t_0)S(t_0)\Delta t - bI(t_0)\Delta t, \tag{2}$$

hvor b angiver den rate, hvormed syge individer bliver raske og ophører med at smitte. Da populationsstørrelsen er konstant, kan $R(t)$ beregnes som $R(t) = N - S(t) - I(t)$.

Ligning (1) og (2) er udledt under antagelse af, at S og I ikke ændres i løbet af tidsrummet Δt. Tidsintervallet skal altså være lille og ved grænseovergangen $\Delta t \to 0$ fås nu to koblede ikke lineære differentialligninger:

$$\begin{aligned} S'(t) &= -\beta S(t)I(t) && (3)\\ I'(t) &= \beta S(t)I(t) - bI(t), && (4) \end{aligned}$$

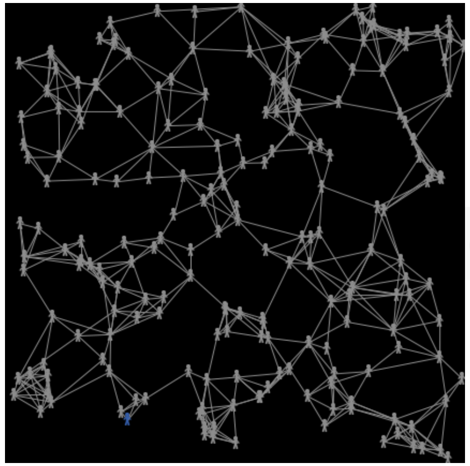
hvor mærket efter den variable betyder differentiation efter tiden. Parameteren $\beta = c/N$ kaldes *transmissionsparameteren*; den er matematisk bekvem, men vanskelig at tolke biologisk.

Central

Stochastic methods, differential equations and lots of other "good stuff"

25 pages of non-trivial mathematics

Epidemic model



The Washington Post

March 14, 2020

```
ask people with [infected?][
  ask my-links with [random-float 1 < probability-of-interaction] [
    if random 100 < 15[ ask other-end [ if not infected? and not immune? [set infe
    ]
  ]
```

Decentral

Simple local rule
Emergence and dynamics "for free"

Agent-based modelling (ABM)

Kristen Nygaard: To program (is to model) is to understand

1926-2002

Nygaard on object-oriented programming

A program execution is regarded as a physical model, simulating the behaviour of either a real or imaginary part of the world.

"To program is to understand"

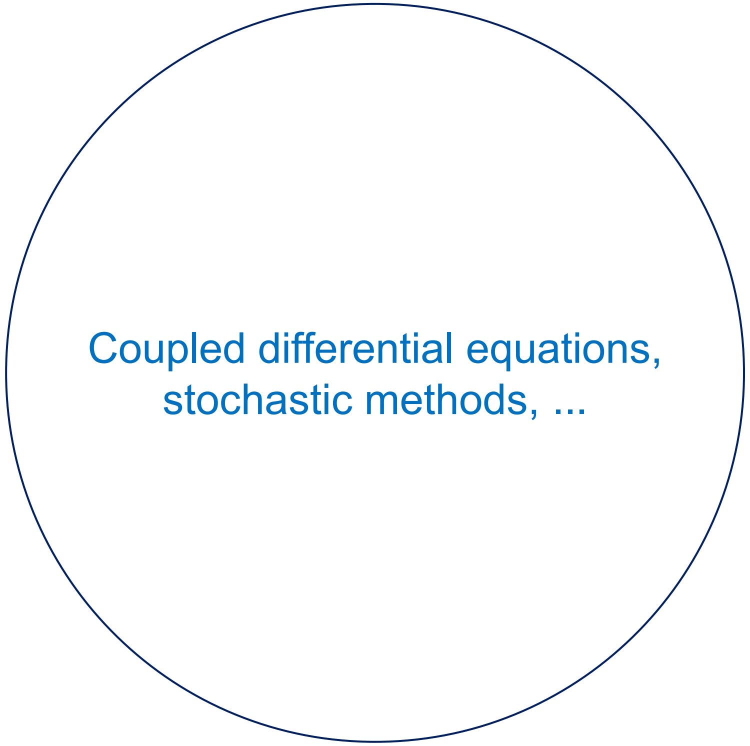
Nygaard thought programming should not be considered a low-level technical discipline designed just to accomplish a specific task, but that writing a program should enhance understanding of the problem domain.

