

Chapter 2

A Genealogy of Software Applications

Abstract This chapter outlines a broad genealogy of two areas within software culture: Digital Non-Linear Editing (DNLE) and Computer-Aided Design (CAD) software. Emerging from distinct institutional environments, their respective historical developments and the implications these have generated within their professional domains provide a broader context for the software at the centre of this educational research project (see Chaps. 3 and 4). Each of these histories demonstrate how decisive the institutional and industrial contexts of their creation were in inscribing the affordances, interfaces and conceptual frameworks coded into these software.

2.1 Introduction

Every software application comes with its own history; it emerges from a particular context. The institutional context and corporate history for its emergence imparts a legacy, embedded within the code itself which evolves as the software becomes adopted by the user base and increasingly entrenched within professional and other practices. These contexts and histories are fundamental to the overall conceptual framework underlying the software. They also inform and shape its set of affordances (the set of actions possible within the software) and how these are organised in the design and configuration of such things as an application or platform's interface. As outlined in Chap. 1, these aspects of the software serve to enable and constrain the possible sets of practices to which such software can be applied by users. In a broader sense, software and its users evolve together, and a history of their development provides a necessary wider frame from the research 'snapshot' generated through our specific research project. The software explored in this chapter offer useful illustrations of how many applications developed in close partnership with professional practitioners, evolving into the software which later came to be used more widely, including becoming embedded into educational and training environments (the focus of our particular study).

The two disciplinary contexts we explore in our research are media studies and engineering (see Chaps. 3 and 4). This chapter begins with a (necessarily brief) narrative of the field of Digital Non-Linear Editing (DNLE) software, which is part

of a core set of media editors which have arguably transformed media production practices over the last 20 years. The second half of the chapter provides an overview of the development of Computer-Aided Design (CAD) software, which is central to a range of material practices, including those within the engineering discipline. The aim is not to provide an exhaustive history or genealogy of the specific software discussed in later chapters, but to suggest something of the trajectory of their development and outline the implications of their acceptance by practitioners within their respective creative fields.

2.2 The Development of Digital Non-linear Editing Systems (DNLE)

The long development of Digital Non-Linear Editing (DNLE) systems arguably represents a transformation as significant to moving images as word processing was for writing. However, this development is comparatively poorly researched as with much of *cultural software*, to use Manovich's term (see Chap. 1). In a sense, we are talking here about a *cut, copy and paste* approach to the construction of moving images. Earlier (analog) editing practices entailed a *destructive assembly* process, where a film strip was literally cut into pieces and reassembled until there was agreement on the final edit. While film could theoretically be endlessly recombined, in practice the materiality of film strips meant this gradually became more and more difficult. Digital systems, in contrast, allow for the random access retrieval of digital material in order to build a sequence that exists virtually, and with editing outcomes that are usually recorded (and outputted) as an edit decision list (EDL) (Murch, 2001). The key, and quite profound, advancement offered by DNLE systems, was that they allowed the creation of as many versions of film sequences as a user wanted as all that was being manipulated were digital files. The penultimate output from a DNLE system (the final EDL) was used as the blueprint to *cut* or *print* the film itself. The integration of this software into professional film making practices had enormous implications for production workflows, and arguably ultimately changed the ways in which audio-visual production came to be *imagined* by practitioners (as is discussed below). Despite the significance of this translation of the established practices of editing into software code, it is important to emphasise that film and television producers were by no means early adopters of incorporating computer-based tools into their production practices. The transition to a fully digitised production workflow faced many obstacles; some of these were technical as ambitions were delayed by the limitations of available technologies while others were more cultural and institutional. Innovations in this area happened first with small groups of early adopter professional practitioners, before slowly becoming more widely available as the cost of specialised editing systems decreased and editors became more accustomed to using these tools. The Moviola editor, a ubiquitous flatbed system for viewing/cutting film strips which had served as the standard analogue editing system since the

introduction of sound in 1927, survived late into the 20th century, evidence of the decades it took for DNLE to achieve dominance within the industry.

