Turner's conception of computer programs as technical artefacts: a critical assessment

Maarten Franssen, Delft University of Technology

Structure of the talk

1. Introduction
2. Dual-Nature view of technical artifacts
3. Critique of Turner's reading of Dual-Nature view
4. Critique of Turner's view on computational artifacts
5. Alternative view on computational artifacts

Dual character of computer programs

Moor: “[C]omputer programs can be understood on the physical level as well as the symbolic level.”

Irmak: “Many philosophers and computer scientists share the intuition that software has a dual nature.”

Indurkhya: “Software ... can be considered a mathematical object ... [but] also an empirical object.”

Colburn: “How can something, namely a computer program, be at once concrete and abstract?”

Dual-Nature view of technical artifacts

- Technical artefacts can be said to have a dual nature: they are (i) designed physical structures, which realize (ii) functions, which refer to human intentionality.
- Technical artefacts are thus conceptualized as physical and as functional objects.
- This … combines two fundamentally different ways of viewing our world: the physical and the intentional.

Kroes & Meijers, *SHPS* 2006, pp. 1-2.

Dual-Nature view of technical artifacts

- Technical artifacts have a physical as well as a functional description: they have a physical structure and they ‘are for’ something.
- But this doesn’t mean that if an object has a physical description as well as a functional description, it is a technical artifact.
- Social artifacts also have a physical structure and are for something – or at least need to be conceptualized intentionally.

Dual-Nature view of technical artifacts

artworks are social artifacts

Dual-Nature view of technical artifacts

… and so are banknotes

Dual-Nature view of technical artifacts

- *Technical* artifacts are for being *used* as physical instruments (for which two interfaces, to couple to an object and to a user, are required).
- *Technical* artifacts allow for being used according to their function *on the basis of their physical structure alone* (as long as the user has matching characteristics).
- *Technical* artifacts need to be given their structure through a design process that requires engineering expertise.

Dual-Nature view of technical artifacts

- “So, the kind of technical artefacts I am interested in may be characterized as human-made physical objects for solving practical problems.”
- “This means that software programs fall outside the scope of this book. I consider software programs to be ‘incomplete’ technical artefacts; only in combination with the appropriate hardware … are they able to fulfil their technical function.”

Kroes, *Technical artefacts*, 2012, p. 2.

Turner's reading of the Dual-Nature view

- "The following schematic summarizes this view of technical artifacts:

Function → *Design* → Structure → *Implementation* → Artifact.

This will be the perspective that we carry over to computational artifacts – the technical artifacts of computer science."

Turner, *Computational artifacts*, 2018, p. 27.

Turner's view of computational artifacts

- "*Computational artifacts* are the technical artifacts of computer science. *As such they have a function, a structure, and a physical existence*."
- "[*C*]*omputers*, *programs*, and *systems*, are representative of the major kinds of computational artifacts."
- "[T]he *implementation* is itself a computational artifact."
- "*Fabrication* is itself a technical artifact…"

Turner, *Computational artifacts*, 2018, pp. 41, 57, 50, 36.

Turner's view of computational artifacts

- "Via the semantics of their containing language, symbolic programs are *abstract* mathematical objects. ... However, programs are not purely mathematical entities. *As they are technical artifacts*, the mathematical nature of the symbolic entity has to be physically manifested. In this sense, the concept of a technical artifact provides the unifying notion for the symbolic and physical guises of programs."

Turner, *Computational artifacts*, 2018, p. 52 (*italics mine*).

Turner's view of computational artifacts

- "Our original duo of symbolic program and physical manifestation has been replaced by the trinity of specification, structure, and artifact:

Specification → *Design* → Symbolic program → *Implementation* → Physical process

Specification provides the function, a symbolic program is taken as the structural description, and the physical process is generated by the implementation."

Turner, *Computational artifacts*, 2018, p. 52 (*colours mine*).

Turner's view of computational artifacts

- "This provides a more robust and complete conceptualization of the ontology of programs. We shall call these *ontological bundles* program artifacts. These entities are *complex packages* of *abstract concepts* and *physical devices or computations* that are tied together via their correctness criteria."

Turner, *Computational artifacts*, 2018, p. 52 (*italics mine*).

Functional and structural description are **components** of 'program artifacts'.

Turner's view of computational artifacts

- "This provides a more robust and complete conceptualization of the ontology of programs. We shall call these *ontological bundles* program artifacts. These entities are *complex packages* of *abstract concepts* and *physical devices or computations* that are tied together via their correctness criteria."

Turner, *Computational artifacts*, 2018, p. 52 (*italics mine*).

If not complex, then the symbolic program (sequence of lines of code) and the physical 'program' (pattern of high & low voltages) can't describe same thing.

Dual character of computer programs

Moor: “[C]omputer programs can be understood on the physical level as well as the symbolic level.”

Irmak: “Many philosophers and computer scientists share the intuition that software has a dual nature.”

Indurkhya: “Software ... can be considered a mathematical object ... [but] also an empirical object.”

Colburn: “How can something, namely a computer program, be at once concrete and abstract?”

Dual character of computer programs

Moor: "[C]omputer programs can be understood on the physical level as well as the symbolic level."

Junger: "As I see it, the phrase 'computer program' has two quite distinct primary meanings. One meaning refers to the text of the program that is made up of signs and symbols ... The other meaning refers to the process that takes place inside a computer when the steps of the program are carried out."

My view of computers and programs

software side
programming languages

- programs present as *linguistic objects*;
- potential role for mathematical objects in semantics of these linguistic objects.

hardware side
computer as a *machine*

- *programming* present as an activity, physical manipulation of the machine.
- programs 'present' as *physical structure* of the machine;
- programs absent as *objects*.

My view of computers and programs

***software* side**

programming languages

- programs present as *linguistic objects*;
- potential role for mathematical objects in semantics of these linguistic objects.

philosophy of language

philosophy of mind

hardware side

computer as a *machine*

- *programming* present as an activity, physical manipulation of the machine.
- programs 'present' as *physical structure* of the machine;
- programs absent as *objects*.

My view of computers and programs

Are computer programs technical artifacts?

- Just like any linguistic object, they are:
  - material; artifactual; functional (they are for being used).
- *Not* just like any linguistic object, they are:
  - for a non-linguistic sort of use as well;
  - the product of expert design.
- But design directed at ***linguistic***, not *physical*, structure...
- ...Matching design directed at *physical* machine structure.