A.M. Turing Award

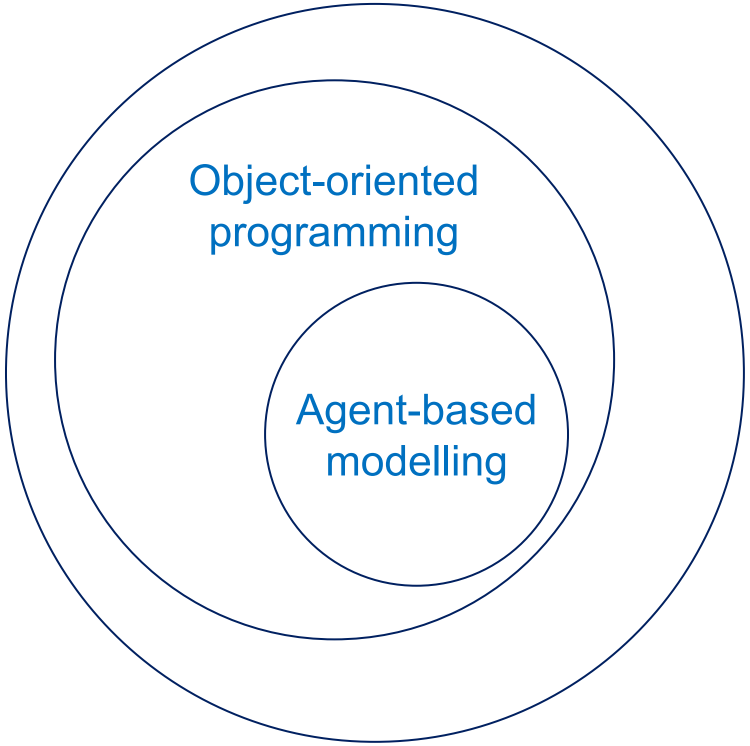
The BETA book

Modelling of dynamic systems

– mathematics vs. informatics

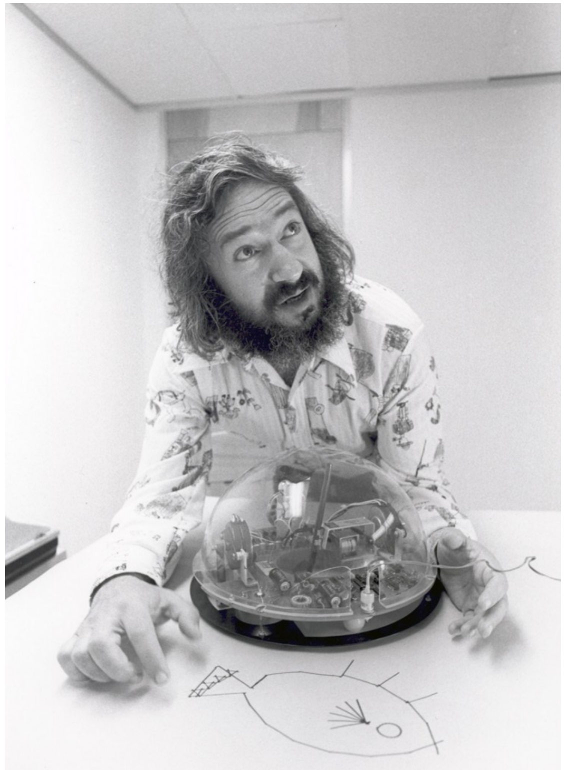


Mathematical modelling



Computational modelling

Beyond the centralized mindset

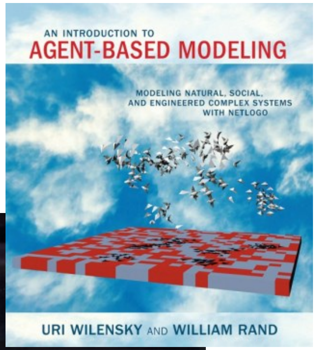


Seymour Papert and one turtle

Logo, 1967

Mitch Resnick and 1,000 turtles

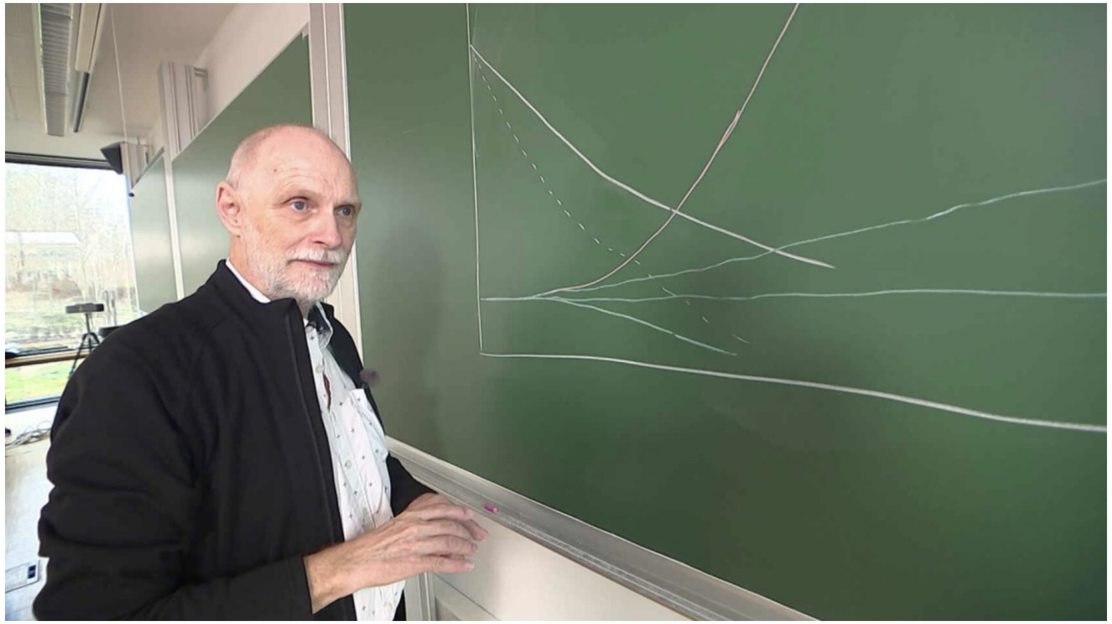
StarLogo
(PhD, 1992)



Uri Wilensky (and a dolphin)

NetLogo
(PhD, 1993)

Introduction to mathematical infection epidemiology



Viggo Andreasen

Associate professor in mathematical epidemiology

Roskilde University

Frequently on national TV during the pandemic...

Introduktion til matematisk infektionsepidemiologi

Viggo Andreasen

IMFUFA

Roskilde Universitetcenter, 4000 Roskilde

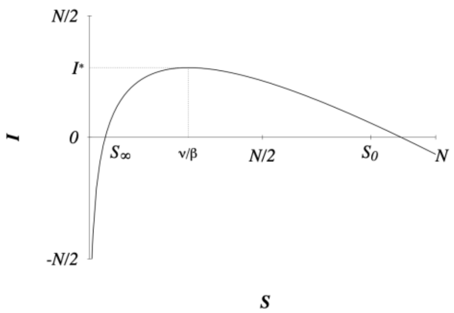


Figure 3: Epidemikurven efter elimination af t

sådan at løsningen med begyndelsesbetingelse $(S(t_0), I(t_0)) = (S_0, I_0)$ kan bestemmes som

$$I(S) - I_0 = \int_{S_0}^{S} \frac{dI}{dS}\, dS.$$

Ved at isolere $I(S)$ fås

$$\begin{aligned} I(S) &= -S + \frac{b}{\beta}\ln S + (S_0 - \frac{b}{\beta}\ln S_0 + I_0) \\ &= (N - R_0) - S + \frac{b}{\beta}\ln\frac{S}{S_0}. \end{aligned}$$

Funktionen $I(S)$ giver sammenhængen mellem S og I gennem epidemien, fig. 3. Man kan ikke aflæse det tidslige forløb direkte af $I(S)$, men da vi ved at S aftager gennem hele epidemien, vil punkterne på kurven blive gennemløbet *fra højre mod venstre.*

Kurven $I(S)$ har maksimum i

$$S^* = \frac{b}{\beta},$$

1.3 Epidemimodellen

Efter disse indledende bemærkninger kan vi nu opstille vores model for forløbet af en epidemi. Vi vil gå ud fra, at værtspopulationen er *lukket*, sådan at der hverken forekommer immigration, emigration eller fødsler og dødsfald under epidemien. For epidemier med et kortvarigt forløb som pestepidemien fra fig. 1 virker denne antagelse rimelig (antallet af pestdødsfald for epidemien i fig. 1 er forholdsvis beskedent; "kun" en tredjedel af Københavns befolkning døde). For at gøre udregningerne simple vil vi endvidere gå ud fra, at latenstiden er 0, og at sygdommen giver livslang immunitet sådan, at sygdomsforløbet følger diagrammet

$$S \to I \to R,$$

mens den totale populationsstørrelse $N = S + I + R$ er konstant.

Hvis antallet af modtagelige og smittende individer til tiden $t = t_0$ er $S(t_0)$ hhv. $I(t_0)$, så vil antallet af modtagelige til tiden $t = t_0 + \Delta t$ være

$$S(t_0 + \Delta t) = S(t_0) - c\frac{I(t_0)}{N}S(t_0)\Delta t, \qquad (1)$$

idet hver af de modtagelige individer har $c\Delta t$ kontakter, og sandsynligheden for at en kontakt fører til smitte er I/N. Tilsvarende bliver antallet af smittende individer

$$I(t_0 + \Delta t) = I(t_0) + \frac{c}{N}I(t_0)S(t_0)\Delta t - bI(t_0)\Delta t, \qquad (2)$$

hvor b angiver den rate, hvormed syge individer bliver raske og ophører med at smitte. Da populationsstørrelsen er konstant, kan $R(t)$ beregnes som $R(t) = N - S(t) - I(t)$.

Ligning (1) og (2) er udledt under antagelse af, at S og I ikke ændres i løbet af tidsrummet Δt. Tidsintervallet skal altså være lille og ved grænseovergangen $\Delta t \to 0$ fås nu to koblede ikke lineære differentialligninger:

$$\begin{aligned} S'(t) &= -\beta S(t)I(t) && (3) \\ I'(t) &= \beta S(t)I(t) - bI(t), && (4) \end{aligned}$$

hvor mærket efter den variable betyder differentiation efter tiden. Parameteren $\beta = c/N$ kaldes *transmissionsparameteren*; den er matematisk bekvem, men vanskelig at tolke biologisk.

At frequent meetings in the Prime Ministry, health minister Magnus Heunicke used the poster with the two curves to illustrate the development of infection in the population.

1st ABM example: Spread of infection (1)

Mathematical analysis vs. agent-based modelling

Coupled differential equations

Description of the phenomenon at systems level – a centralised model

$$\frac{dS}{dt} = -\frac{\beta}{N} I(t)S(t)$$

$$\frac{dI}{dt} = \frac{\beta}{N} I(t)S(t) - \gamma I(t)$$

$$\frac{dR}{dt} = \gamma I(t)$$

Accessible for almost none

Agent-based modelling

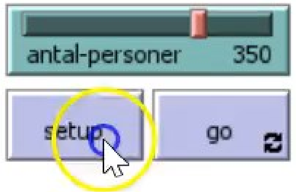
Description of the phenomenon at agent level – a local model

