

## Efficient Worker or Reflective Practitioner? Competing Technical Rationalities of Media Software Tools

Ingrid Forsler and Julia Velkova, Södertörn University

Following processes of digitalisation, the material preconditions for media production in the media industries have been transformed. Software and computers have replaced most of the mechanical tools and parts of manual labour used for the creation of symbolic, cultural goods to be circulated in the global media markets. At the same time, the work of creating such symbolic goods represents a form of craft that needs to be exercised in close relations to, now, digital materials and tools.

Yet, conventional modes of production and distribution of media software have been based on the separation of technological production from technological use, and have resulted in software with overdetermined functionality and conventions about how media work is to be performed, producing what some craft theorists have described as “a workmanship of certainty” (Dormer 1997, p.141). Grounded in an anticipatory logic according to which programmers and technology designers envision what potential software users across diverse fields of practice may need (Frabetti 2015), the frameworks of production of media software have been thereby establishing hierarchies between different forms of knowledge – practical and theoretical and putting them at tension in relation to the dynamics media work.

This chapter discusses this tension through an analysis of three different production frameworks of software in the field of computer graphics media, and more specifically computer graphics animation. Using these frameworks as an example, we argue that each distinct framework of software production assumes a different way of knowing and understanding knowledge in relation to practice. Despite choosing a rather specialised field of media practice, we believe that the discussion that we develop is relevant for other fields of digital work, among which game development and digital print media production.

Our suggestion is that different production frameworks of software tools for computer graphics are underpinned by competing visions of the media practitioners<sup>1</sup> and their knowledge, as shown in the table below.

<FIGURE 7.1 ABOUT HERE>

---

1 The term *media practitioner* comes from Donald Schön’s knowledge theory, used in this chapter, and is chosen over related terms such as *creator* or *user*, to cover a wider range of practices and to emphasize media production as labour. However, we also occasionally employ the term *user* to note a division of media labour through specialization.

These competing visions, we argue, are not inevitable or singular effects of industrialisation and capitalism. Rather, we conceptualise them as a possible object of contestation and redefinition at the level of practice, by changing the epistemological orientation of technical systems in terms of their outlook towards the potential users, the media practitioners, thereby acting on the contradictions of technologies of digital work.

One of the reasons for choosing a seemingly marginal field of media production – that of computer graphics media – is related to its ubiquity both across diverse spheres of media work and everyday life. Irrespective of whether we turn on an electronic device such as a computer, a mobile telephone, a GPS navigator, a game, or TV, each time we see an advertisement, or just browse a magazine with housing interior or fashion we consume computer graphics. The multiple genres of computer graphics media – computer games, visual effects, user interfaces, digital comics, computer animations and simulations – have yielded some of the most financially worthy and aesthetically valued symbolic goods produced by the media industries and computer cultures in the past two decades, ranging from family entertainment films produced by studios such as Pixar; through computer games to user-produced animations as in the Machinima animation community.

The production of computer graphics media is heavily reliant on software, much of which emerged in the late 1990s. It coincided with a widespread enthusiasm about the democratic potential of new communication technologies, like the internet, and the affordability of the computer as provider of new possibilities for creative practice (e.g. Bolter & Grusin 1999). In the fields of web design, print, games and animation creation, software programs like Adobe Photoshop, Premiere, Illustrator and In-design, Macromedia Flash (now owned by Adobe) or Autodesk AutoCad, 3D Studio Max and Maya quickly became some of the key toolsets of work for an increasing number of media practitioners who aspire to engage with computer graphics in the global cultural industries. These new digital tools became popular with their relatively short learning curve which allowed new producers of media to create good looking content quickly and efficiently. In effect, an expanding pool of free-lancers aspiring for a more individualistic mode of work, as part of the new connectionist ‘spirit’ of capitalism (Boltanski & Chiapello 2007), could engage in producing content for broadcasting companies, advertising agencies and web design companies rapidly. While these developments could be interpreted as signs of larger processes of de-

professionalisation and democratisation of the mediascape<sup>2</sup>, we can also see signs of exclusion of media producers from certain forms of knowledge. These signs, we suggest, can be identified if we take a closer look at the technical rationality embedded in the production frameworks of such software, and particularly their epistemological orientation in relation to media practice.

