# An Agential Theory of Implemenation for Computer Science

– Nick Wiggershaus
PROGRAMme

Zürich 27/10/2021

HAPOC

savoirs - textes - langage
STL

Université de Lille

# Content

# 1. Introduction

# Main Issue

How are computer programs *physically* implemented in a concrete computing system?

# Goal

***Explicit*** formulation:

An Agential Theory of Implementation
for computer science

**Strategy:** (1) Connect discourse of Imp in CS & Prob of Imp.
(2) Use insights of scientific representation literature.
(3) Use “accessible” historical example

# 2. Brief Survey:
# Implementation ≠ Implementation?

# (A) Implementation in Computer Science

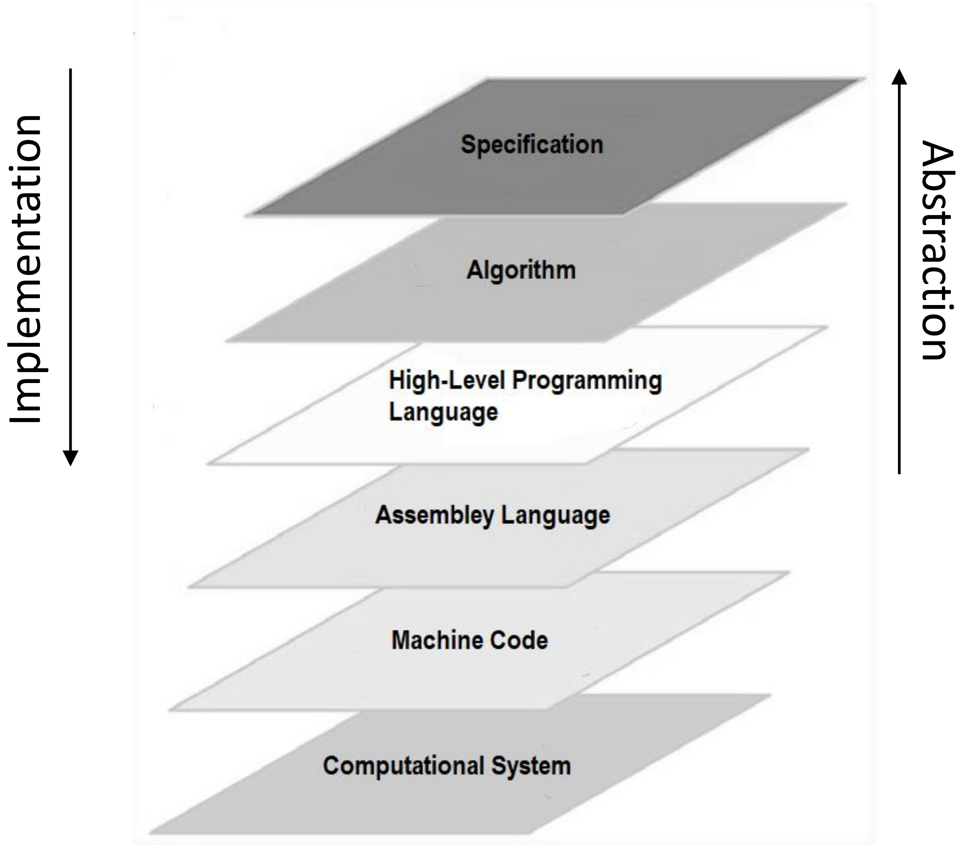
- **Rappaport (1999, 2005):** Something is an implementation of some syntactic domain *A* in medium *M* iff it is a **semantic** interpretation of a model of *A*.

- **Turner (2012, 2018):** function-ascription (external semantics) → Notion of Computational Artifacts. Specifications have **correctness-**jurisdiction over the artifact.

- Implementation as relation between specification (functional) and artifact (structural).

- **Primiero (2019): Correctness** & Implementation are coupled **–** Epistemology-Ontology Relation.