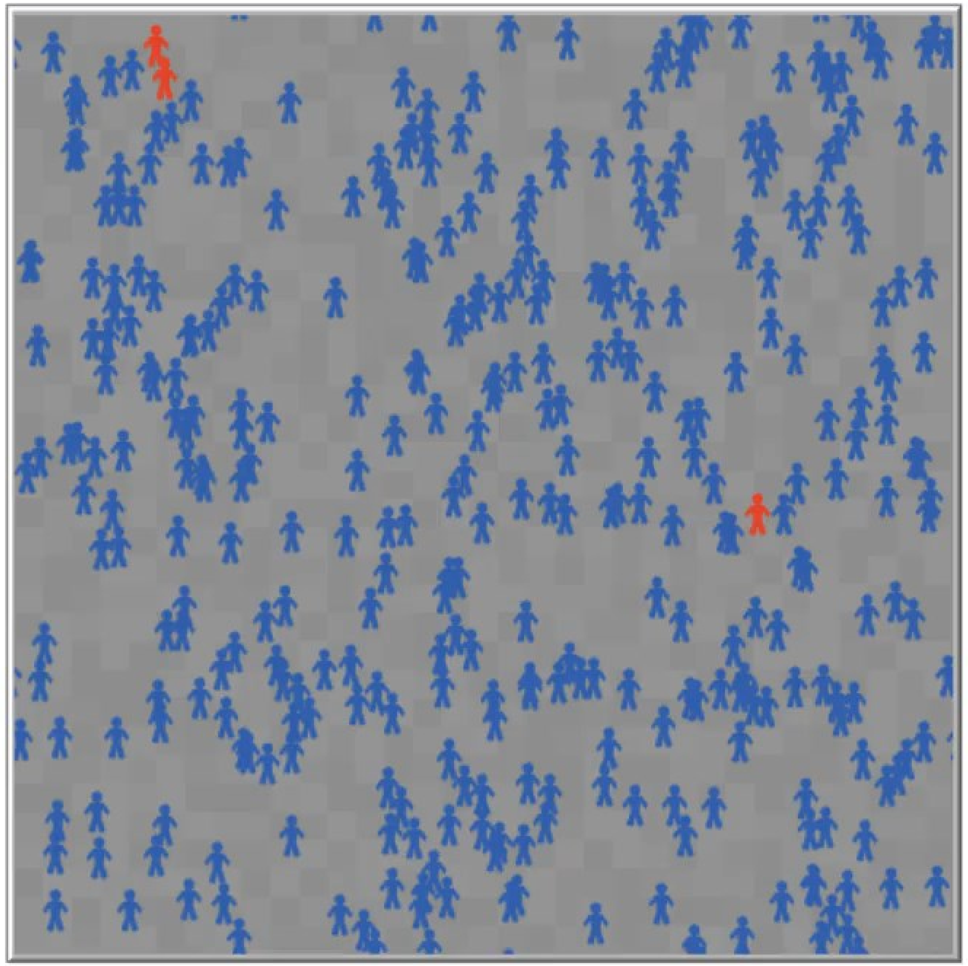
```
ask persons with [ color = blue ] [
  if any? persons-here with [color = red] [ set color red ]
]
```

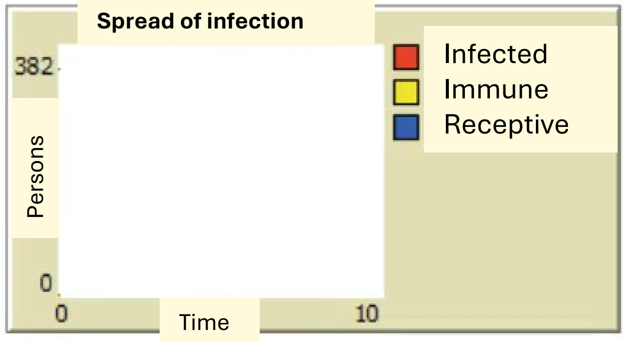
```
ask persons with [color = red] [
  if infection-time > 100 [ set color yellow ]
]
```

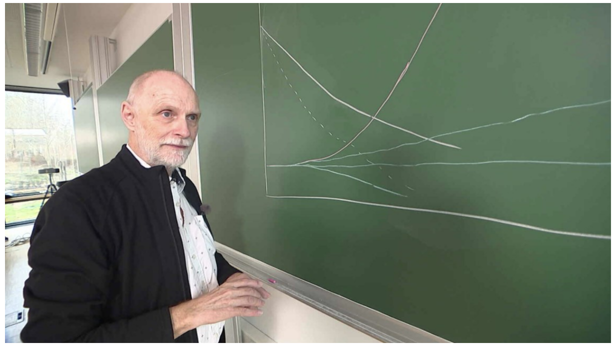
Transparent and accessible for almost all

1st ABM example: Spread of infection (2)









A white paper from 2013

Future Directions in Computing Education Summit Part One: Important Computing Education Research Questions

[Peer-reviewed version in proceedings of Koli Calling 2014](#)

Model-Based Thinking and Practice

A Top-down Approach to Computational Thinking

Palle Nowack and Michael E. Caspersen
Centre for Science Education
Aarhus University, Denmark
{nowack, mec}@cse.au.dk

ABSTRACT

In this paper, we discuss using models and modeling in a new way to teach basic computing to pupils within the K-12 segment. We argue why we believe understanding and creating models are fundamental skills for all pupils as it can be characterized as the skill that enable us to analyze and understand phenomena as well as design and construct artifacts. We also try to characterize the essence of model-based thinking and practice. We propose that a strong focus on the relation between mental models (of real or imaginary systems) and computerized models (embedded in computer-based systems) could provide a new approach to teaching computing. This approach should clarify and make explicit the role of models in computing in connection with other subject areas. We believe that such an approach would strongly broaden the participation in computing, as it will allow more pupils to become active creators with computing.[1]

It-vest
networking universities

2nd ABM example: Tipping point – forest fire (1)

