RECIIS
Electronic Journal of Communication Information & Innovation in Health
[www.reciis.cict.fiocruz.br]
ISSN 1981-6286

Original Articles

Rethinking formalization processes in computerized systems: analyzing the co-evolution between software and organizational practices

DOI: 10.3395/reciis.v3i2.256en

João Porto de Albuquerque
School of Arts, Sciences and Humanities of São Paulo, São Paulo, Brazil
jporto@gmail.com

Abstract

Computerized systems are being increasingly used with the purpose of improving the treatment and integration of the medical area, computerizing work processes in the health sector. It is therefore necessary to obtain a broad comprehension of social and technical imbrications implied in the development and use of these systems, and to go beyond simplistic assumptions that benefits obtained with computerization result solely from the technology employed. To provide effective instruments for the achievement of this understanding, this work comments on theories and practices related to the concept of *formalization* in the context of development of computerized systems, examining their prerequisites and summarizing them in a reference chart called formalization axis. Supported by recent works carried out in the area of Science and Technology, a perspective starts to emerge on formalization processes as the association of different elements in social and technical networks, which then create a co-evolution between software and organizational practices.

Keywords

computerization; formalization; organizational practices; social and technical networks

Introduction

The past few years have witnessed a growing dissemination in the use of Information and Communication Technologies (ICT) in various segments and activities of the society, including the health sector. The introduction of ICT to support information systems in hospital is often justified by the need to improve the treatment and integration of data within the medical area, computerizing work processes in the health domain and generating gains in productivity and quality of health services (HEALTH, 2004). However, even though computerized systems are usually considered the vectors of innovation that causes the transformation of organizations, the level of efficiency of these systems after implementation not always lives up to the expectation surrounding them. Such disappointments are often assigned to the difficulty, lack of ability, or resistance of users when dealing with technology, in an asymmetric appreciation according to which the success obtained from the use of computerized systems result from characteristics intrinsic to the technology, whereas the failures result from "non-technical" social problems (TEIXEIRA et al., 2007; ZARAMA-VASQUEZ et al., 2008).

Surveys in the area of Science and Technology Studies[1] (STS), carried out in the last decades, have questioned this asymmetric explanation for the relationship between technological artifacts and organizational practices, in an attempt to explain the results produced by computerized systems in the health sector, as discussed by Zarama and Vinck (2008), not due to "technical" characteristics intrinsic to technology, or to the "non-technical" societal system, or even to a combination of "technical" and "non-technical" factors. These surveys attempted to attain a broader understanding of the social and technical imbrications in the development and use of computerized systems, through an analysis of practices carried out by the players involved in their *situated actions* (SUCHMAN, 2007), reclassifying the technological artifacts - particularly computerized systems – as heterogeneous networks that articulate both human and non-human entities (LATOUR, 2005). This perspective – broadened to simultaneously and indissociably encompass "technical" aqnd "non technical" aspects in one social and technical examination (CUKIERMAN et al., 2007) – is therefore necessary in order to achieve deeper understanding about the implications of the growing computerization of organizational processes in the health sector.

In consonance with this perspective, this paper intends to question the idea of *formalization*, often used in the development of ICT to indicate a process where descriptions of social practices are converted into computer software. Therefore, the study starts with an analysis of the usage of the term *formalization* (Section 2) followed by the examination of ontological prerequisites frequently assumed concerning informal practices and software artifacts, summarizing them in a reference chart called formalization axis (Section 3). I then suggest that this reference chart implies an aporethic discussion on ways to conceive the relationship between organizational practices and the software, defined by the former or by the latter (Section 4). The formalization axis chart is then compared to the theoretical reference of STS surveys (Section 5), serving as a base for an initial reclassification of formalization processes, which considers formalization to be an association of different elements in social and technical networks, thus generating a co-evolution between software and organizational practices (Section 6). The essay is concluded in Section 7.

The double meaning of formal

The term *formalization* is frequently used in the development of ICT, usually indicating a process where vague and inaccurate descriptions of procedures and practices of the "social world" are gradually converted into precise and executable computer models, and are ultimately implemented in computer software. In a logical and mathematical reading, the formalization involved in software development essentially consists of the expression of procedures using a certain *formalism* or *formal language*, that is, following a grammar with rigorous and explicitly defined syntax.[2] Software would be seen as the implementation of a finite sequence of symbolic manipulation operations (also called algorithm), that is, as a formal artifact whose functioning could be described, in its most abstract and finished form, as a deterministic symbolic machine.[3]