### **Methodological considerations**

Before discussing these different epistemological orientations, we wish to foreground the methodological and analytical approach used hereforth. We choose to discuss three examples of distinct frameworks of software for computer graphics media that can be broadly defined as industrial – developed by companies that specialise in either software production or in Hollywood entertainment computer graphics media, or as user-driven – developed or initiated primarily by media practitioners rather than by industrial manufacturers. Our knowledge on the latter is illustrated by a case of the 3D animation software Blender, and is derived from a larger ethnographic project on user driven technologies for media production conducted in the period 2012-2017 (see Velkova 2016; Velkova and Jakobsson 2017). The empirical material collected in the course of this research featured qualitative interviews with 37 computer graphics practitioners engaged in work for the media industries but developing their own software tools for computer graphics, participant observation of computer graphics film production and an analysis of online and other relevant documents. In addition, we have performed one semi structured interview with a practitioner working with off the shelves software ('Anna', p. 9), specifically for this chapter.

Our discussion on the former, industrial driven frameworks of software production, is made through the cases of the production frameworks of commercial, off-the shelf software for computer graphics, and of in-house software development practiced by Pixar, a pioneer company in computer graphics entertainment. Our knowledge about these frameworks is based primarily on secondary literature from the fields of science and technology studies and anthropology that discusses computerization of work and the entrance of computer graphics in work practices (Henderson 1999, Downey 1998); histories of software and the computer graphics industry (Sito 2013; Tai 2012; Pfiffner 2003); insights gained from the interviews with media practitioners described above as well as with one of the co-founders of Autodesk; and finally, online videos and documentation released by companies such as Pixar and Autodesk. Our reliance on secondary literature for discussing these frameworks is provoked

---

2 The question of democratisation of media through digital technologies of production and communication has been for more than a decade now criticised by political economists through the debate on free labor and online user exploitation (e.g. Andrejevic, 2008; van Dijck, 2009). Whereas these debates are important, we have chosen to put them aside in this chapter, and instead focus on questions of epistemologies in relation to technologies of media work.

by the general scarcity of research in this field of practice, and particularly on industrially developed software for computer graphics production.

Throughout this chapter we use occasionally citations derived from all this research material, but they should not be seen as representative nor as an empirical basis of analysis out of which we derive our conclusions. Their task is to illustrate, clarify and support our theoretical argument rather than serve as a starting point for drawing conclusions.

Using these primary and secondary sources, we have applied feminist theory of technological development (Suchman 2002), and theories of knowledge in relation to practice (Schön 2003), to delineate different kinds of knowledge prioritized by the different frameworks. The contribution of this chapter is thereby a proposal for a conceptual distinction that outlines some of the contradictions that emerge between these three different frameworks of software development in terms of distinct visions about the kind of practitioners that are supposed to engage with these tools based on their epistemological outlook.

As shown in figure 7.1, we have chosen to term these visions as imagining media practitioners in the following way: ‘The media practitioners as an efficient worker’; ‘The media practitioners as a craftsperson in a fordist framework of labour’ and ‘The media practitioners as a master of tool development’. The concepts for describing knowledge that we use in the table are appropriated from the work of Donald Schön (2003) on the reflective practitioner, namely: knowing-in-action, reflection-on-action and reflection-in-action.

Each of the three software production frameworks, we suggest, exhibits a different technical rationality that arises on the axis between projected ideas of the end-user, their creative practice and different forms of knowledge. Our aim is not to promote one production framework over another, but rather to highlight different epistemological outlooks of technology that emerge at the intersection between production modes and actual practice.