- Implementation is a *relation of instantiation* between *LoA* and *any higher one*

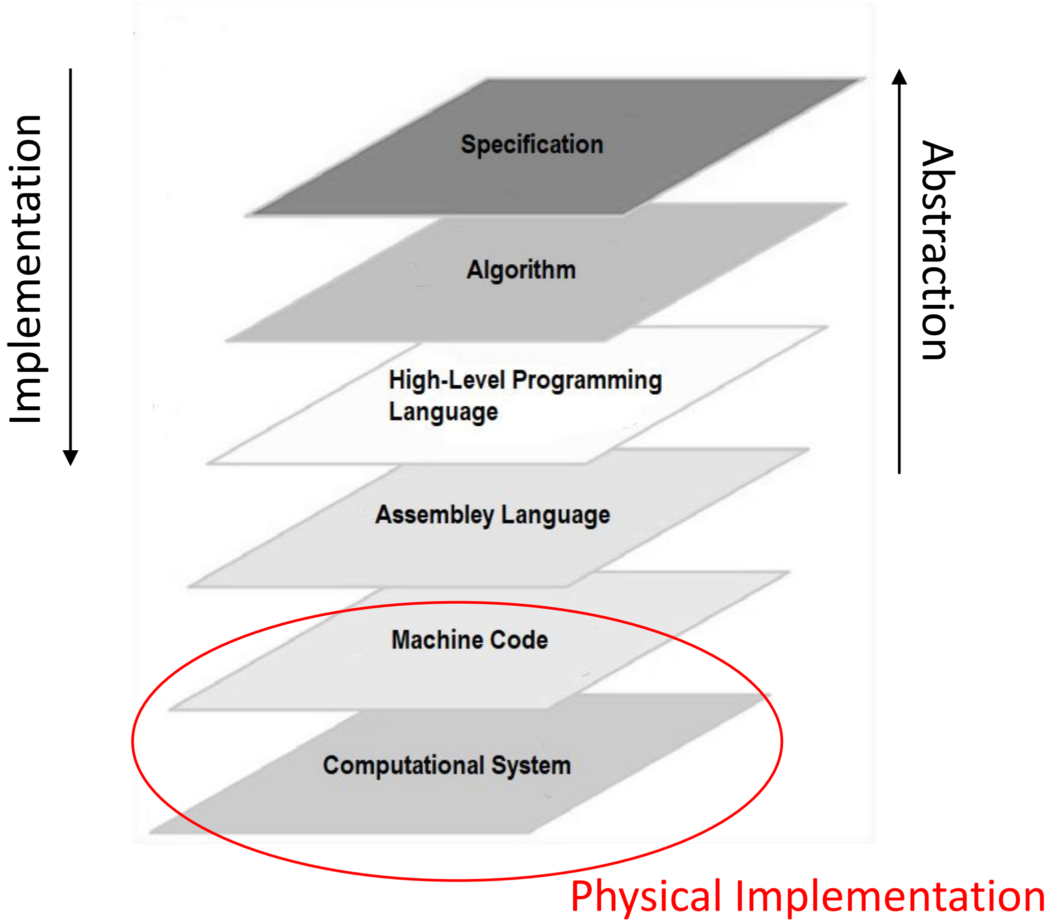


# (A) Implementation in Computer Science

- **Rappaport (1999, 2005):** Something is an implementation of some syntactic domain *A* in medium *M* iff it is a **semantic** interpretation of a model of *A*.

- **Turner (2012, 2018):** function-ascription (external semantics) → Notion of Computational Artifacts. Specifications have **correctness-**jurisdiction over the artifact.
- Implementation as relation between specification (functional) and artifact (structural).

- **Primiero (2019): Correctness** & Implementation are coupled **–** Epistemology-Ontology Relation.
- Implementation is a *relation of instantiation* between *LoA* and *any higher one*



# (B) The problem of Implementation

“Implementation may be defined as a relation between an abstractly defined computation and the **concrete physical process** that carries it out. […]. An account of implementation aims to **specify the conditions** under which a physical system performs a computation defined by a mathematical formalism – it is a theory of physical computation.”

(Ritchie&Piccinini 2018, 192-3)

# (B) The problem of Implementation

"Implementation may be defined as a relation between an abstractly defined computation and the **concrete physical process** that carries it out. […]. An account of implementation aims to **specify the conditions** under which a physical system performs a computation defined by a mathematical formalism – it is a theory of physical computation."

(Ritchie&Piccinini 2018, 192-3)

**(∃*f*) (*f*** is a mapping from $\mathcal{M}_{\mathrm{C}}$'s formal structure to $\mathcal{S}_{\mathrm{C}}$'s *causal* structure)

*& mechanistic?*

*& semantic*?