In other contexts, however, something is considered formal when it fits established norms or conventions – as opposed to informal, colloquial – such as, for instance, a set of written rules. This meaning is linked to the common distinction between the formal part of an organization – that is, its explicitly defined structures, rules, and procedures, usually by means of documents such as organization charts and standards – and the informal and trivial component of organizational practices. Actually, according to the *Houaiss Portuguese Dictionary* the verb *to formalize* means: "1. to create standardized rules, norms, models, procedures; 2. to execute in accordance with formulas, rules, habits, etc.; to sanction" (HOUAISS et al., 2001). This second definition given by the dictionary also highlights the *normative* characteristic of what is formal, that is, by saying that a certain attitude is formal, one implicitly states that this is the correct attitude, sanctioned by its submission to the norms of a social group. To formalize is, therefore, to always "differentiate between what is legal and what is illegal" (BOWERS, 1992, p.243).

Even though the formalization inherent to the process of software development and its utilization in organizations is frequently conceived as a technical transformation that eliminates the inaccuracies of the social world. This double meaning of *formal* implies that this process always includes dimensions that do not fit the classification of software as a "purely" technical artifact, and spread through social relations and practices. In order to examine these broader implications of formalization processes, the next section analyzes the frequently assumed prerequisites about software and organizational practices.

The formalization axis

Berg (1997) distinguishes two common discourses about the relation between computerized systems and work practices in organizations that, despite being admittedly stereotypical, can be a good starting point. One standpoint is supported by a group Susan Leigh Star calls "naive formalists" (*apud* BERG, 1997, p.405). For that group, software are formal symbolic processing artifacts whose intrinsic quality is to behave according to logical procedures accurately specified (that is, algorithms), which makes them essentially superior to the vague informal procedures of the social world[4]. The formal model, according to this viewpoint, captures – thanks to its inherent levels of abstraction and rationality – the *essence* of the elements of the social world it represents. Still, according to this opinion, the formalization process is understood as a technical purification from vague to accurate (AGRE, 1992), in such a way that the main problems faced in software construction revolve around the work necessary to eliminate the ambiguities and inaccuracies of the procedures of an organization, representing them with symbols and defining operations that must be executed by formal artifacts involving these symbols. After its development, a software could be profitably used in the organization to "streamline" working practices. Due to his belief that formal logical structures are intrinsically superior to informal structures, and due to the assumption that the structure of the world can be essentially described in mathematical terms – or, to use Galileo's metaphor: "the book of nature is written in the language of mathematics", this theoretical standpoint is linked to the philosophical tradition of positivism.

At the other extreme, we can find the theoretical standpoint that I call "saviors of the world of life" (due to reasons I expect will become clear below). Those who advocate this viewpoint assume that formal artifacts are nothing more than a fundamentally impoverished version when compared with the wealth of the empirical world. Formal artifacts that model working practices based on rules or formulas, therefore, can only offer a coarsely incomplete and superficial draft of the social world they represent, resulting in inflexibility to the organization when applied to specific situations. By working in an inevitably rigid and impoverished way, formal artifacts – according to this perspective - would de-humanize and de-skill the work of those who are imprisoned in their instrumental rationality, in a taylorist nightmare (BERG, 1997, p.406). If these rigid artifacts eventually become more or less successful in their practice, "saviors" say that this fact will result from human flexibility and creativity in dealing with the hindrances of the "world of life" and thus compensate, with their versatility and interpretation skills, the rough limitations of formal artifacts.

Although these viewpoints are diametrically opposite concerning the importance assigned to formal artifacts, both are based on common assumptions worth explaining. The first assumption concerns the *essential* characteristics of formal artifacts: for both opinions, the *Formal* is a specific domain of reality, the symbolic realm of homogeneous abstraction, ruled by determinist laws that can be accurately and exhaustively described by means of mathematical formalisms. This is a point of agreement for both naïve formalists and saviors of the world of life. The two groups fiercely disagree about the importance that must be assigned to these characteristics of formal artifacts: for the former group, the *necessity* of the *Formal* world establishes the highest ideal of perfection to be attained, whereas the latter despises it as a limitation that must be overcome. The *Informal*, on the other hand, has characteristics that are complementary to those of its pair: it is the realm of the inaccuracy of the concrete world, of contingency, and of freedom of the human spirit. While formalists think this "mess" of the social domain must be harnessed and purified[5], "saviors" define themselves as the bastions of freedom and of human genius in their fight against the cold and mechanical technical rationality.