### **Technical rationality and material knowledge in practice**

Following broader processes of disjuncture (Appadurai 1990) and reorganisation of labour and structure of capitalism (Boltanski and Chiapello 2007), contemporary work is generally organised based on the disconnection between diverse kinds of practitioners according to specialisations. This disconnection is embedded not only in formal organisations of labour, but is also reflected in the frameworks of producing technologies that underpin such work, and that produce disconnect between practices through prioritising different forms of knowledge.

In research aiming to transcend simple dichotomies of theory and practice, the work of Donald Schön (2003) has been very influential. He points out that professional work, as implemented in fordist frameworks of production, departs from a model of *technical rationality* that separates theoretical knowledge from engineering and practical skills. The concept of technical rationality is also discussed by philosopher of technology Andrew Feenberg (2009) as the way in which technologies represent modern forms of hegemony through which power gets established. Technologies reflect dominant ideas and ideologies of societies which get inscribed in the modes of technological development and its outlook. Technical rationality, according to Feenberg, encompasses not only technical devices and systems but also work, knowledge production and everyday communication. In work practice this has implied increased specialisation and standardisation of production techniques and frameworks to reflect largely, an economic rationality of ever-increasing efficiency. Technical development, in this context, becomes an application of theoretical knowledge, as well as the basis for practical work, creating an asymmetric relationship between different kinds of knowledge. This fragmented understanding of knowledge does not only value theoretical knowledge over practical skills, but also assumes that knowledge can be taken out of context (Schön 2003, pp. 21-30).

Separating knowledge from experience is in Donna Haraway's (1991, pp. 88-91) terms, a 'god-trick', an attempt to distance the knowing subject from the studied and practiced Western knowledge-based production. For Haraway, knowledge is always situated and partial. Building on Haraway's discussion about knowledge, Lucy Suchman (1994) puts forth a feminist approach to technology development. It is based on erasing artificially created borders between different knowledge frameworks through establishing "working relations" between users and producers of technology. Technical expertise, Suchman suggests, is a necessary, but not sufficient, form of knowledge for the production of new technologies, and she states that 'the development of useful systems must be a boundary-crossing activity, taking place through the deliberate creation of situations that allow for the meeting of different partial knowledges' (Suchman 2002). For our enquiry this means that a separation between producers of software (engineers and programmers), and its users (media practitioners) in the context of production frameworks of technologies of work, specifically software, represents also a separation between different forms of knowledge, based on differences in the nature of their working experiences. Schön complicates this understanding by stating that knowledge itself cannot be separated from the act of working, or practice: 'Our knowing is ordinarily tacit, implicit in our patterns of action and in our feel for the stuff

that we are dealing. It seems right to say that our knowing is *in*<sup>3</sup> our action' (Schön 2003, p. 49).

Far from being an application of a decontextualized knowledge framework, *knowing-in-action* is a bodily and sensory form of tacit knowledge, characterized by actions and judgement being carried out spontaneously, without thinking (Schön 1992, p. 5; 2003, p. 54). This kind of automated skill demands practice, or repetition. Through encountering a situation over and over again, the practitioner develops a repertoire of expectations, images and techniques that makes him or her better prepared for unexpected or uncertain situations in work practice.

Schön's work has been fairly criticised, among other things for being too individualistic and not taking social context into account (e.g. Thompson & Pascal 2012). Despite its shortcomings however, we believe that his understanding of knowledge and reflective practice are helpful to inform a discussion on how knowledge and practice can be configured through different modes of *producing the technologies* on which practice depends. In the next section we will discuss how the production frameworks of software, as the new technology for creative work, connect to problems of epistemology of practice.

### **The media practitioner as an efficient worker**

The production frameworks of today's market monopolists of computer graphics software for animation production are conditioned by several historical developments. In part, they reflect the earlier history of software development in which, between the 1980s and 1990s, computer graphics software was conceptualised as a tool to increase the productivity of workers engaged in industrial design (Downey 1998; Henderson 1999). This position has carried an assumption of the intrinsic superior qualities that the computer and new visualisation technologies possess, echoing an instrumentalist worldview, according to which the creative worker should be empowered and made more efficient through tools that the industry provides for her.