# (B) Theory of Concrete Computation

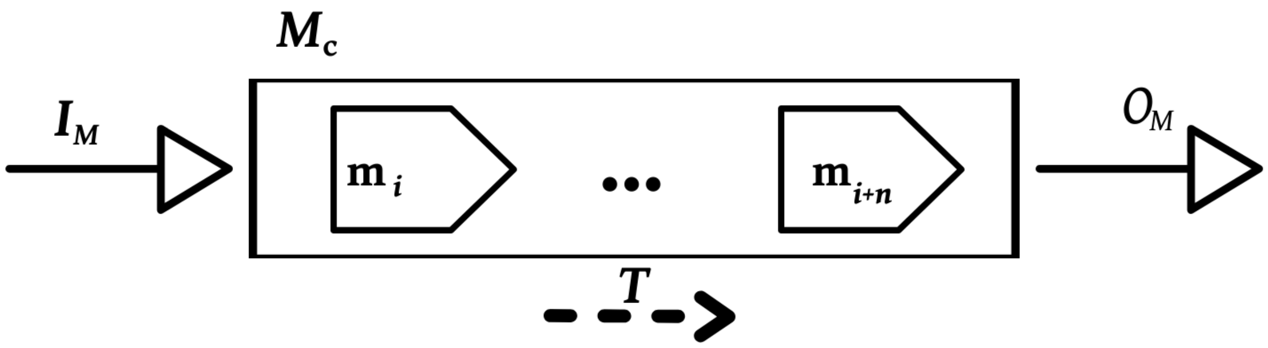
$M_C$ : Model of Computation

$m_i$ : Computational states

$T$ : Transition Function

$I_{M/S}$ : Input

$O_{M/S}$ : Output



# (B) Theory of Concrete Computation

$M_C$ : Model of Computation

$m_i$ : Computational states

$T$ : Transition Function

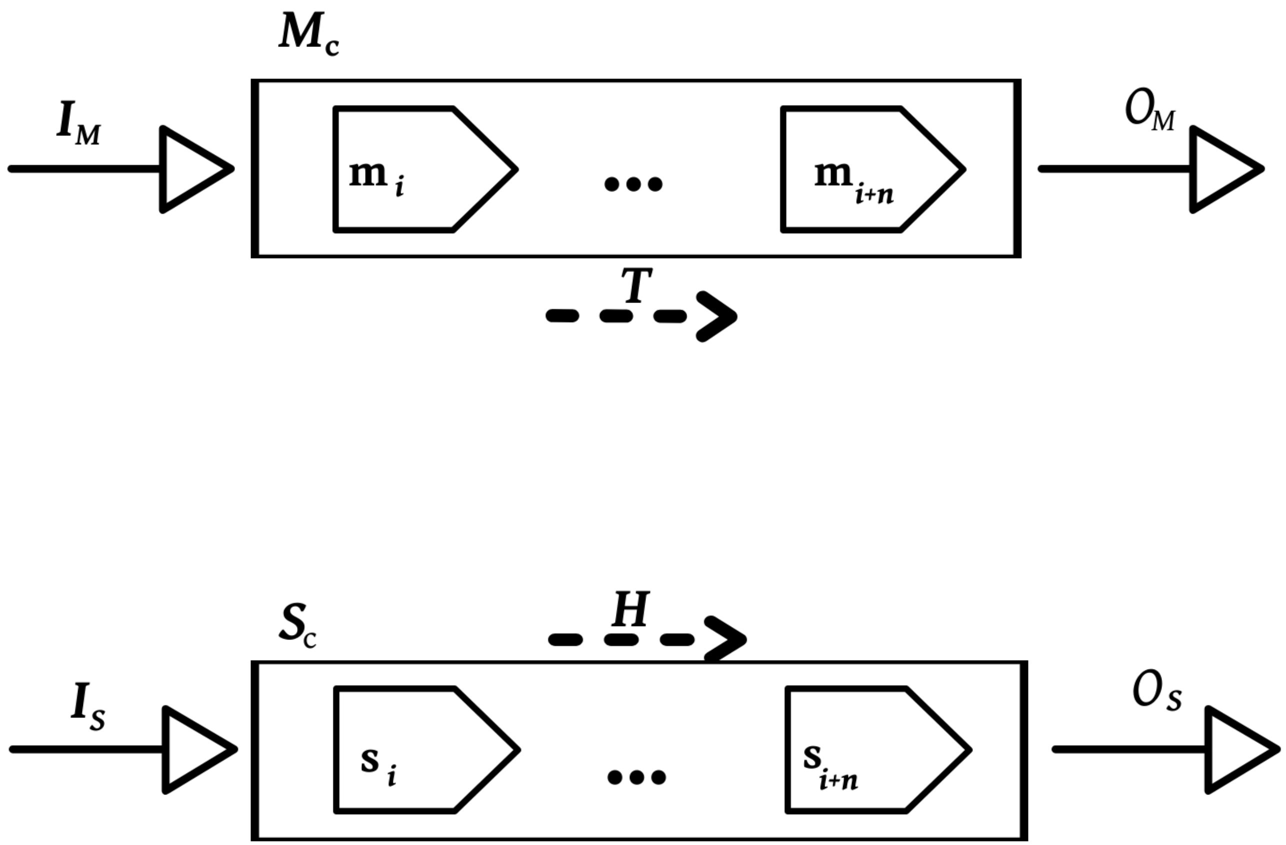
$I_{M/S}$ : Input

$O_{M/S}$: Output

