

Theses

The dissertation applies classical philosophical approaches to contemporary empirical material, the practice of software development. This involves the extensive reinterpretation, adaptation and critique of the classical conceptions. I think the most important original result of this work is the link itself that has been thus created between the two fields. My detailed theses are the following.

T1. The everyday practices of writing, deciphering, using, adapting, and appropriating software artifacts by users and developers can be understood as hermeneutic practices. I'm defining this expression in order to bring emphasis on the following meaning components of the concept "hermeneutic":

- a.) Hermeneutic practice is a process of interpretation that unfolds itself in skillful action rather than explicit, linguistic articulations. The resulting understanding does not mean holding and manipulating explicit assumptions or symbols, but rather the capability of orienting ourselves – coping – within a shared lifeworld.
- b.) All practical engagement involves a hermeneutic component, because it already involves perceiving the world as purposeful, as one that offers affordances, as one that can be translated into a field of possibilities for action and existence.
- c.) Hermeneutic practice is situated and embodied, that is, relative to the bodily-perceptual skills of the interpreter, and embedded in the material and social concreteness of the situation. The meaning of objects and symbols has to be reflected within the local context of other objects, social circumstances, skills, etc.
- d.) Hermeneutic practice presupposes having already taken up a position within a holistic horizon of meaning, constituted by a shared lifeworld and shared traditions of interpretation. Understanding even a single symbol of the code involves understanding a precisely non-delimitable set of other symbols, skills, traditions, and situations.
- e.) Understanding as a process is inevitably circular, because we don't have any other means to assess the relative value of our interpretation than to engage in further interpretation, based on whatever understanding we've already arrived at so far. In the case of interpersonal communication, this circularity takes the form of asking back and then reinterpreting what has been said so far; in the case of interpreting texts, it takes the form of taking a fragment that is already believed to be understood and then reassessing its relations within the totality of the text; and in the case of interpreting software, it takes the form of actively interacting and experimenting with it.

The relevance of the concept thus defined is supported in the dissertation by three case studies, comprising more than six elaborated cases, chosen from the field of open-source, commercial and military software development.

T2. Software, as a medium of hermeneutic practice, can determine the field of practical interpretations that can be carried out within it: the field of possibilities of understanding, experience and action for those who interact with it. Especially in the case of computer language, this field influences the potential dimensions of conceptual abstractions that can be applied during the interactions with the software.

T3. Software and programming language can constitute a field of power. By transforming the space of possible interactions that can be carried out with it, developers can restrict the multiplicity of its possible interpretations; they can inscribe preferred patterns of usage in the software as privileged interpretations. Software thus can be leveraged to maintain more traditional forms of power relationships, to normalize practices, and to serve as obligatory passage points (Latour, 1987), as the case studies about the Ada programming language and the Netscape/Mozilla project illustrate.

T4. During the process of user appropriation, software – just like texts and other artifacts – can always be subject to unexpected reinterpretations that challenge the preferred patterns of usage instilled by its developers. The Netscape/Mozilla case study in the dissertation emphasizes the significant role of open-source software development in these appropriation processes, both because the field of possible reinterpretations is wider, and also because these interpretations are argued and contested in front of public scrutiny.