The initial goals of companies like Adobe and Autodesk were rooted in ideas about automation of labour as a path towards greater work efficiency, and subjected to the logic of algorithms developed by engineers detached from actual practices of use. Later on, these ideas were adapted by tuning software to satisfy the needs and speed of production of specific branches of the media industries, while at the same time trying to make it into a universal tool that fits a broad range of practices. As Aylish Wood (2015, pp. 28–29) recalls, the

---

3 italics in original

popular 3D software Maya has been specifically developed for the productivity and expressive demands of the markets and industries dealing with visual effects, animation and games by favouring particular styles of movement, photorealism and simulations of textures. From such a perspective, the popular consumer software that became a standard component of 3D animation today was not envisioning to democratise media production, but rather to produce new efficient workers necessary for the entertainment media industries undergoing digitisation.

In effect, the production model of such software was tuned as to reflect this vision, and implicitly reproduces divisions of labour and knowledge through separating producers of software and the greater majority of media practitioners using it. Embedded in proprietary frameworks of distribution, market-dominant software for computer graphics follows the general anticipatory mode of software development. Communication between media practitioners and developers of software is hierarchically organised and mediated through the figure of the 'technical support' (Downey 1998). Programmers anticipate the needs of practitioners in a relative disconnect from their actual, practical needs and problems. At the same time, the economic logic that guides technological development makes software producers to strive for creating products that fit the broadest possible groups of users. Henderson (1999) describes the impossibility of creating one software for computer graphics production that fits any design need, and points out how the work practices, from individual design processes to the organization of workplaces, adapts to the software instead of the other way around. Henderson further points out that the designers of such technological systems have limited models of media practitioners and their needs, and can never fully imagine the problems that can emerge in situations of practical use of software. Instead, the media practitioner must adapt to the logic of the tool, as described in the following quote by a freelancing video editor who works with such software.

What an editing tool looks like, is ...a kind of an agreement since way back, it's a norm in its basic structure that I, as a user, simply have adapted to. The method gets kind of embodied. [...] I'm not very interested in dialogue [with developers], since I find that the software I'm currently working with satisfies my needs. Which I may be happy about... In this, competition plays an important role. If the user is not satisfied, he or she simply moves to another software. This means the manufacturer must listen, improve and renew. And I think this is working.

*(Interview with 'Anna', freelancing video editor, April 2017)*

What 'Anna', describes as an embodied method is the kind of knowledge Schön (2003, p. 54) calls *knowing-in-action*, tacit skills in using a specific tool. It is important here to note that the practitioner in this case does not oppose the technical rationality of this framework, but is herself part of an efficiency paradigm of media production that prioritises speed over innovation. That the framework prioritize *knowing-in-action* of course does not mean that other forms of knowledge do not exist within it, or that practitioners within this framework do not reflect over their practice. However, the production framework of the software which they use may or may not allow them to act on those reflections through modifying the tool. This means that if a software tool is insufficient for working with the problem set by the practitioner, he or she has to solve this in other ways such as through often time consuming workarounds, or as this case, by changing software.

This production framework also reflects a linear and hierarchical model of knowledge according to which, first engineers develop software, and then it is taken by media practitioners in turn to solve problems of work. In real life professional practice however, 'problems do not present themselves to practitioners as given' (Schön 2003, p. 40). Instead, problems are invented by practitioners to solve or at least frame a problematic situation in a dynamic, problem-solving and problem-finding rhythm as suggested by Sennett (2008). This situation can in many times be uncertain, confusing and contradictory, making the task of a pre-defined and overdetermined technical commission impossible. Instead, the process of defining decisions must depart from the production tools of the situation – in this case, software.