The idea underlying the two positions described above is that *Formal* and *Informal* refer to two distinct domains of reality – the 'social domain' and the 'technical domain', respectively – which are *essentially* and *fundamentally* different (Figure). The formalization process can, therefore, be understood as a one-dimensional displacement along an axis linking the social and technical domains. In this axis, intermediary points contain shapes composed of some formal and some informal elements; these two groups remain, however, always separate and juxtaposed, like immiscible liquids, as it were. Accordingly, as one shifts along the axis towards the ideal symbolic representation and determinist execution, a larger number of elements of the composition will belong to the theoretical domain; consequently the level of formalization grows.

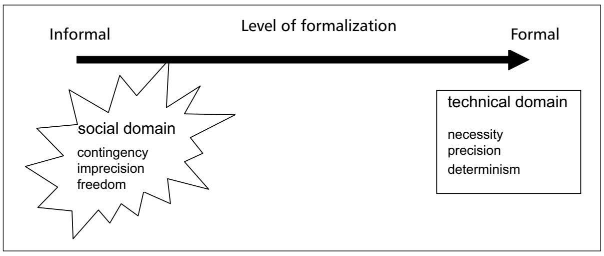

Figure - The formalization axis.

Given the panorama above, the core point of contention becomes *what* and *how much* must be formalized? For the advocates of the two stereotypical positions described above, answers are ready-made: for the former, formalization must be taken to the maximum extent possible, whereas for the latter, one must fight as much as possible against formalization attempts. In the past years, however, there has been a growing perception that these entrenched and extreme positions are not very useful in dealing with the complex interactions between formal artifacts ad informal practices. Nevertheless, even though radical extremists such as those have perhaps become more rare nowadays, the reference chart defined by the formalization axis characterized above still regulates, to a great extent, the decisions and discussions about the development and utilization of software in organizations.

Instead of arguing for or against the formalization of organizational practices (or even trying to define a "correct" level of formalization on the axis), we believe that the solution for the dilemmas described above can only be attained after a change in the ontological reference inherent to these issues, that is, we should disregard the reference cart defined by the formalization axis and search for new perspectives for the analysis of formalization processes. Fortunately it is possible to detect reflexes of this task in studies undertaken in the past few years in different areas, such as distributed cognition, Computer Supported Collaborative Work, and Science and Technology Studies. Based on these studies, we will now examine theoretical resources capable of opening up new possibilities for us to consider the relation between formal artifacts and social practices in organizations.

Software between technical and social determinants

In the previous discussion we were able to identify the emergency of the contrast between formal software or artifacts, on the one hand, and practices of the social or organizational world, on the other. This contrast is generated by the discourses revolving around the development and use of ICT – both by the developers themselves and by their critics. Employing a common term in the anthropological discourse, also used by Bower and Star (2000, p.299), these discourses *naturalized* the software object as an exclusively "Formal" artifact (understood as a self-contained domain of reality), turning into invisible the countless associations of persons and things necessary and implied in order to allow ICT to work under the model of the deterministic logic machine, such as: users "configured" to suit software expectations (WOOLGAR, 1991), or even the indispensable and complex electricity network, with its various human and material elements (HUGHES, 1983). To illustrate this point, to those who argue that the circuits of a computer are nothing but logical artifacts that 'due to their binary nature' could only assume the status of "0" or "1", it is worth mentioning Pirsig's observation (1984, p.329): when the computer is off, it assumes neither state, but a third possibility that cannot be described wither by "0" or by "1"[6] - which demonstrates the existence of implicit prerequisites when one thinks of the 'binary nature' of the digital computer.