The technical barriers to a completely digital process were not insignificant and generated caution from some of those who might have been early adopters. This reticence derived partly from the division between different physical formats, which made it difficult for developers of new versions of editing systems to recreate all editorial workflows. For example, there are key material differences between film and video editing (used respectively for film and television production) which meant that initially it was not possible for developers of editing systems to cater to both. The technical challenges of digitising editing practices also derived from the initial limitations of computer technology itself, such as storage and processing power. The distinction between offline and online editing,¹ for example, has gradually disappeared as available computers have become powerful enough to handle editing at full resolution. Some technical issues were too deeply embedded within existing production technologies to immediately solve, such as the distinction between the standards for European film and video (25 frames per second) versus US film and video (24 fps + 30 fps). These added costly conversions to and from film or video media files as part of production workflows—these challenges remain as legacies within file formats and compression codecs² today. Other challenges only emerged after earlier problems had been solved, and are part of a much broader and familiar trajectory of emerging technologies such as the transition to high definition (HD), 4K video and so on, which are beyond the scope of this overview.

It is important to emphasise that editing practices—that is, the ways in which editors conceptualised and organised their workflows—developed very much in tandem with each other in incremental ways that entailed a slow transformation of the nature of editing itself (Thompson, 1994). Rubin's narrative of this transformation highlights the key period of 1989–1993 as the real emergence and dominance of fully digitised NLE (Rubin, 2000), but there are complex, overlapping developments in this history with most periods characterised by long time lags as established technologies survived even when there was rapid development of digital technologies. Even today there are some film directors who insist on creating motion pictures on film despite the process of distribution and exhibition having been largely digitised. There are a number of potentially significant (but largely under-theorised and under-researched) milestones in the slow reconceptualization of audio-visual editing as a coded practice (see Dancyger, 2011; Ohanian 1998; Rubin, 2000; Thompson, 1994). The year 1995 is frequently cited as a key year in which digital editing became more widespread within the more elite editing practices (and budgets) of Hollywood production. This is marked especially with the dominance of the Avid Film Composer system which epitomised developments that had been made in solving key technical challenges,

¹Offline editing involved transferring film to video, to make it easier for editing systems to deal with the digitised footage.

²A compression codec (short for coder-decoder) encodes a media file for storage and distribution, and decodes it for playback or editing.

including the problem of video/film footage transfer. Together with Apple's Film Cut Pro and Adobe Premiere, these constituted the big three of professional level editing systems. Most crucially for this history, the unbundling of software from hardware systems fostered a more competitive environment which resulted in the rapid widespread adoption of any innovations pioneered by any one vendor. With the increasing competition between a small number of key players, innovation became secondary to standardisation, and this is most evident in the emergence of the now familiar template for editing interfaces.

2.2.1 *The Interface*

The graphic user interface (GUI) for digital non-linear editing—the interface which editors engage with on their computer screens—is the what confronts users when opening any audio-visual editing application, and the elements of its basic design and key features are replicated across both professional-level applications and those designed for more novice users. In its layout and terminology, this interface retains something of the legacy of the physical operations of film flatbeds (such as the Moviola system). The various elements of this interface gradually came together in successive iterations of DNLE applications produced by various vendors over a numbers of years. Ohanian provides a useful summary of the key elements of the contemporary DNLE system (Ohanian, 1998, pp. 52–56):

- The Clip: the granular component of all editing, derived from *the shot* in film editing, which tends to represent a single continuous set of footage, and which is represented by an icon, text, and frame in the interface.
- The Transition: derived more videotape editing, where there was more of a need to *fill in* the space left when a shot was trimmed, but now appearing as a variety of options for editors to apply across cuts between clips.
- The Sequence: a sequential series of (trimmed) clips, stills and other material (such as various kinds of audio), which can in turn be combined into larger sequences and so on. These sequences, a key building block for editing, might be generated by different editing teams on large productions, and combined later.
- The Timeline: the centre of the interface, where multilayered sequences (combining layers for different video and audio material) can be combined to create sequences which play out over time.

These essential elements of this interface have been replicated across editing software and have become deeply integrated with other applications to form the basis of a profoundly transformational approach to the construction of moving-image media content. This is how editing is now performed and imagined by editors (and other personnel in the production line).

2.2.2 *The Implications of DNLE*

The provision of fully-digital workflows, fostered by software from major players such as Avid, Apple, and Adobe, prompted an immediate reorganisation of production workflows. For many large productions, instead of a single editing team providing a single bottleneck for all raw footage, there might be a number of machines operating simultaneously, working on different footage, to be combined later.

The implications of DNLE have been more profound than this; however, the extent to which *editing* has been redefined through digital workflows is heavily debated within industry circles. Walter Murch is one of the few practitioners who has reflected on the transition from analogue to digital editing practices. His seminal text *In The Blink of An Eye* (2001) reveals his enthusiasm as an early adopter of digitised editing across the film industry. His perspective is an informed one, and a good illustration of the close and dialectical relationship between professional practitioners and the evolution of the systems they adopted.