The distinction that is maintained between media practitioners and developers may be explained as a logical continuation of a longer history of industrialisation and reorganisation of labour in capitalist societies, as suggested earlier. However, this is not the only way in which software for computer graphics gets developed. Indeed, such a mode of development is typical for products that are aimed at being sold at the end-user market of consumer software. Other industries, such as the Hollywood entertainment production companies, favour other modes of software production that configures knowledge and practice in a different way. To illustrate the difference, we take Pixar's framework of software production as a reference point.

### **The media practitioner as a craftsperson in a fordist framework of labour**

Pixar's technological production framework, as described by Tai (2012) and Sito (2013) is based on an assumption of animators, riggers, and modellers employed to work on film productions as craftspeople rather than as mere efficient workers who have to be supplied with tools. As such, they have to be provided with possibilities to change the technological instrumentarium on which their work depends in ways as to fit their creative ideas and the problems that they invent. This logic arguably stems from the need of the media industries to harness creativity and at the same time provide unique products on the market in order to be competitive. In this case, technology and the possibility of media practitioners to influence its development gain importance for upholding such a competitive advantage. Software is used and developed in a mode of cooperation between computer scientists, in-house programmers and artists working on Pixar productions. Representing still different fields of work, they are brought to work together in order to develop the aesthetic vision, movements, and the technology and tools for each production (Sito 2013; Tai 2012).

For pragmatic, marketing reasons this cooperation has historically tended to be veiled in a discourse of magic, emphasising that the products of Pixar, like those in other fields of the media industries, are made by the talent of exceptional artists (Catmull, 2014, p. xi). As Alfred Gell (2010) writes, magic in the context of artistic work encapsulates the idea of making something with no effort or knowledge. The image of magic in popular discourse comes to represent an ideal technology, and shifts attention from the tremendous role of technology in creative practice. Pixar's mode of software production in relation to the production of form – animation films – enables the company to create visual media beyond the limits of existing technological possibilities.

This mode of work frames the media practitioners as commissioners of features in a tool, but at the same time preserves the border between the professional knowledge of the developer and the media practitioner, as evident in this quote from a Ted talk where light designer Danielle Feinberg describes the process of visualising life under water from 'Finding Nemo', where she emphasises the need to bring together science and art development.

We use science and the world we know as a backbone, to ground ourselves in something relatable and recognizable. 'Finding Nemo' is an excellent example of this. A major portion of the movie takes place underwater. But how do you make it

look underwater? [...] we're using the science — the physics of water, light and movement — to tether [the] artistic freedom.

*(Danielle Feinberg, The Magic Ingredient that brings Pixar movies to life, Ted talk, Nov 2015)*

For something to be brought together it needs to first be separated. We therefore argue that this process builds on a fragmented concept of knowledge that understands science as the foundation for art making practices. The quote also consolidates the linear view of creative work as problem-solving. Although the media practitioner in this framework, whether an animation artist, modeller, light designer or script writer, can articulate his or her needs, they must be framed as a problem – such as how to simulate underwater environment in film - in order to be understood by the developer.

Suchman (2002, p. 95) describes this organisation of labour in terms of *detached intimacy* and claims that it characterises much of what we know as technical development. Her suggestion, to reshape this mode of organisation, is to allow for partial knowledges to come together. The first step in doing so is to recognize our own position in the development process, as well as the visible and invisible work conducted, in other words a reflection upon one's own practice (Suchman 2002, p. 100). This aligns with Schön's (2001, p. 61) argument that reflection is a way to overcome the selective inattentiveness of tacit knowledge, but he also differs between different temporalities of reflection. Practitioners can think *about* their work on multiple levels: the decisions they took, what could have been done differently, how they felt in a certain situation, what tacit norms or feelings that underlie a judgement or their role in an institutional context to mention a few. Schön (2001, p. 61) calls this distanced or retrospective thinking *reflection-on-action*. In order to ask for new features in a software tool, the practitioners must be able to critically examine and articulate their professional practice and imagine new ways to do things. The view of the media practitioner as commissioner of features in a tool therefore requires some kind of such reflection-on-action. However, the media practitioners envisioned here are not, as Dormer (1997, p. 140) puts it, 'masters or mistresses of the available technology' but craftspeople situated in a fordist framework of labour. To find the craftsperson described by Dormer, we must turn to a third production framework that both adopts and extends elements of the first two frameworks.