Considering that the formal artifact is considered, in this naturalized view, as an object with specific properties and functions, defined at the time of its construction and independent of its later 'application' to contexts of specific usage, this position also corresponds to *technical determinism.* Even though few authors explicitly adopt this label, the technical determinism is usually associated with the various works, such as many in the technical software area, that are based on the assumption that a technology is capable of determining, due to its intrinsic characteristics, the way individuals and organizations use it. The relationship between technologies and organizational practices in these works is usually formulated (when this is done at all) as an "impact" caused by technology on an organization or on the society that employs them. Accordingly, although software developers have to deal in practice with different associations of heterogeneous elements in the construction and deployment of software in organizations – each with their own specific characteristics – achieving what Law (1992) calls "heterogeneous engineering" – for most of the members of the technical computing and informatics[7] community, the object software appears naturalized as a formal artifact or algorithmic logic machine, freed from the contingencies of its creation and of the situated nature of its meaning (BOWKER et al., 2000 p.299). This leaves them deprived of theoretical concepts and resources to question the challenges they face in practice. One of the goals of this paper is, therefore, to find resources to find the software not as a "pure technical object", but in a state of *unfamiliarity* that can soften the effects of the naturalized view and of the technical perspective, making visible the social and technical complexes in software production and use in organizational contexts. By highlighting the situated aspect and the contingencies involved in software utilization, however, it is necessary to avoid the risk of moving to the other extremity. The technology sociology work line known as *social constructivism*, for instance, is based on the correct assumption that technologies are inexorably linked to the interpretations bestowed on them by different social groups, and concludes – hastily, in our opinion – that it is these (sometimes conflicting) interpretations that *determine* the practical effects of technology, and not their technical characteristics. As a consequence, one technology would always possess "interpretative flexibility" (PINCH et al., 1987), that is, it would essentially be a social construction resulting from the interpretations of its users, instead of a mere reflex of the capacities of the machine (GRINT et al., 1997, p.10). The problem of this type of argument is that is still operates in the formalization axis shown in the Figure, only displacing the effect of a software – or, to use a term by Orlikowski (2000) "technology in practice" – to the opposite pole, inserting it in the "social domain". The technical determinism - which considers the functioning of the software simply as the update of its algorithmic logic machine power – is replaced by a social determinism that assigns full power and autonomy to users for them to arbitrarily redefine the technical artifact, which "could always be ignored, resisted, or remodeled to attain goals usually perceived as being linked to their implementation in specific situations", as incisively criticized by Kallinikos (2004, p.141). In practice, however, 'interpretative flexibility' is quite often limited and some aspects and behaviors of technologies appear to be considerably reticent about the desires of their users. On the other hand, totally denying the influence of formal artifacts on human behavior leaves us blind to the generative power these artifacts exert in practice.

In order not to incur in the extremes above, some authors advocate an intermediary position where the relation between technology and society (or between software and organization) is understood as an interaction between certain characteristics due to technical factors inherent to the technology and others originating in social interpretations (ROSE et al., 2005). This position would be equivalent, in our context, to locating organizational practices involving software utilization at some intermediary point between the technical and social domains in the formalization axis of the Figure, that is, to define the effects of software usage as a juxtaposition of some "technical" and other "social" elements, as previously observed. However, the core issue of the relation between software and organizational practices would still remain unanswered, for how would we be able to define which are the 'characteristics inherent to technology/software' without entering into an essentialist technical determinism, and symmetrically, how would we define those 'emerging characteristics of the social world/organization' without falling into a reductionist social determinism?

Questioning software ontology

The issues discussed in the previous section are similar to the controversy about the status of scientific facts between *realism* and *social constructivism* in the science studies (in short: are scientific facts discovered in nature, or invented by man?). Rouse (2002) argues that this conflict results from the supposition implicitly assumed by both sides of the discussion that nature and society comprise two stagnant and self-contained components of the world that interact between themselves. The basic assumption here is what Bruno Latour calls "modern constitution", that is, an implicitly assumed ontological separation between Nature *versus* Culture or Object *versus* Subject – or yet Technology *versus* Society, - which Latour (1993) characterizes as typical of modernity. Rouse (2002, p.63) compares this basic assumption to vampires, living-dead who continue to haunt our concepts and interpretations about nature, culture, and science. Thus, as a solution to the conflict, Rouse points toward several recent works that are based on the radical denial of the assumption that dichotomizes Nature / Society (definitely piercing the heart of this vampire with a pole), to reconsider the world as a "*complex field of discursive-material practices*" (ROUSE, 2002, p.77). Therefore, by looking at these discursive-material practices, it becomes possible to transcend essentialist boundaries *a priori* between nature and society, seeing the latter as the *result* of the stabilization of a set of practices. This perspective is subsumed by Rouse under the label of *post-constructivism* (ROUSE, 2002, p.69) which Latour (2005, p.88ss.) simply prefers to call constructivism, without the "social" qualifier that would preserve the dichotomy (also see BARAD, 2003 and WEHLING, 2006).

Following the line of these researches, we can conclude that the issue described above between social determinism and technical determinism cannot be solved productively, as it is based on the implicit assumption of the formalization axis characterized in the previous section, insisting on the dichotomic opposition *a priori* between software as non-contextual technical artifact on one side, and specific contexts and situations concerning its utilization on the other[8]. That is, incorporating the dichotomy Nature *versus* Society into the pair of opposites Formal versus Informal, and considering them as large incommensurable categories essentially different from one another. Having considered this assumption, the effects of software usage, a typical example of a hybrid between 'technical and social domains', are understood as a *mixture of two pure forms* (LATOUR, 1993, p.78), which must be separated so that one can discern what originates in the subject – that is, in the social domain of the user organization – from what originates in the object – or in the technical domain: in the software "nature". Thus, even the mildest opinions, which try to escape from extreme technical or social determinisms described above, maintain, however, a belief in the ontological abyss between the technical and social domains, hindering one from facing software construction and usage processes in all their dense complexity, and leading to the aporethic discussion about the effects resulting from the "nature" of the software and those resulting from the interpretation of social players.