Trees are the agents

Green trees

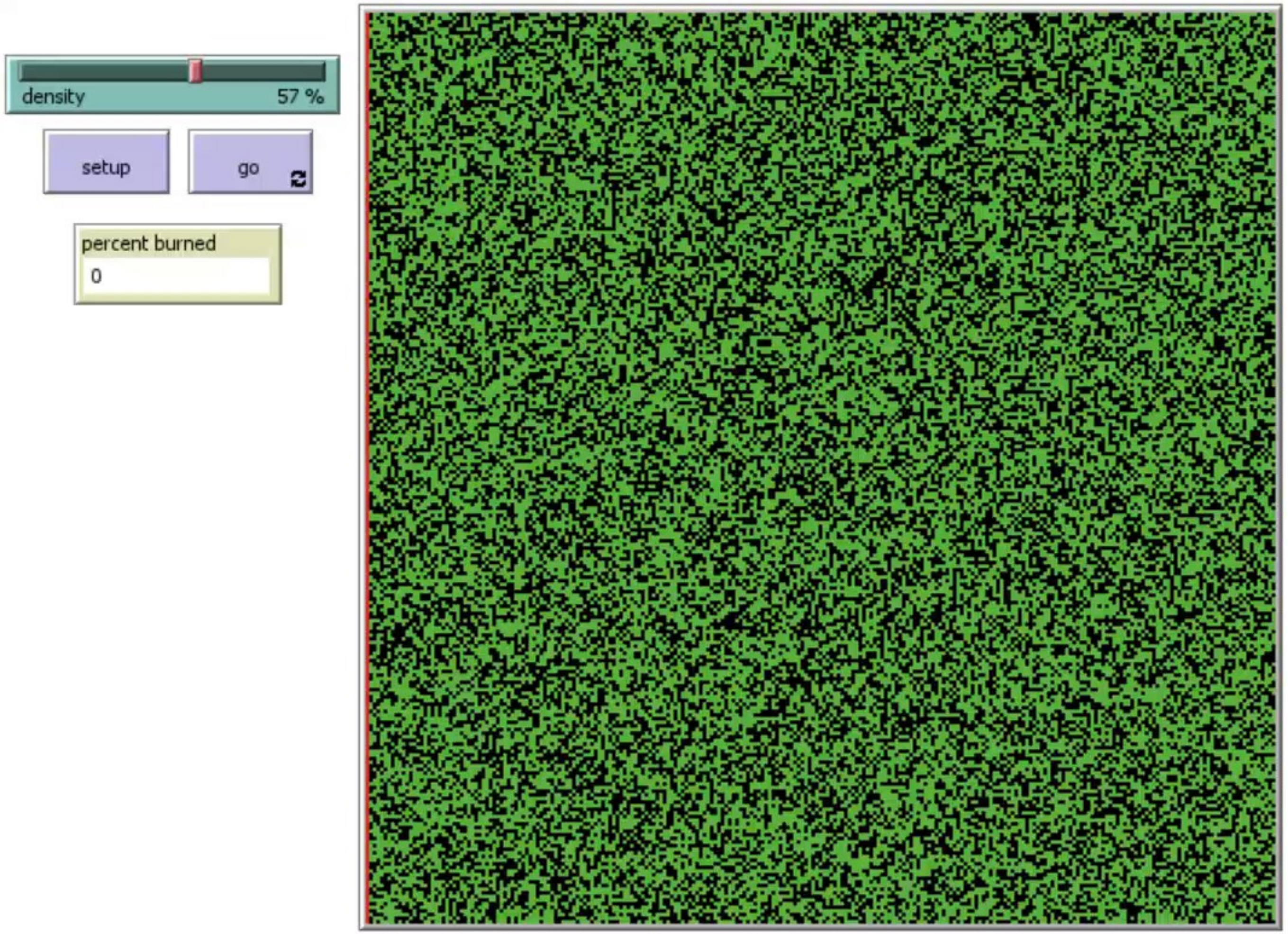
One rule: 1) Do nothing

Red (burning) trees

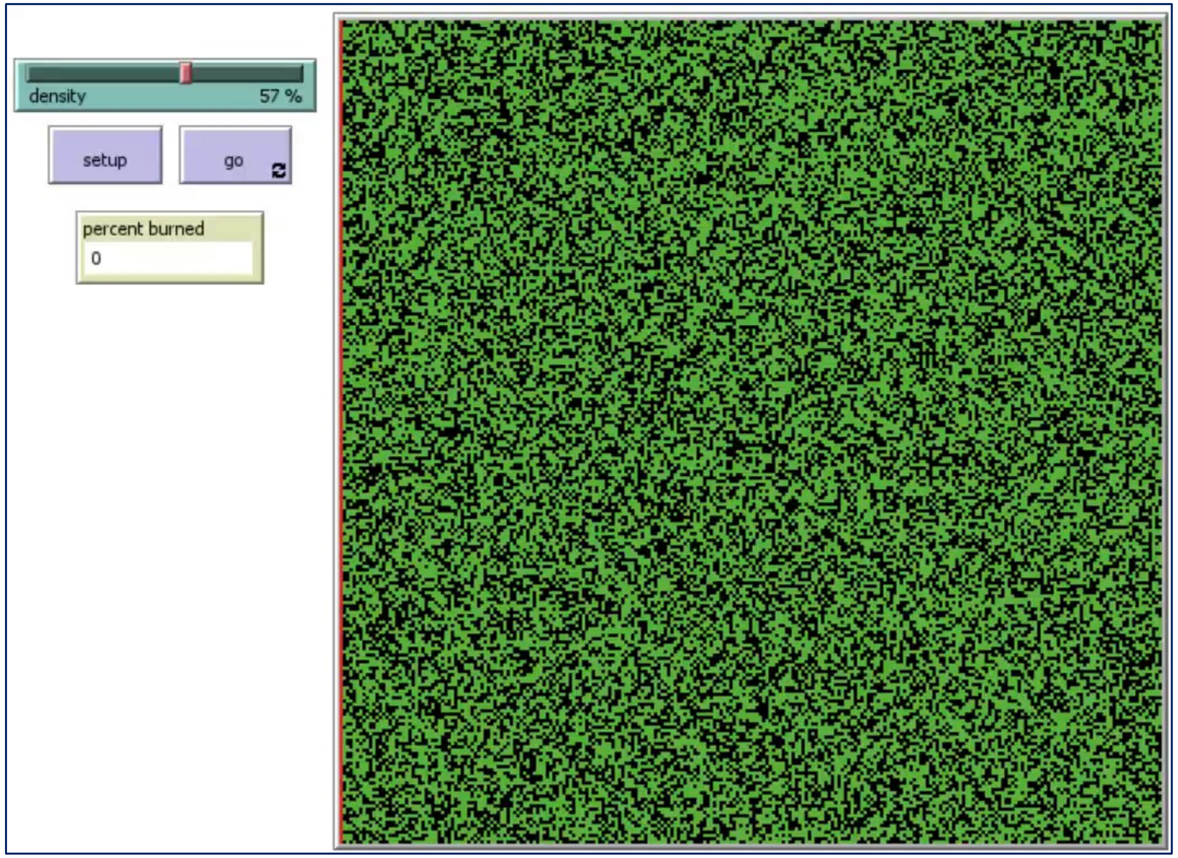
Two rules:

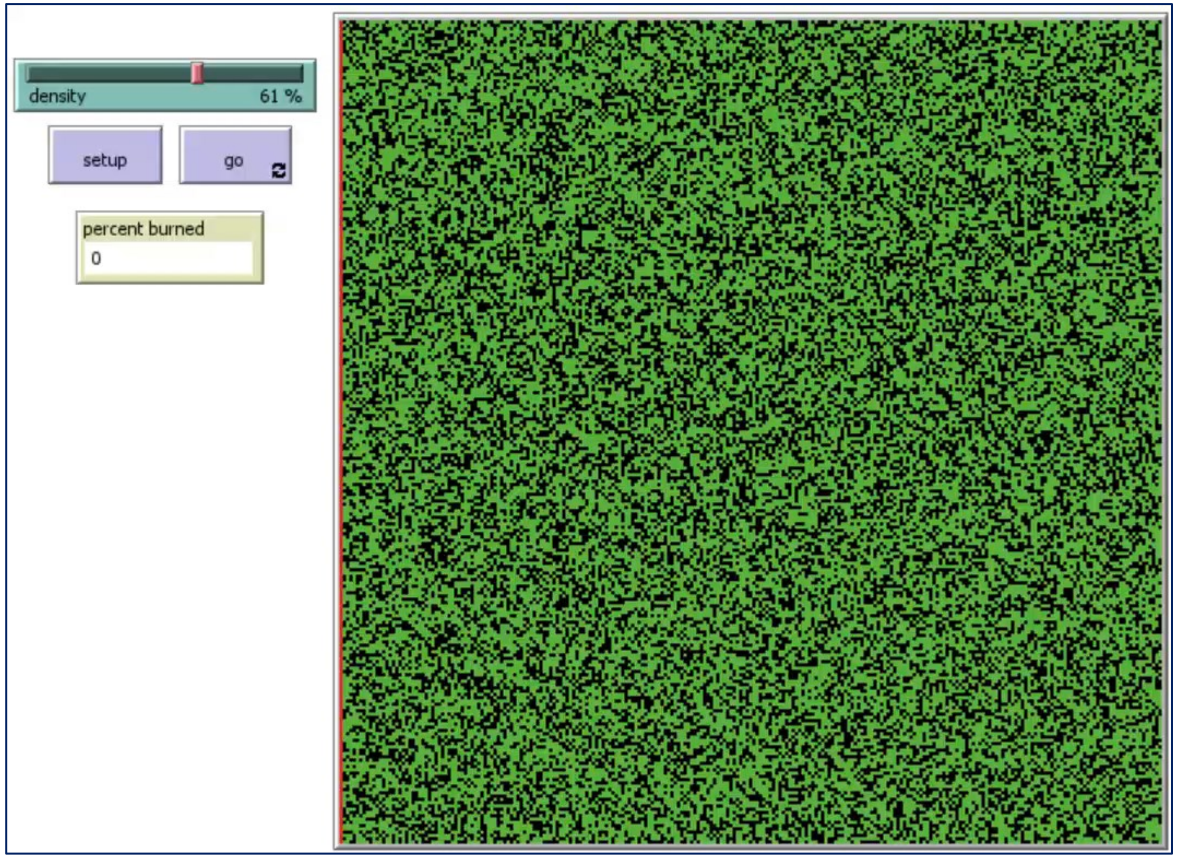
1) Ignite neighbouring trees
2) Burn out

Density: 57 %



2nd ABM example: Tipping point – forest fire (2)





Density: 57 %

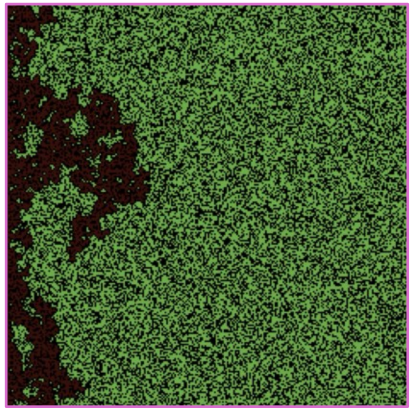
Repeated simulations

Density: 61 %

2^{nd} ABM example: Tipping point – forest fire (3)

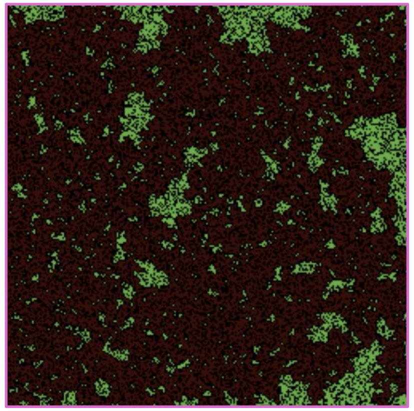
Density: 57 %: ~ 8,3 %

[12.7 ; 9.1 ; 8.0 ; 7.2 ; 4.4 ; 8.0 ; ...]