### **The media practitioner as a master of tool development**

More than a decade ago, a Dutch animation studio called Blender Institute started experimenting with crafting an alternative framework for computer graphics software

production, that mimics the one in Pixar by allowing media practitioners to access and shape the software that they need according to their individual practice. In contrast to Pixar which keeps its in-house developed software as a proprietary technology that is not made available to media practitioners broadly, Blender relies on a radical approach of sharing online software, films, computer graphics assets and training materials as a way to establish connections between producers of media (media practitioners) and producers of software (developers). Starting with de-commodifying a proprietary software for 3D animation in the early 2000s called Blender, the Blender Institute engaged in developing short animation films with the aim to extend the capabilities of this software in cooperation with media practitioners and developers (Velkova 2017). In a course of a year, a selected group of the latter worked together to develop a script and a film with high technical challenges that would require to develop completely new features in Blender (Velkova 2016). At the end of the production, the film, its data assets, as well as large parts of the technology were released online under a Creative Commons license, and as free software. Important to note is that these projects have not been done online, nor for free by volunteers: they were made in Amsterdam, and produced by a mixture of public funding and crowdfunding that prepaid the production (Velkova & Jakobsson 2017). Using sharing online as a form of gifting that enforced relations of exchange and thereby cooperation between media practitioners and developers of software, the Blender community of media practitioners has grown to a few hundred thousand over a decade (Velkova 2016).

What is important for our argument here is the epistemological orientation that software gets when practitioners can take it and bend it to their current needs, and when this happens as a result of enforcing recursive relationships of reciprocity between media practitioners and developers through sharing code and media made with such software online. First, this approach reverses a long established superiority of engineers and programmers over the design of work processes for others engrained in software. Second, it requires a mode of reflection that transcends the articulation and afterthought needed to commission tools, and instead means thinking about a certain aspect of practice while performing it. This way of work represents *reflection-in-action* and differs from the reflection-on-action in that it can lead to new decisions that might change the action that provoked the reflection (Schön 2001, p. 62-63)..

In a slow process like software development, this action can be stretched out in time over weeks or even months, but it still differs from the repertoire of past experiences used for knowing-in-action. Reflecting in a situation has little to do with tacit knowledge or skills.

Instead, it is a surfacing of such automated understandings and actions that leaves them open for reconsideration and critique. Reflection-in-action might occur in an unexpected situation when the conceptual frames in place are not sufficient. This means that the practitioner, in order to reflect-in-action, must not only recognize the irregular but also embrace and act upon it, an approach often attributed to artistic work (ibid). This relation between reflective practice and functionally undetermined tools is explicated by a Blender animation director in the following way:

Free software matches very good with the artistic idea because no artist wants to be locked into what they can do—a lot of the process of making art is about making the tools. (*Bassam, animation director, archived blog post, 2014*)

Characteristic for reflective practitioners is also how they *relate* to other kinds of knowledge. Rather than keeping the black box of her special expertise closed, the reflective practitioner allows uncertainty to be visible to others and a source for learning and development (Schön 2001. p. 300). Viewing the visual media practitioner as a master, or co-developer, of tools demands precisely this responsiveness towards the other, and her knowledge, feelings and thoughts. In the process of developing new tools for media production, this applies to both engineers and media practitioners. Through reflective practice perhaps different forms of knowledge can be articulated, put into conversation and used to create new technology that better reflects the creative process and allows for non-linear work and uncertainty.