In order to productively examine this issue we must therefore abandon the reference chart defined by the formalization axis, rejecting the essentialist Formal / Informal dichotomy, and redefining the *ontological status*[9] of software and at the same time of organizational practices of its use, in order to regard them neither as pure forms resulting respectively from the technical and social domains, nor as an aggregate of "pure" elements from these two domains. Therefore, our starting point will be analyzing a software always within its construction and utilization practices, as a hybrid at the same time formal and informal, or, as Latour (1993, p.51ss.), a *quasi*-object, *quasi*-subject, that is, a social and technical artifact located outside the formalization axis of the Figure, between and below two poles. The software becomes, therefore, a hybrid formal-informal *mediator* of variable ontology (LATOUR, 1993, p.85), that is, the technical/social and formal/informal adjectives are not seen as essential categories capable of providing us with ready explanations about the effects of software, but as the result of the stabilization of artifact development and utilization practices – results that require us to provide explanations.

As Pickering (1995) proposes, this viewpoint corresponds to the movement of a *representationist* perspective for an *enacting* language, which implies moving away from the perspective of technical artifacts (software) as mere impoverished representations of the social world in favor of the consideration of artifacts and their properties (for example, rigidity, formality, durability, determinism, etc.) as results continuously produced by means of the *enactment* of discursive-material practices. Therefore, the form, the meaning, and the properties of a software are not characteristics defined *a priori*, but resulting from the negotiation processes among all players (human and non-human) involved in its development and use, and must be regarded as precarious achievements (LAW, 1992) that must be continuously confirmed in the practices. If, on the one hand, software artifacts produced inscribe intentions, prescriptions of organizational forms, interests, and visions of the world – thus collaborating to the configuration of the social space where they are employed - ; on the other hand, the material-discursive practices in the utilization context associate software with elements different from those originally expected – thus changing the very sense and meaning assigned to it (AKRICH,1992).

Formalization as a multidimensional practice of association

Questioning software ontology as we did above also means abandoning our view of the formalization process as a one-dimensional movement along the axis of the Figure, adding the consideration of other dimensions involved in the practices of software construction and usage. The *multidimensional* process of formalization thus generates the simultaneous production of formal artifacts and of organizational practices (both formal and informal), in a movement of mutual co-definition that Bowker and Star (2000, p.82) call *convergence* and that, infused with the same spirit, I prefer to call the *co-evolution* of software and *organizational practices*, the latter encompassing not only the practices of the organization utilizing the software, but also those of its developers.

Within this perspective, we can consider the formalization axis of the Figure as the one-dimensional projection of multidimensional formalization processes. Therefore the *Formal* category – linked, as we saw above, to the algorithmic logical machine model in the context of software development – consists of the element that defines this one-dimensional projection, where the latter is a result of the stabilization of professional practices in the computing area and of its contingent historical constitution process, with strong bonds with both the polarized and militarized world of the cold war (EDWARDS, 2000) and with the modern style of thought (TEIXEIRA, 2006). What Suchman states about a hammer, however, is also true in the case of a software:

> "Although the durable materiality of the hammer supports the statement that it exists before and after the moments of its use, it is nonetheless clear that its status as a hammer rests on its incorporation into the practice of some form of carpentry." (SUCHMAN, 2007, p.21).

Although the software does not possess the evident "durable materiality" of a hammer, Suchman's statement can still be applied to our context: notwithstanding the 'durable characteristic' of a software behaving as an algorithmic machine for symbolic processing[10], the status of an ICT as a formal artifact depends on its incorporation into formalization practices in its development process. Accordingly, this is not about flatly denying the algorithmic or formal characteristic of a software, but of acknowledging the network of heterogeneous elements that needs to be woven – that is, associating individuals, concepts, professional programming and testing practices, infrastructure of material components, organizational procedures, etc. - for a software to be *enacted* according to the deterministic logical machine. The formalization process, understood in its complex multidimensionality, therefore consists in the establishment of a heterogeneous network of individuals and things around the software artifact, whose real effect is both the software performance within organizational practices, and the projection of these practices on the formalization axis as elements of the "social world", and of software as a deterministic machine in the "technical domain". These two moments are: the establishment of the heterogeneous network around the software and its projection on the formalization axis – correspond to what Latour (1993, p.80ss.) calls the "multiplication of hybrids" and "purification", thus allowing one to speak – in professional conversations on computing and informatics – of software as "pure" formal artifacts (that is, purified), whereas software production and utilization practices will actually work by associating complex heterogeneous networks (that is, proliferating hybrids).