$S_C$ : Physical System

$s_i$: Physical states

$H$ : Dynamics



# (B) Theory of Concrete Computation

$M_C$ : **Model of Computation**

$m_i$ : **Computational states**

$T$ : **Transition Function**

$I_{M/S}$ : **Input**

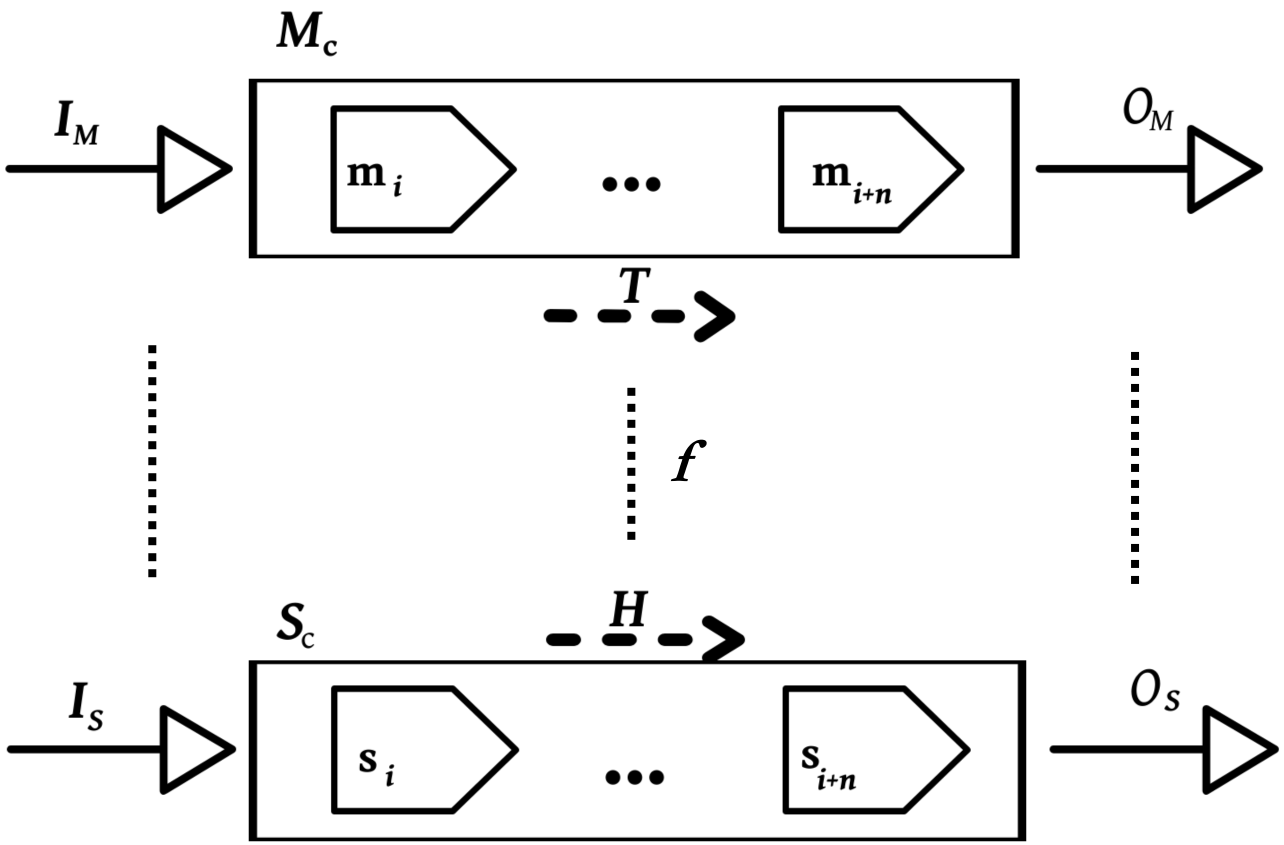
$O_{M/S}$: **Output**

$S_C$ : **Physical System**

$s_i$: **Physical states**

$H$ : **Dynamics**

$f$: “mapping relation”