### **Conclusions: an epistemology of creative digital media practice**

In this chapter, we have suggested a theoretical model of how different production frameworks of software tools for computer graphics are underpinned by different epistemological assumptions and competing visions of media practitioners.

To begin with, approaching the media practitioner as an efficient worker, builds on a fragmented concept of knowledge and demands skilled workers who can perform their tasks as they appear and without too much thinking. These media practitioners, we suggest, are envisioned by the frameworks of industrial producers of software for the mass-market as needing to possess predominantly *knowing-in-action*, the spontaneous and automated action and judgment that comes with training and repetition. This vision reflects a doctrine of productivity that emerged since the 1980s in the United States, in which software was envisioned as a tool to increase the efficiency of workers engaged in industrial design. It

builds on a fragmented concept of knowledge where creative skills are separated from other forms of knowing. Of course, some media practitioners can engage to a limited degree with modifying such software, and thereby gain access to other forms of knowledge, but our point is that this is not the dominant orientation of such production frameworks.

Other frameworks, as in the case of industrially made software by the entertainment industry, such as Pixar, focus more on invention and specificity of tools for its own production needs. We have argued that such vision understands the media practitioner as a craftsperson, that has to be able to modify her tools to fit specific production needs, rather than as a worker who is trained to use the provided and predefining tools. These modifications are not performed by the practitioner herself, but commissioned by her to a developer or a scientist. The technical rationality of this framework thus demands that the practitioner *reflects-on-action*, becomes aware of their own problem setting process, imagines alternative ways of doing the work and communicates these needs to the developer.

Although not as linear as the model of the efficient worker, the craftsperson view presupposes some kinds of causality. First, the media practitioner does some kind of work, either problem setting or creation, then reflects on it and frames this as a need. Only after that can she call for a tool or feature that answers to this need. For a more intuitive work process that includes the dynamic of the problem-solving and problem-finding rhythm (sennett, 2008), the media practitioner must be able to modify their software tools as part of the working process. This in turn calls for a holistic and isochronal form of knowledge, a conscious way of looking at one's own practice and reconsider choices and judgements while performing them. In other words, to envision the media practitioner as a master of tool development, as in the case of Blender, *reflection-in-action* is needed.

Indeed, the third, user driven production framework that we discuss through the example of the free software for 3D animation Blender, borrows elements from the above two but also contests them. Blender aims to be both a tool for efficient work, and a functionally underdetermined tool that allows to be modified and extended in practice, as need occurs. Its functionality and features get shaped in different work practices and production contexts by the practitioners themselves. This, we suggest, builds on a holistic understanding of knowledge that transcends the articulation and afterthought needed to commission tools and instead means thinking about a certain aspect of practice while performing it.

These three epistemological orientations of software should not be seen in terms of which is superior. From the point of view of the industries that employ them, we see all of

them as necessary means to achieve certain ends. In the first case it is to create efficient workers who can supply quickly content to subcontractors, such as broadcasters and advertising agencies. The more flexible framework of Pixar also strives for efficiency, but in the sense of making the most of the media practitioners' creative potential, and strengthening it through technological development, as well as for gaining market dominance by technological innovation. In the case of Blender, we can distinguish an attempt to make media practitioners outside high-end media production companies part of the media technology development process. However, these frameworks also contain a number of contradictions, such as between innovation and efficiency; linearity and messiness; and situated knowledge through specific software tools versus universal knowledge through general-purpose tools.

These contradictions ultimately reflect the difference in technical rationalities embedded in the software for computer graphics media production – the different view on knowledge and role of media practitioner in relation to work. At the same time, as the example of Blender suggests, such rationalities are not inevitable, but they can occasionally be modified by media practitioners when they engage in creating software frameworks that allow them to change the software tools on which their work depends. Although not superior to other modes of production and development, only frameworks that enable such changes ultimately has the potential to act on the contradictions of technologies of digital work.