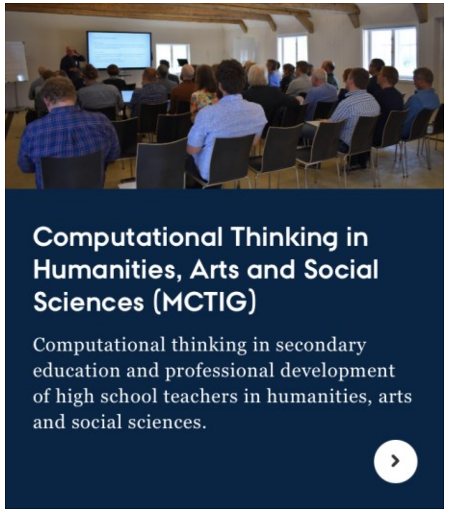
Density: 61 %: ~ 83 %

[81.5 ; 75.5 ; 84.5 ; 86.0 ; 85.8 ; ...]



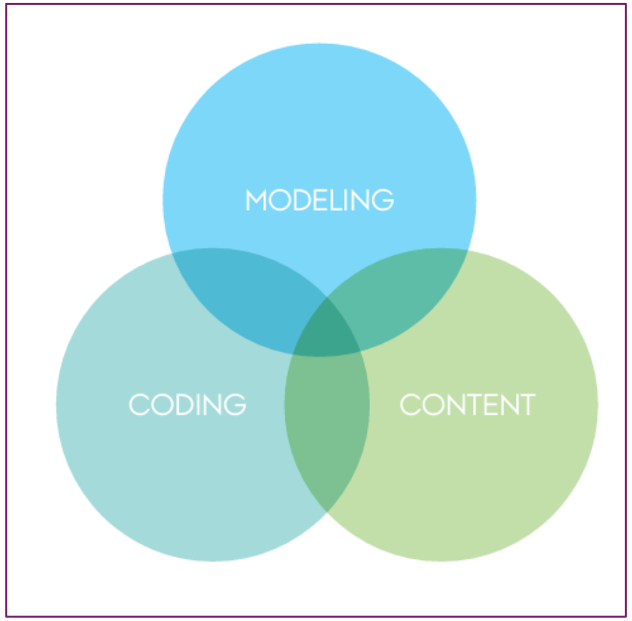
Informatics in other subjects in high school

– some projects since 2017

2017	2018-2020	2018-2022	2019-
			
CT in high school subjects	Modelling and CT in high school subjects	CT in math and natural science subjects	Digital empowerment in high school subjects
1,5 mio. DKK	2,2 mio. DKK	7,75 mio. DKK	0,6 mio. DKK

An additional ABM project in 2025-2026

Students learn using ...



The CMC Approach: Teaching activities integrating

- Content (subject matter, e.g. Biology)
- Modelling
- Coding

https://youtu.be/9R9r5w2Gr-w?si=WRQEj9KlThc7xeDm

Computational Thinking in the Danish High School: Learning Coding, Modeling, and Content Knowledge with NetLogo

Line Have Musaeus†
Center for Computational Thinking and Design
Aarhus University
Denmark
lh@cc.au.dk

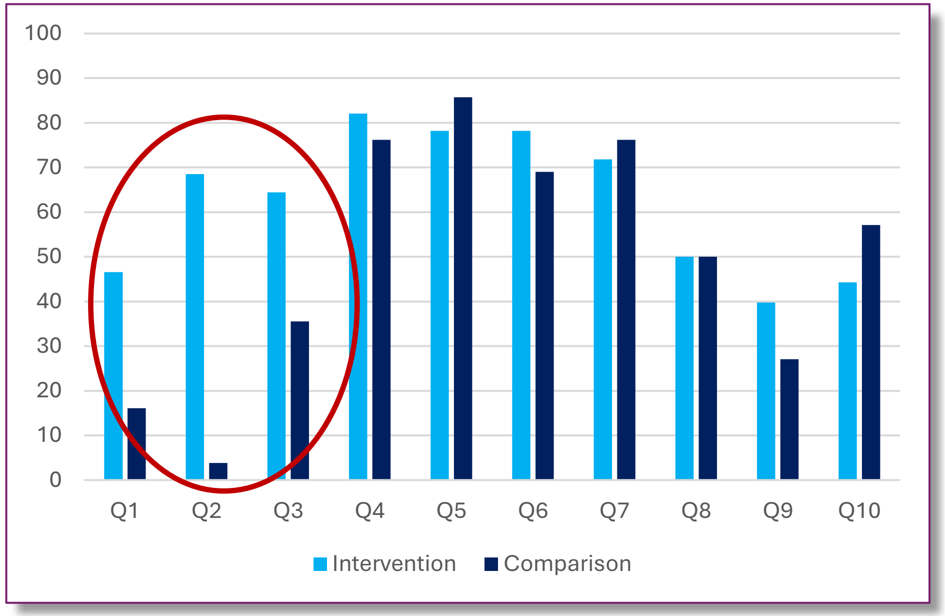
Peter Musaeus
Centre for Health Sciences Education
Aarhus University
Denmark
peter@cesu.au.dk

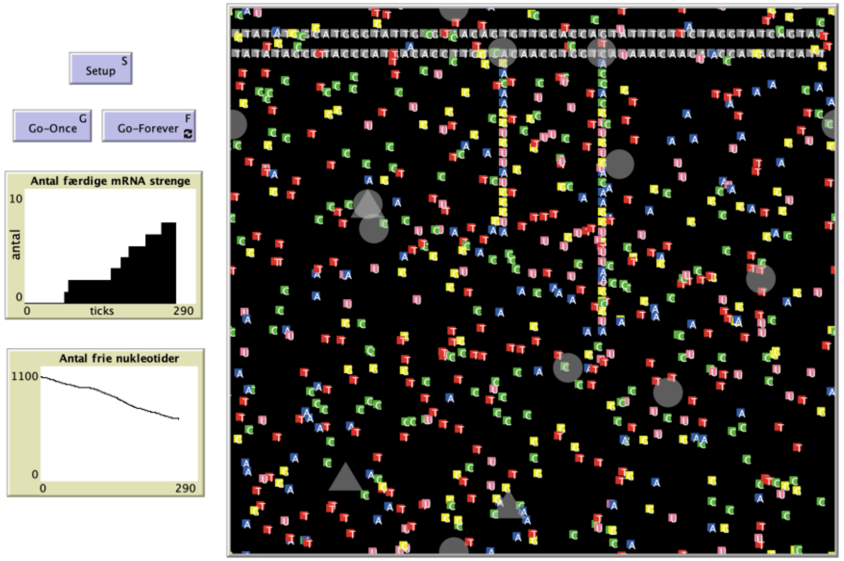
SIGCSE TS 2019

Students learn
both subject matter and computational modelling

- Agent-based models in Biology
- Tested by concepts inventory questions

Q1-3: Biology questions on randomness and variation – traditionally difficult topics





Developed by Palle Nowack PhD

JOURNAL OF BIOLOGICAL EDUCATION
https://doi.org/10.1080/00219266.2022.2118353

J. Bio. Ed. 2022

Routledge Taylor & Francis Group
Check for updates

Computational Modelling in High School Biology: A Teaching Intervention

Line Have Musaeus[a], Deborah Tatar[b] and Peter Musaeus[c]

[a]Center for Computational Thinking & Design, Aarhus University, Aarhus, Denmark; [b]Department of Computer Science, Virginia Tech, Blacksburg, VA, USA; [c]Centre for Educational Development, Aarhus University, Aarhus, Denmark

Students' transfer of learning

Intervention group participted in teaching activities in Math and Social Science
Comparison group participated in traditional teaching in Math and Social Science

Both groups participated in modelling tests unrelated to the teaching activities

Identify elements and simplifications in the model related to the real-life phenomenon.

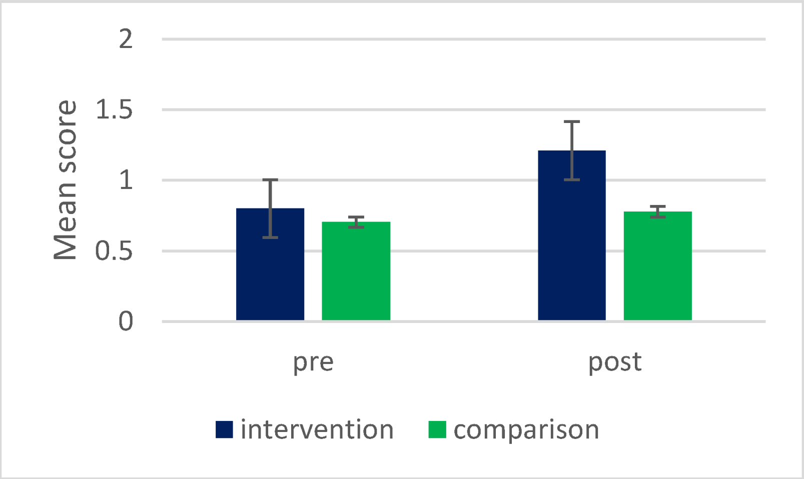


Figure 6. Mean scores of students' answers pre- and post-interventions.