Final considerations

This paper attempted to analyze the concept of *formalization* within the context of the development of computerized systems and their utilization in organizations. In the light of recent STS works, we could verify how the reference chart of the *formalization axis* is implicit in some of the recent discussions about the effects of software on health organizations, suggesting that the solution for the conflicts between social and technical determinism must be found outside that chart. Therefore, an understanding of the formalization process related to software has been outlined, as a multidimensional practice of association of heterogeneous elements, resulting in a movement of co-evolution between software and organizational practices.

This increased comprehension sheds light on dimensions which are not usually discussed in the technical areas of informatics, but usually treated as secondary and seen as unwelcome interferences. This makes these dimensions also be neglected in several projects of computerized systems in the health area which lean on a narrow technicist view about information and informatics in the health sector (cf. MORAES & GÓMEZ, 2007).

One of the important dimensions of the formalization processes thus unveiled concerns the "sanctioning" characteristic of formal artifacts mentioned above (Section 2): the production of software and its incorporation into organizational practices bring with them the "formalization" of certain practices, in the sense that they are made – if not official and sanctioned – at least preferential and facilitated[11]. By making these associations visible, therefore, I expect to contribute to the acquisition of analytical resources that will not only enable a critical appreciation of formalization practices in organizational contexts of the health sector, but also be useful in practical intervention in the sense that they build computerized systems that may effectively contribute to the quality of health services, taking into consideration the social and technical unfolding of their development and use.

Acknowledgements

To Professor Dr. Arno Rolf, from the University of Hamburg (Germany) for his support and encouragement; to Edouard Simon and to all the members of the *Mikropolis* network, for the fruitful discussions that largely contributed to the development of the ideas contained in this paper; and to the Alexander von Humboldt Foundation, for the fundamental financial support provided.

Notes

1. The Science and Technology Studies (STS) are synthetically presented in (HACKETT et al., 2007). For works more relevant to this article see, for instance: Akrich (1992), Callon (1986), Latour (1993, 1999, 2000, 2005), Law (1992); Law and Hassard (1999). For a summary of the various lines of thought about the relations between technology, innovation and society see Machado (2006).

2. There are actually several types of formal language, as defined and hierarchized by linguist Noam Chomsky (1956).

3. Indeed, the mathematical model known as the Turing machine (TURING, 1936) became the computing paradigm *par excellence,* although, as (SCHNIZEL, 2004) states, alternative computing models resulting from more recent innovations such as genetic algorithms and quantum computing, may challenge this core position assigned to the algorithmic processing of symbols.

4. Even though a large amount of technical works in the software area still fit within this classification, it is actually possible to refine it a little more and discern – as Eden (2007) does, different paradigms within the computing area, each one with a different ontological status for the software artifact. In greater accordance with sociology, MacKenzie (2001, p.299ss.) presents an excellent discussion about the different meanings of the term "proof" in different sub communities of the computing area, comprising what he calls "proof communities".

5. See Latour (1993) for the concept of purification within this context and the discussion of Section 4.

6. Pirsig evokes the Japanese tradition of zen buddhism by saying that the circuit turned off is in a state of "Mu", word used by zen masters to indicate that the question has been made incorrectly and thus cannot be answered by a "yes" or by a "no", and therefore means something like "withdraw the question" (PIRSIG, 1984, p.329).

7. There are, as usual, notable exceptions that already have a long tradition, such as, for instance, in the European computing community – Porto de Albuquerque (2007) – and with a growing representation in the Brazilian academia - Cukierman et al. (2007).

8. Please note that, in the second edition of the classic *Plans and situated actions,* Schuman (2007, p. 21) notes, with remarkable intellectual honesty, that the suggestion made in the first edition, saying that formal artifacts (plans, in this case) are somehow outside the action had the side effect of reinforcing the dichotomy between plan and execution, which exactly what should be questioned.

9. About ontological questionings proposed by recent works in STS see Mol, 2007.

10. As Law reminds us (1996) the same concept of "durable materiality" of an artifact can be put to test in the relational perspective of STS when the stability of relations that allow us to verify such durability changes, and then the very state of the artifact as such undergoes alterations.