## References

- Adamson, G. (Ed.). (2010). *The craft reader*. Oxford; New York: Berg Publishers.
- Andrejevic, M. (2008). Watching Television Without Pity: The Productivity of Online Fans. *Television & New Media*, 9(1), 24–46.
- Appadurai, A. (1990). Disjuncture and Difference in the Global Cultural Economy. In M. Featherstone (Ed.), *Global culture: nationalism, globalization, and modernity: a Theory, culture & society special issue*. London; Newbury Park: Sage Publications.
- Boltanski, L., & Chiapello, È. (2007). *The new spirit of capitalism*. London: Verso.
- Bolter, J. D., & Grusin, R. (1999). *Remediation: understanding new media*. Cambridge, Mass.: MIT Press.
- Catmull, E. E. (2014). *Creativity, Inc.: overcoming the unseen forces that stand in the way of true inspiration*. (A. Wallace, Ed.). New York: Random House.

- Dormer, P. (1997). *The culture of craft: status and future*. Manchester, New York: Manchester University Press.
- Downey, G. L. (1998). *The machine in me: an anthropologist sits among computer engineers*. New York: Routledge.
- Feenberg, A. (2009). "Critical Theory of Technology" in Olsen, Jan-Kyrre Berg, Pedersen, Stig Andur & Hendricks, Vincent F. (red.) (2009). *A companion to the philosophy of technology*. Malden, MA: Wiley-Blackwell
- Frabetti, F. (2015). *Software theory: a cultural and philosophical study*. London: Rowman & Littlefield International.
- Gell, A. (2010). The Enchantment of Technology and the Technology of Enchantment. In *The craft reader* (English ed, p. check). Oxford; New York: Berg Publishers.
- Haraway, D. (1991) "Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective", in *Simians, Cyborgs and Women. The Reinvention of Nature*, London: Routledge
- Henderson, K. (1999). *On line and on paper: visual representations, visual culture, and computer graphics in design engineering*. Cambridge: MIT Press.
- Pfiffner, P. S. (2003). *Inside the publishing revolution: the Adobe story*. Berkeley, Calif: Peachpit Press.
- Schön, Donald A. (1992) "Designing as Reflective Conversation with the Materials of a Design Situation" *Knowledge-Based Systems*, 5, 1992, 3-14.
- Schön, Donald A. (2003 [1995]). *The reflective practitioner: how professionals think in action*. Repr. Aldershot: Ashgate
- Sennett, R. (2008). *The craftsman*. New Haven: Yale University Press.
- Sito, T. (2013). *Moving innovation: a history of computer animation*. Cambridge, Massachusetts: The MIT Press.
- Suchman, L. (1994) Working Relations of Technology Production and Use. *Computer Supported Cooperative Work (CSCW)* 2:21-39.
- Suchman, L. (2002) "Located accountabilities in technology production," *Scandinavian Journal of Information Systems*: Vol. 14 : Iss. 2 , Article 7. Available at: <http://aisel.aisnet.org/sjis/vol14/iss2/7>
- Tai, P. (2012). *The Principle of Animation: History and Theory of a Social Technology*. Irvine: Doctoral Dissertation. University of California Irvine.
- Thompson, N., & Pascal, J. (2012). Developing critically reflective practice. *Reflective Practice*, 13(2), 311–325.

- Velkova, J. (2016). Open cultural production and the online gift economy: The case of Blender. *First Monday*, 21(10). Available at <http://firstmonday.org/ojs/index.php/fm/article/view/6944>
- Velkova, J., & Jakobsson, P. (2017). At the intersection of commons and market: Negotiations of value in open-sourced cultural production. *International Journal of Cultural Studies*, 20(1), 14–30.
- van Dijck, J. (2009). Users like you? Theorizing agency in user-generated content. *Media, Culture & Society*, 31(1), 41–58.
- Wood, A. (2015). *Software, animation and the moving image: what's in the box?* Houndmills, Basingstoke, Hampshire, UK; New York, NY: Palgrave Macmillan.