Describe a procedure as a sequence of instructions related to a problem within a specific computer model.

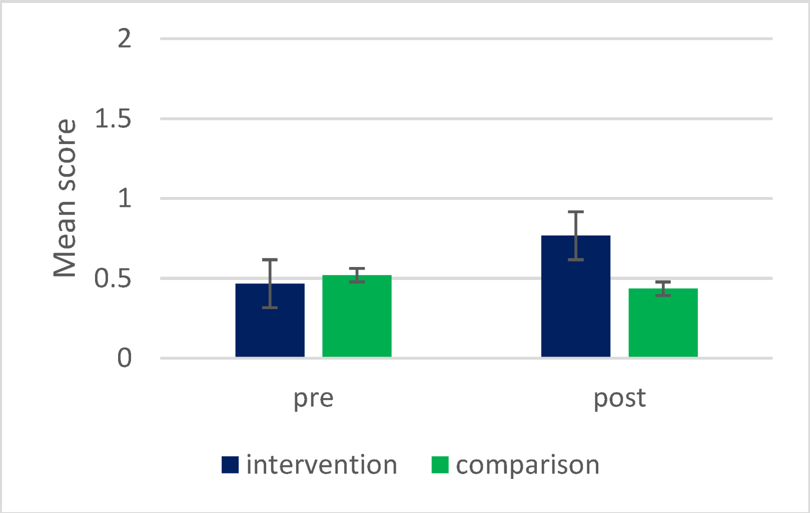


Figure 7. Mean scores of students' instruction sequence descriptions: pre- and post-interventions.

Musaeus, L. H., & Musaeus, P. (2024). Computational Thinking and Modeling: A Quasi-Experimental Study of Learning Transfer. *Education Sciences*, *14*(9), 980.

Professional development
for high school teachers

- Modelling in Math, Science, and Social Science
- Using the CMC approach
- Educating in-service high school teachers: three runs of the course (2019, 2021, 2022), a total of 120 teachers representing 44 high schools, and more than 2000 students.
- 84 teaching units available at: https://graspit.dk/
- During the course, a template for designing teaching activities in modelling was developed by design-based research in collaboration with teachers.

Supported by CCTD at Aarhus University, Villum Foundation, and a collaboration of high schools with focus on STEM education

CENTER FOR COMPUTATIONAL THINKING AND DESIGN AARHUS UNIVERSITET
DASG Danske Science Gymnasier
VILLUM FONDEN

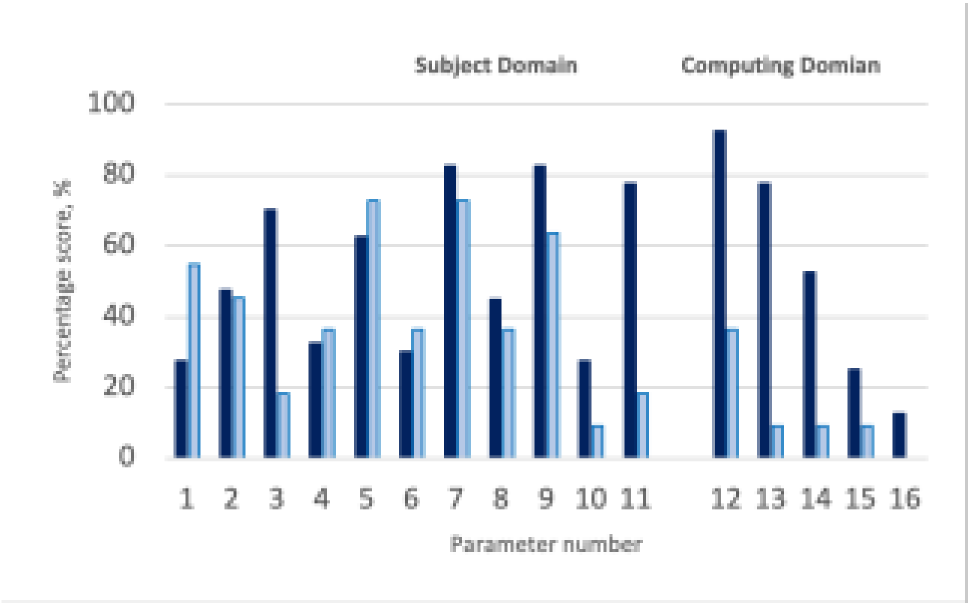
A Template
for Teaching Computational Modelling in High School

Table 4: Template with modelling parameters versus taxonomical levels of didactical questions (posed by teachers to students) in teaching activities. [] authors comments.

Taxonomical levels of questions	Low	Medium	High
1 Range of the model	Can the model show temperature?	What can the model predict?	Discuss whether the opportunities to influence input and output are realistic.

Table 3: Modelling parameters in template. Subject domain modelling (parameters 1-11) and computer domain modelling parameters 12-16)

Number	Modelling parameter
1	Range of the model
2	Explanatory power of the model
3	Causality between the elements in the model
4	Behaviour of the model
5	Elements included in the model
6	Elements not included in the model
7	How the model relates to a real-world phenomenon
8	Formulation of hypothesis that can be tested by the model
9	Results from repeated execution of the model
10	Emergent phenomenon developing over time
11	Use of and interaction with the model
12	Changing a value or variable
13	Adding a variable
14	Changing a procedure
15	Adding a procedure
16	Creating a new model



Percentage of teaching activities relating to each parameter in the 'Prog+' group (dark) and the 'Prog-' group (light)

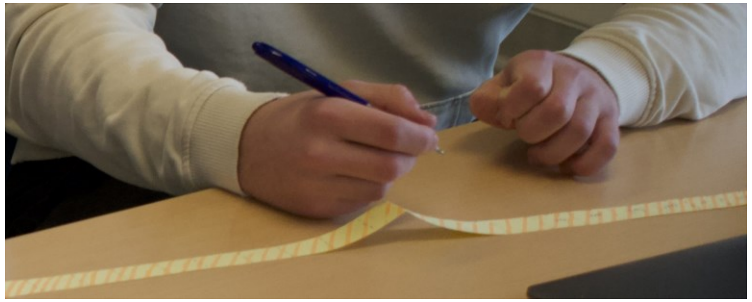
Musaeus, L. H., Caspersen, M. E., & Musaeus, P. (2023, September). A Template for Teaching Computational Modelling in High School. In *Proceedings of the 18th WiPSCE Conference on Primary and Secondary Computing Education Research* (pp. 1-10).

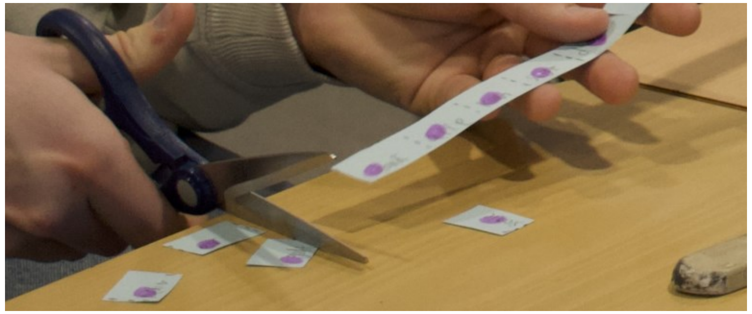
Modelling
in other subjects

Data-driven computing pedagogical approach

Inspired by Morales-Navarro & Kafai and Tedre et al.