(cf. Ladymen (2009), **Horseman et al.(2018**), Piccinini&Maley(2021), Scheutz(1999), Milkowski(2013), Pappayanoplous(2020))

# 3. Agential Theory of Implementation for computer science

# What’s *f*? Smells like Scientific Representation

Questioning metaphysical nature of *f* (cf. Suarez 2003: non-naturalization)

Scientific models *represent* a target system (*by means of an **agent***):

“*There is no representation except in the sense that some things are used, made, or taken to represent some things as thus or so.*” (van Frassen 2008, 23)

“S[cientist] uses X to represent [an aspect of the] W[orld] for purposes P.”
(Giere 2004, 743)

# What’s *f*? Smells like Scientific Representation

Questioning metaphysical nature of *f* (cf. Suarez 2003: non-naturalization)

Scientific models *represent* a target system (*by means of an **agent***):

“*There is no representation except in the sense that some things are used, made, or taken to represent some things as thus or so.*” (van Frassen 2008, 23)

“S[cientist] uses X to represent [an aspect of the] W[orld] for purposes P.”
(Giere 2004, 743)

**Intentionality**: Pragmatic/instrumentalist notion
**Directionality**: “One-way mapping”
**Misrepresentation:** accuracy; “correctness”
**Surrogative Reasoning:** “Predicting devices” (cf. Horseman et al. 2018)

# An agential theory of implementation

**(A)**

How to account for:

i. Human programmers & engineering practice
ii. Intention: Normative concepts, e.g., *correctness* (cf. Turner 2018, Primiero 2020)
iii. Direction "top-down"

**(B)**

Computation:

i. Tripartite structure
ii. Nature of *f*: Realism vs anti-realism

"**(A)** + **(B)** = Agential Theory of Implementation"

# Taking Stock

**Agential Theory of Implementation *for computer science****

**Three ingredients:**

a) **Mapping *f* as (non-deflationary) "scientific representation"**
(cf. Ladymen (2009); Timpson&Maroney(2018), Fletcher (2018), Curtis-Trudel (2020), Szangelies(2020), Papayannopoulus(2020))

b) **Computational system $S_C$ + mapping $f$ + $M_C$**

c) **Programmability**

***compatible with computational realist and anti-realist positions**

# 4. DEKI & MONIAC

# **MONIAC** (**M**onetary **N**ational **I**ncome **A**nalogue **C**omputer)

Aka: *Phillips-Newlyn Machine*; est. 1949.