11. In this sense the concept of *inscription* as proposed by Akrich (1992) is especially interesting. For an example in the context of business process modeling see Porto de Albuquerque and Christ (2007).

Bibliographic references

AGRE, P.E. Formalization as social project. **Quarterly Newsletter of the Laboratory of Comparative Human Cognition, v.**14, n.1, p.25-27, 1992.

AKRICH, M. The de-scription of technical objects. In: BIJKER, W.; LAW, J.(Ed.). **Shaping technology:**building society. Cambridge, London: The MIT Press; 1992. p.205-24.

BARAD, K. Posthumanist performativity: toward an understanding of how matter comes to matter. **Signs**, v.28, p.801-31, 2003. [doi: 10.1086/345321].

BERG, M. Of forms, containers, and the electronic medical record: some tools for a sociology of the formal. **Science, Technology & Human Values**, v.22, n.4, p.403-433, 1997. [doi: 10.1177/016224399702200401].

BOWERS, J. The politics of formalism. In: LEA, M. (Ed.). **Contexts of computer-mediated communication**. New York: Harvester Wheatsheaf, 1992. p.233-61.

BOWKER, G. C.; STAR, S. L. **Sorting things out**: classification and its consequences. Cambridge, MA: MIT Press, 2000.

CALLON, M. Some elements of a sociology of translation: domestication of the scallops and the fishermen of St. Brieuc Bay. In: LAW, J. (Ed.). **Power, action and belief**: a new sociology of knowledge. London: Routledge, 1986. p.196-223.

CHOMSKY, N. Three models for the description of language. **IRE Transactions on Information Theory,** v.2, p.113-124, 1956.

CUKIERMAN, H. et al. Guest editor's introduction: the challenge of a sociotechnical perspective to software engineering. **Scientia,** v.18, n.1, p.4-6, 2007.

EDEN, A.H. Three paradigms in computer science. **Minds and** Machines, v.17, n.2, p.135-167, 2007. [doi: 10.1007/s11023-007-9060-8].

EDWARDS, P. **The closed world**. Cambridge, MA: The MIT Press, 2000.

GRINT, K.; WOOLGAR, S. **The machine at work**. Cambridge: Polity Press, 1997.

HACKETT, E.J. et al. The handbook of science and technology studies. 3th. ed. Cambridge, MA: MIT Press, 2007.

HOUAISS, A.; VILLAR, M.S. **Dicionário houaiss da língua portuguesa**. Rio de Janeiro: Objetiva, 2001.

HUGHES, T.P. **Networks of power**: electrification in western society, 1880-1930. Baltimore: Johns Hopkins University Press, 1983.

KALLINIKOS, J. Farewell to Constructivism: technology and context-embedded action. In: AVGEROU, C.; CIBORRA, C.; LAND, F. (Ed.). **The social study of information and communication technologies**. Oxford, UK: Oxford University Press, 2004. p.140-161.

LATOUR, B. **We have never been modern**. Cambridge, MA: Harvard University Press, 1993.

LATOUR, B. **Pandora's hope:** essays on the reality of science studies. Cambridge, MA: Harvard University Press, 1999.

LATOUR, B. **Ciência em ação**: como seguir cientistas e engenheiros sociedade afora. São Paulo: Editora Unesp, 2000.

LATOUR, B. **Reassembling the social**: an introduction to actor-network theory. Oxford: Oxford University Press, 2005.

LAW, J. Notes on the theory of the actor-network: ordering, strategy and heterogeneity. **Systems Practice,** v.5, n.4, p.379-393, 1992. [doi: 10.1007/BF01059830].

LAW, J. **Organizing modernity.** London: 1996.

LAW, J.; HASSARD, J. **Actor network theory and after**. Oxford: Blackwell; Keele: Sociological Review, 1999.

MACHADO, C.J.S. As relações entre tecnologia, inovação e sociedade. **DataGramaZero**, v.7, n.1, 2006.

MACKENZIE, D. **Mechanizing proof**: computing, risk, and trust. Cambridge, MA: MIT Press, 2001.

MINISTÉRIO DA SAÚDE. **Política nacional de informação e informática em saúde**: proposta versão 2.0. Brasília: Ministério da Saúde, 2004. Available at: <http://bvsms.saude.gov.br/bvs/publicacoes/PoliticaInformacaoSaude29_03_2004.pdf>. Accessed: 30 Apr. 2009.