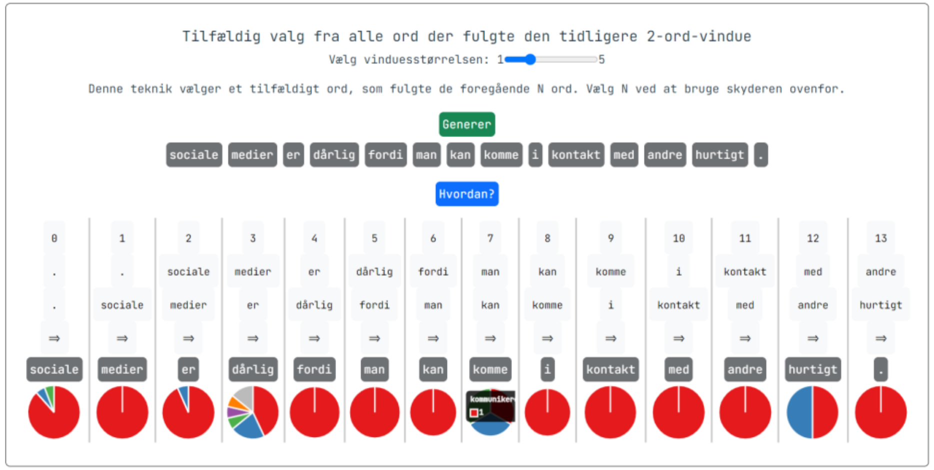
1. step: NLP hands-on teaching activities





‘Bag of words’

Developed by Luke John Conelly



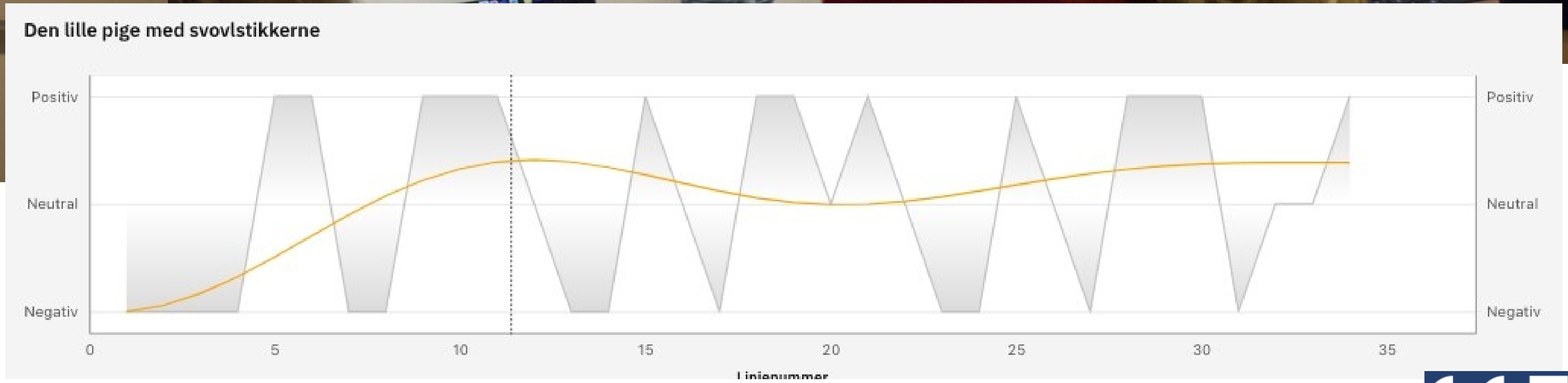
Luis Morales-Navarro and Yasmin B Kafai. 2024. Unpacking Approaches to Learning and Teaching Machine Learning in K-12 Education: Transparency, Ethics, and Design Activities. In Proceedings of the 19th WiPSCE Conference on Primary and Secondary Computing Education Research. 1–10.

Matti Tedre, Peter Denning, and Tapani Toivonen. 2021. CT 2.0. In Proceedings of the 21st Koli Calling International Conference on Computing Education Research . 1–8.

Machine learning in Danish

2. step: Fine-tuning of a LLM

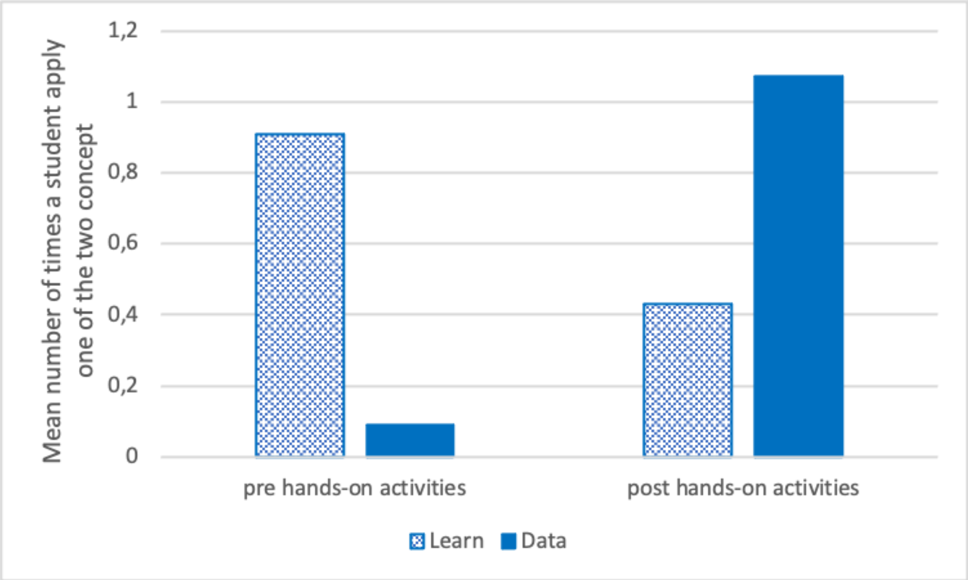
Sentences from a HC Andersen fairytale are categorised by the students as negative, positive, or neutral.



Developed by Karl-Emil Bilstrup
& Martin Thomsen Østergaard

Students’ understanding of ML

Template used to measure students’ knowledge of technical concepts and traning a model



Students’ use of ‘learn’ and ‘data’ in their answers to template questions.

Specification of ’data’ in answers to open-ended question	Comparison	Intervention
Pre intervention	0.2	0.1
Post intervention	0.0	1.1

Table 1: The average number of times a student specified the concept of ’data’ when answering an open-ended question in the online questionnaire.

Intervention and comparison group: Students’ use of ‘data’ in their answers to template questions

The results indicate that students already have an everyday understanding of ML in their everyday life. The students all used concepts such as data, train, learning, and pattern recognition, before the activities. However, after the hands-on experiences, students showed a more nuanced use of the concept of ’data’ and ’learning’ in both their written and verbal communication. This was not seen in the comparison group.

Musaeus et al. (submitted). Teaching Tools and Activities for NLP and LLM in High School Informatics. ITiCSE2025.

Some resources

Epistemic programming

Key points

- Programming is not only a necessity in order to build new software products but also a tool for thinking, gaining and presenting (new) insights.
- Epistemic programming ...
  - will be an important literacy for communication in a digital and data-driven world,
  - allows a different introduction to programming and programming languages that might spark interest and motivation in such learners who would not be interested in learning programming per se,
  - allows a low threshold to include data science in the computer science classroom,
  - is not solely focused on teaching job-related skills but (like the photographer mentioned in the beginning of the chapter) to foster subjectivation and developing individual capacities to pursue one’s own interests and curiosity.

Epistemic Programming
by Sven Hüsing, Carsten Schulte, and Felix Winkelnkemper

Chapter 22 in
Sentance, Barendsen, Howard, and Schulte (Eds.),
Computer Science Education: Perspectives on teaching and learning in school,
Bloomsbury, 2023.

Agent-based modelling in education

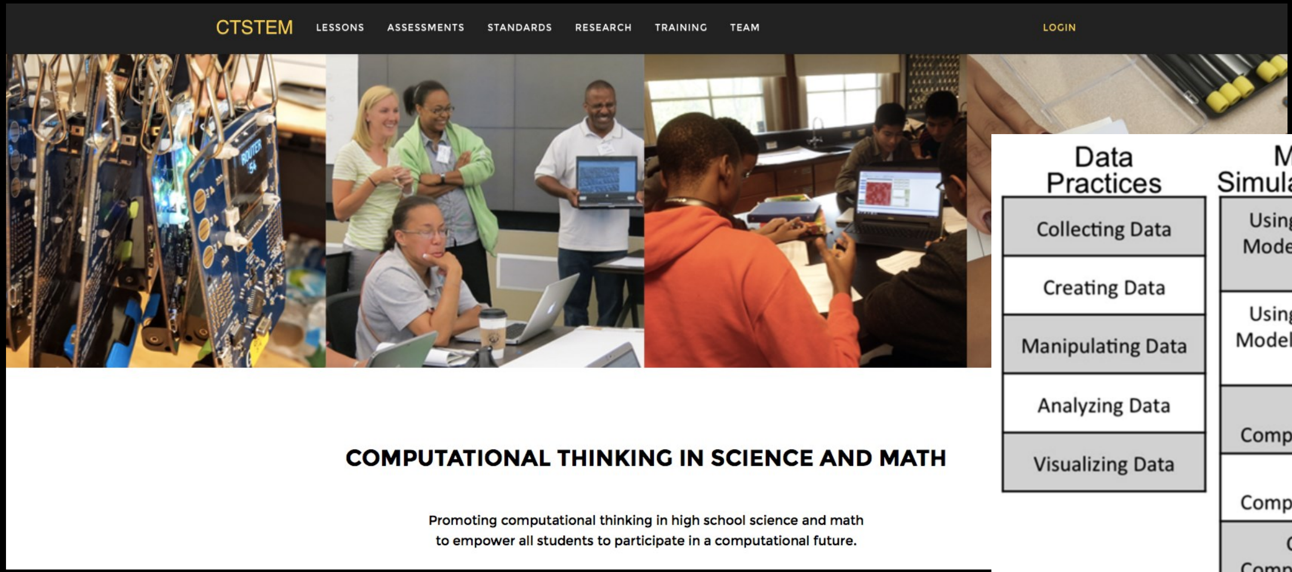
by assistant professor Arthur Hjorth, Aarhus University