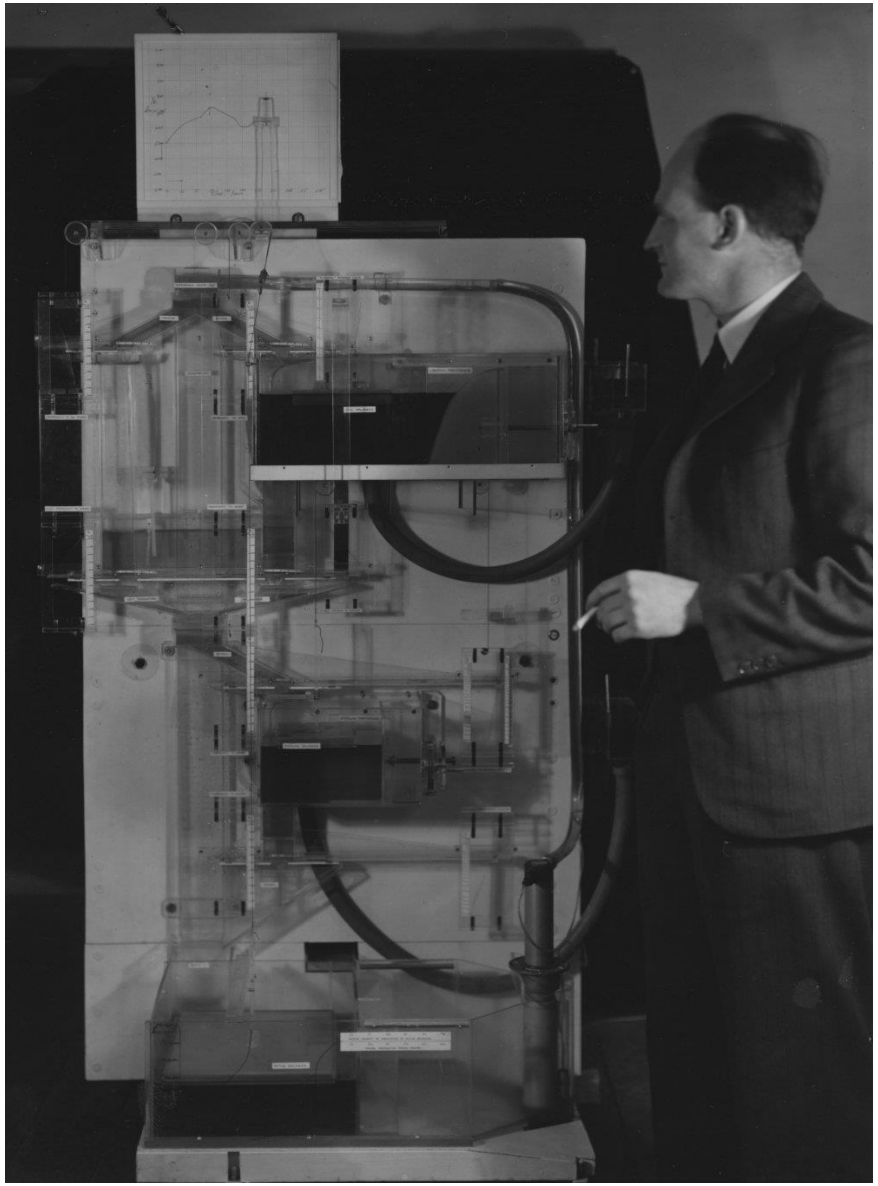
Hydraulic-analogue computer

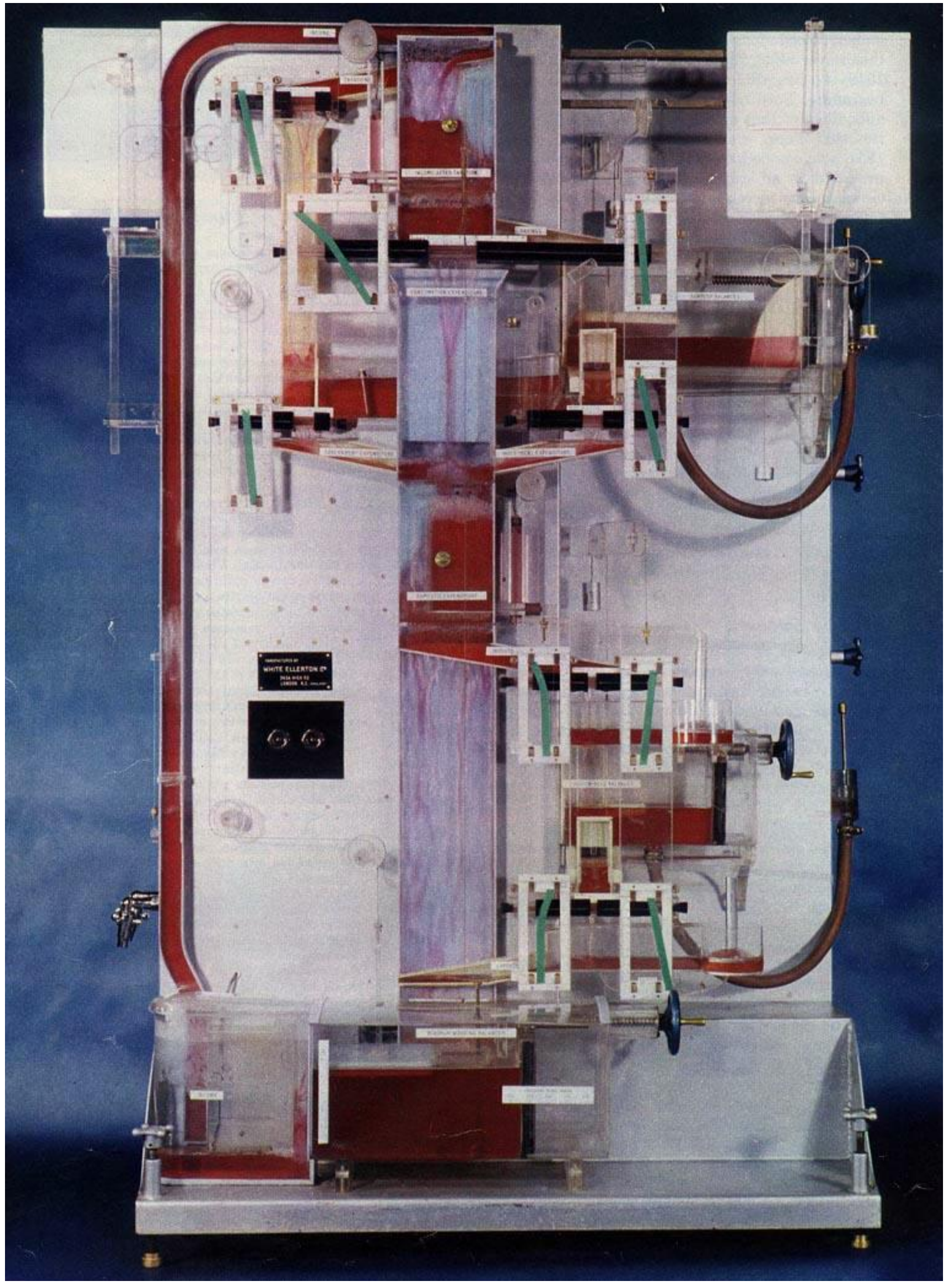
Modelling a national economy

“accessible” (historical) example

→ Connection with Frigg & Nguyen’s novel DEKI account

# MONIAC





# **DEKI** (**D**enotation, **E**xemplification, **K**eying-up, **I**mputation)

Scientific Representation account by Frigg and Nguyen (2018, 2020)

Let $C=\langle S_C, I\rangle$ be a computer, where $S_C$ is an object and $I$ an interpretation. Let $P$ be the program. $C$ ~~represents~~ implements $P$ as $Z$ iff all of the following conditions are satisfied:

# DEKI (Denotation, Exemplification, Keying-up, Imputation)

Scientific Representation account by Frigg and Nguyen (2018, 2020)

Let $C=\langle S_C, I\rangle$ be a computer, where $S_C$ is an object and $I$ an interpretation. Let $P$ be the program. $C$ ~~represents~~ implements $P$ as $Z$ iff all of the following conditions are satisfied:

*i.* $C$ denotes $P$.

# **DEKI** (**D**enotation, **E**xemplification, **K**eying-up, **I**mputation)

Scientific Representation account by Frigg and Nguyen (2018, 2020)

Let $C=\langle S_C, I\rangle$ be a computer, where $S_C$ is an object and $I$ an interpretation. Let $P$ be the program. $C$ ~~represents~~ implements $P$ as $Z$ iff all of the following conditions are satisfied:

*i.* $C$ denotes $P$.

*ii.* $C$ $I$-exemplifies $Z$-properties $Z_1,\ldots,Z_n$.

# DEKI (Denotation, Exemplification, Keying-up, Imputation)

Scientific Representation account by Frigg and Nguyen (2018, 2020)