MOL, A. Política ontológica. Algumas idéias e várias perguntas. In: NUNES, J. A.; ROQUE, R. (Ed.). **Objectos impuros**: experiências em estudos sociais da ciência. Porto: Edições Afrontamento, 2007.

MORAES, I.H.S.; GÓMEZ, M.N.G. Informação e informática em saúde: caleidoscópio contemporâneo da saúde. **Ciência & Saúde Coletiva**, v.12, n.3, p.553-565, 2007. [doi: 10.1590/S1413-81232007000300002].

ORLIKOWSKI, W.J. Using technology and constituting structures: a practice lens for studying technology in organizations. **Organization Science,** v.11, n.4, p.404-428, 2000. [doi: 10.1287/orsc.11.4.404.14600].

PINCH, T.J.; BIJKER, W.E. The social construction of facts and artefacts: or how the sociology of science and sociology of technology might benefit each other. In: BIJKER, W.; HUGHES, P.; PINCHT, T. (Ed.). **The social construction of technological systems**. Cambridge: The MIT Press, 1987.

PIRSIG, R.M. **Zen and the art of motorcycle maintenance**. New York: Bantam Books, 1984.

PORTO DE ALBUQUERQUE, J. Aspectos sociotécnicos da computação: contextualizando o desenvolvimento de sistemas de computação com o modelo. **Mikropolis**, v.14, p.181-197, 2007.

PORTO DE ALBUQUERQUE, J.; CHRIST, M. Formal models, flexible processes? Lessons from a socio-technical analysis of business process modelling. **Scientia**, v.18, n.1, p.14-22, 2007.

ROSE, J.; JONES, M. The double dance of agency: a socio-theoretic accounts of how humans and machines interact. **Signs and Actions**, v.1, n.1, p.19-37, 2005.

ROUSE, J. Vampires: social constructivism, realism, and other philosophical undead. **History and Theory**, v.41, p.60-78, 2002. [doi: 10.1111/1468-2303.00191].

SCHINZEL, B. Computer science between symbolic representation and open construction. **Lecture Notes on Computer Science**, v.3075, p.59-76, 2004. [doi: 10.1007/b98383].

SUCHMAN, L. **Human-machine reconfigurations**: plans and situated actions. 2th. ed. New York, NY: Cambridge University Press, 2007.

TEIXEIRA, C. Algumas observações sobre os vínculos entre a engenharia de software e o pensamento moderno. In: WORKSHOP UM OLHAR SOCIOTÉCNICO SOBRE A ENGENHARIA DE SOFTWARE, 2., 2006, Vila Velha. **Anais**... Rio de Janeiro: PESC/COPPE UFRJ, 2006.

TEIXEIRA, C.; CUKIERMAN, H.L. Why do software process improvement projects fail? **Scientia,** v.18, n.1, p.24-32, 2007.

TURING, A.M. On computable numbers, with an application to the Entscheidungsproblem. **Proceedings of the London Mathematical Society**, v.2, n.42, p.230-265, 1936.

WEHLING, P. The situated materiality of scientific practices: postconstructivism - a new theoretical perspective in science studies? **Science Technology Innovation Studies**, n.1, p.81-100, 2006. Special Issue.

WOOLGAR, S. Configuring the user: the case of usability trials. In: LAW, J. (Ed.). **A sociology of monsters**: essays on power, technology and domination. London: Routledge, 1991.

ZARAMA-VASQUEZ, G.; VINCK, D. Por que a informatização funciona? Estratégias de compensação dos atores no caso da prescrição médica. **RECIIS,** v.2, n.1, p.19-28, 2008. [doi: 10.3395/reciis.v2i1.159pt].

About the author

João Porto de Albuquerque

João Porto de Albuquerque is a professor of Information Systems of the School of Arts, Sciences and Humanities of São Paulo (EACH/USP). And Research Fellow of the Alexander von Humboldt Foundation (Germany) and current coordinator of the Information Systems Management research group of EACH/USP. His studies focus on social and technical aspects of Information Systems, at an attempt to understand the movements of co-production of technology, organizational forms, society, and human behavior, in a dialogue with the Science and Technology Studies. He recently published *The Challenge of Transdisciplinarity in Information Systems Research: Towards an Integrative Platform* (in: *Information Systems Research Methods, Epistemology and Applications*, IGI, Hershey), *Notwendige und vorläufige Formalisierungslücken in Organisationen* (in Digitalisierung der Arbeitswelt, VS Verlag, Wiesbaden) and *Aspectos sociotécnicos da computação: contextualizando o desenvolvimento de sistemas de computação com o modelo Mikropolis* (Theoretical and Applied Informatics Magazine).