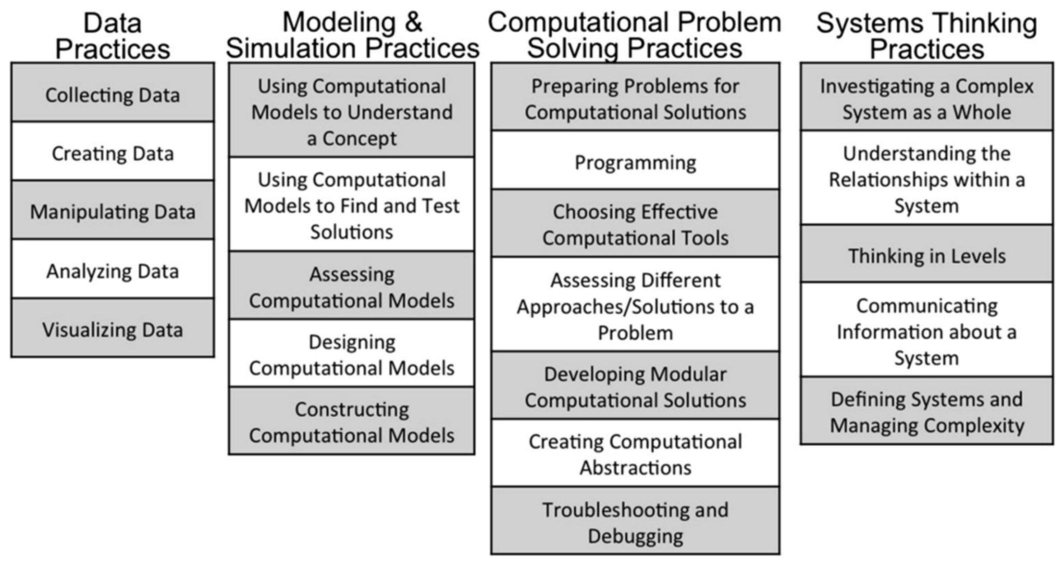
J Sci Educ Technol
DOI 10.1007/s10956-015-9581-5

CrossMark

Defining Computational Thinking for Mathematics and Science Classrooms

David Weintrop[1,2] · Elham Beheshti[3] · Michael Horn[1,2,3] · Kai Orton[1,2] · Kemi Jona[2,3] · Laura Trouille[5,6] · Uri Wilensky[1,2,3,4]





Some tendencies and opportunities

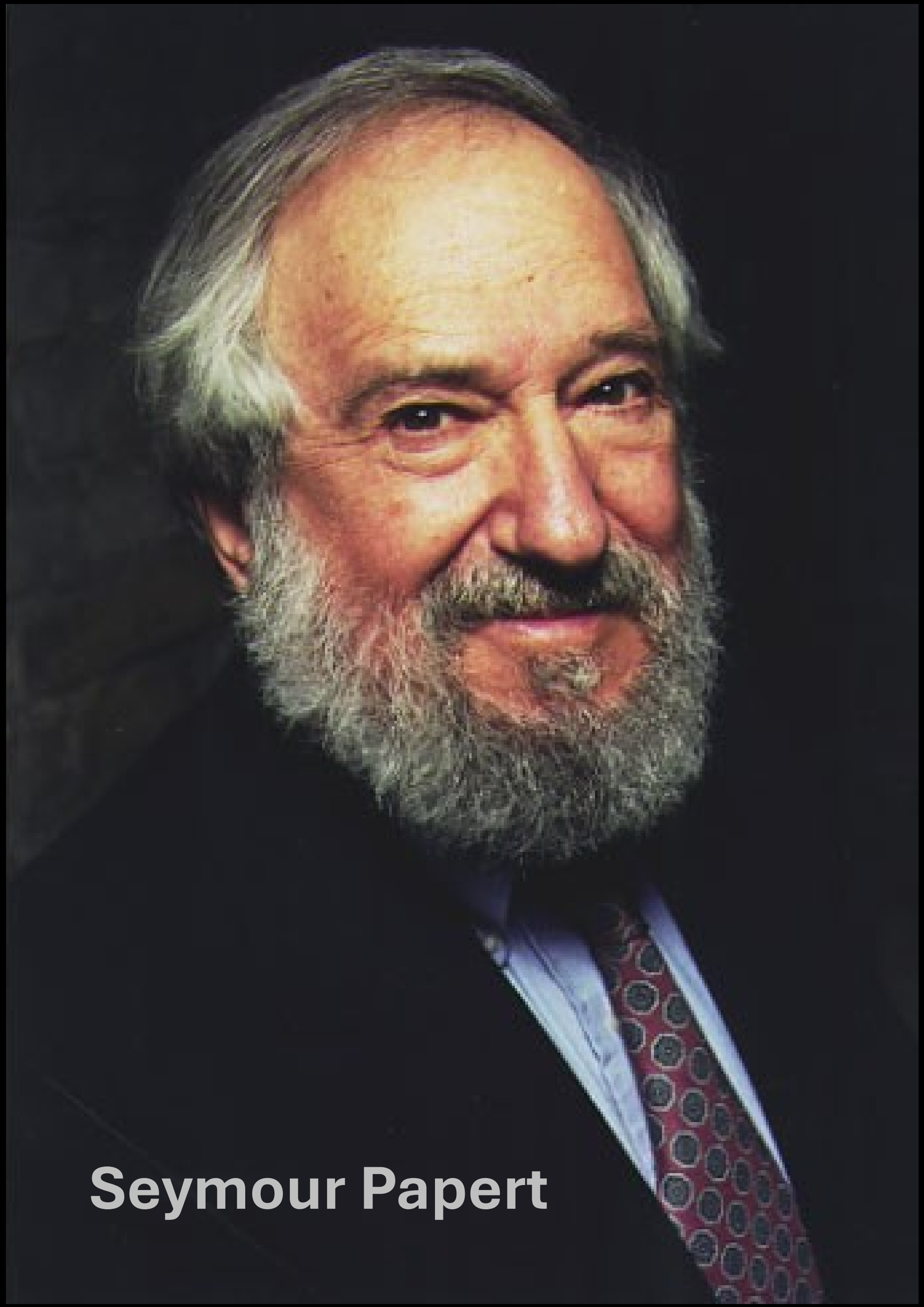
Two highly recommendable talks

Our success as a species rests not on our running speed, thick hide or acute sense of smell, but on our ability to communicate. We communicate through symbols – you did not drag a lion to the campfire to present the concept of 'lion'. We seem to have an unbounded ability to invent new systems of symbols. They could be called languages or more precisely (and boringly) modes of expression.

In this talk I will argue that IT opens the door to a profoundly new mode of expression. The novelty of what we here call computational thinking is on par with writing and mathematics. Through anecdotal evidence from previous revolutions, I try to sketch the enormity of the revolution facing us and end with a modest proposal to improve the chances of (net) success for our journey.

Computing – using computers to solve problems – is changing the practice of the natural sciences, humanities, and social sciences. However, most educational content does not yet reflect this change and the use of computing is currently sparsely integrated into educational curriculums.
This poses an exciting opportunity to rethink the contents of our educations and to integrate the use of computing as a natural component in the various disciplinary subjects. This opens for teaching students computational literacy and allowing them to develop computing as a natural part of their professional skill-set and identity.

We provide examples of how the integration of computing into science education programs has changed both the content and the form of the education. We will focus on how it provides students with opportunities to pursue realistic problems through authentic workflows, to participate in research early in their education, and to develop computational literacy through computational essays.
We provide examples of how similar principles can be integrated in social science, such as economics, and the humanities.



Seymour Papert on computational thinking

Computational thinking is
the use of programming
– as an extension of our mind –
to experience and understand the world,
to manipulate the world,
and to create things
that matter to us.

Seymour Papert (1980). Mindstorms: Children, computers and powerful ideas. New York, Basic Books, Inc. p. 9.