Let $C=\langle S_C, I\rangle$ be a computer, where $S_C$ is an object and $I$ an interpretation. Let $P$ be the program. $C$ ~~represents~~ implements $P$ as $Z$ iff all of the following conditions are satisfied:

*i.* $C$ denotes $P$.

*ii.* $C$ $I$-exemplifies $Z$-properties $Z_1,\ldots,Z_n$.

*iii.* $C$ comes with key $K$ associating the set $\{Z_1,\ldots,Z_n\}$ with a (possibly identical) set of properties $\{Q_1,\ldots,Q_m\}$.

# **DEKI** (**D**enotation, **E**xemplification, **K**eying-up, **I**mputation)

Scientific Representation account by Frigg and Nguyen (2018, 2020)

Let $C=\langle S_C, I\rangle$ be a computer, where $S_C$ is an object and $I$ an interpretation. Let $P$ be the program. $C$ ~~represents~~ implements $P$ as $Z$ iff all of the following conditions are satisfied:

*i.* $C$ denotes $P$.

*ii.* $C$ $I$-exemplifies $Z$-properties $Z_1,\ldots,Z_n$.

*iii.* $C$ comes with key $K$ associating the set $\{Z_1,\ldots,Z_n\}$ with a (possibly identical) set of properties $\{Q_1,\ldots,Q_m\}$.

*iv.* $C$ imputes at least one of the properties $Q_1,\ldots,Q_m$ to $P$.

# Summary

- Fruitful relationship: Implementation & representation-as
- Compatible with computational realism and anti-realism
- Finer grained solution
- Many open questions

# Thanks

**Questions or feedback:**

**wiggershaus.nick@gmail.com**

# References


**(Fletcher, 2018)** Fletcher, S. C. (2018). Computers in abstraction/ representation theory. Minds and Machines, 28(3):445–463.

**(Frigg & Nguyen, 2018)** Frigg, R. and Nguyen, J. (2018). The turn of the valve: representing with material models. European Journal for Philosophy of Science, 8(2):205–224.

**(Frigg and Nguyen, 2020a)** Frigg, R. and Nguyen, J. (2020a). Modelling nature: An opinionated introduction to scientific representation. Springer.

**(Frigg and Nguyen, 2020b)** Frigg, R. and Nguyen, J. (2020b). Scientific representation. In Zalta, E. N., editor, The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, winter 2016 edition.

**(Giere, 2004)** Giere, R. N. (2004). How models are used to represent reality. Philosophy of Science, 71(5):742–752.

**(Horsman et al., 2018)** Horsman, D., Kendon, V., and Stepney, S. (2018). Abstraction/Representation Theory and the Natural Science of Computation, 127–150. Cambridge University Press.

**(Ritchie & Piccinini, 2018)** Ritchie, J.B. and Piccinini, G. (2018). Computational implementation. Routledge.

**(Ladyman, 2009)** Ladyman, J. (2009). What does it mean to say that a physical system implements a computation? Theoretical Computer Science, 410(4):376 – 383. Computational Paradigms from Nature.

**(Miłkowski, 2013)** Miłkowski, M. (2013). Explaining the computational mind. MIT Press.

**(Papayannopoulos, 2020)** Papayannopoulos, P. (2020). Computing and modelling: Analog vs. analogue. Studies in History and Philosophy of Science Part A, 83:103–120.

**(Piccinini and Maley, 2021)** Piccinini, G. and Maley, C. (2021). Computation in Physical Systems. In Zalta, E. N., editor, The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, Summer 2021 edition.

**(Primiero, 2019)** Primiero, G. (2019). On the Foundations of Computing. Oxford University Press.

**(Rapaport, 1999)** Rapaport, W. J. (1999). Implementation is semantic interpretation. The Monist, 82(1):109–130.

**(Rapaport, 2005)** Rapaport, W. J. (2005). Implementation is semantic interpretation: further thoughts. Journal of Experimental & Theoretical Artificial Intelligence, 17(4):385–417.

**(Scheutz, 1999)** Scheutz, M. (1999). When physical systems realize functions... Minds and Machines, 9(2):161–196.

**(Suárez, 2003)** Suárez, M. (2003). Scientific representation: Against similarity and isomorphism. International Studies in the Philosophy of Science, 17(3):225–244. 3

**(Turner, 2012)** Turner, R. (2012). Machines, chapter 4, pages 63–76.

**(Turner, 2018)** Turner, R. (2018). Computational Artifacts. Springer.

**(van Fraassen, 2008)** van Fraassen, B. C. (2008). Scientific Representation: Paradoxes of Perspective. Oxford University Press UK.

# *Appendix*

## DEKI

let $M = \langle X, I\rangle$ be a model. $M$ is an indirect epistemic representation of $T$ iff $M$ represents $T$ as $Z$, whereby $M$ represents $T$ as $Z$ iff all of the following conditions are satisfied:

*i.* $M$ denotes $T$ (and in some cases parts of $M$ denote parts of $T$).

*ii.* $M$ $I$-exemplifies $Z$-features $Z_1, \ldots, Z_m$.

*iii.* $M$ comes with a key $K$ associating the set $\{Z_1, \ldots, Z_m\}$ with a set of features $\{Q_1, \ldots, Q_l\}$: $K(\{Z_1, \ldots, Z_m\}) = \{Q_1, \ldots, Q_l\}$.

iv. (iv) $M$ imputes at least one of the features $Q_1, \ldots, Q_l$ to $T$.

## AToI for CS

Let $C=\langle S_C, I\rangle$ be a computer, where $S_C$ is an object and $I$ an interpretation. Let $P$ be the program. $C$ ~~represents~~ implements $P$ as $Z$ iff all of the following conditions are satisfied:

*i.* $C$ denotes $P$.

*ii.* $C$ $I$-exemplifies $Z$-properties $Z_1,\ldots,Z_n$.

*iii.* $C$ comes with key $K$ associating the set $\{Z_1,\ldots,Z_n\}$ with a (possibly identical) set of properties $\{Q_1,\ldots,Q_m\}$.

*iv.* $C$ imputes at least one of the properties $Q_1,\ldots,Q_m$ to $P$