Murch also appears in an interesting case study on the transition to digital editing practices: Koppelman's (2005) account of Murch editing *Cold Mountain* (Minghella, 2003) on Apple's Final Cut Pro (FCP) system. This was a production using film on location, then the footage was migrated to digital video and came out the other end as film again for distribution to theatres (with film distribution yet to convert to digital projection as the norm). Significantly, FCP was considered, even by Apple itself, as a *prosumer* application, midway between professional and consumer software. FCP was widely used to edit documentaries, but not considered robust enough for feature film production. Murch's adoption of the system on a big budget feature film represented both an early example of the convergence of consumer-level digital video tools and Hollywood film industry, and a fascinating account of a software developer being pushed to re-imagine an application developed exclusively for digital video, and which in fact required third-party tools to operate (Koppelman, 2005).

These kinds of accounts reveal how practice evolved in tandem with software development. Murch's reflections on the nature of DNLE in his own writing, and as relayed by Koppelman in his case study, provides a useful summary and reflection of broader opinion within editing practitioners as a community. Overall, although this is expressed in different ways, there is a recognition that the shift from the materiality of film, from the *destructive* approach of editing film, to a fully digitised system has meant the adoption of a new conceptual approach. This has played out in different ways for different kinds of editing practice, but there are some broad observations articulated by most editors.

DNLE allows for the possibility of increased speed in editing, which itself means a less considered and methodical exploration of the potential ways to combine and recombine clips into sequences. This generally means a more efficient editing process (and hence less costly, a key driver in the adoption of these systems). These efficiencies are somewhat mitigated within digital workflows by the vastly increased volume of footage which is able to be captured using digitised cameras with large storage capacities. As DNLE allows a system to develop and create multiple versions

of an edit, these workflows have also opened up the editing room to the more direct intervention of directors and other personnel. DLNE also allows for a more integrated approach to how image and sound might work together, rather than the older production process of adding sound (Murch, 2001). As Dancyger notes, individually these are all small changes in workflow and the ways in which editing is imagined, but collectively they represent a significant change in practice (Dancyger, 2011).

Murch himself highlights the changes to the nature of analog film editing as a physical, embodied practice. His own practice involved standing at a Moviola flatbed deck, using his whole body to work the viewing and the cutting of film strips in a way that became intuitive (Murch, 2001). He notes that editing involves the logistical wrangling of footage, analysis of the structure of sequences together into a rough edit, and the actual performance of the editing itself, and all three areas are transformed within DNLE (Koppelman, 2005). He also developed specific elements to his workflow that are only possible with the materiality of film; for example, he would physically rewind the editing tape back to the beginning through the viewfinder, meaning he would watch sequences backwards to get a completely new perspective on its structure—something that isn't possible with the *scrubbing* feature of digital video players (which allows users to jump ahead multiple frames, to skip through sequences at high speed).

The overall speed and ease of this cut, copy and paste approach to editing attracted complaints from some practitioners and commentators that this has degraded the quality of the considered reflection that needs to be at the heart of distinctive and innovative editing solutions for each project (Murch, 2001). These accounts point to the emergence of more formulaic and standardised approaches to editing across different kinds of media content as a key implication of digitised workflows. Ellis (2012) argues that accelerating the process of editing has had implications within broader patterns of accelerated cutting in media content, something he characterises as a loss of craft and individual editing styles (Ellis, 2012) and a greater density in cutting styles, such as quicker cutting between multiple perspectives and angles within the same scene.

Perhaps the most profound transformation associated with DNLE, however, is not provided by the affordances of the systems themselves, but facilitated by the ease with which material can be imported and exported to other forms of software. Now it is often the case that different pieces of software will handle specific kinds of image and sound construction and editing, which are then combined as layers within a more generic media editing application. This broader context of exchange of digitised material means that *coded* filmmaking processes have taken on very different qualities to previous eras, and this is manifest in the types of changes exhibited within media content more generally.

Manovich's analysis of the Adobe After Effects (AE) application is a useful addition to debates in this area (AE is part of the package which is taught within universities as industry-standard, including within the media discipline researched within our project, see Chap. 3). Manovich writes as a practitioner, noting the changes to his own practice, and highlights the period 1993–98 where a change in the aesthetics of particular kinds of media content became noticeable. He uses the term Velvet

Revolution (as in the slow drift of revolution in Czechoslovakia in 1989) to describe this gradual transformation, led by AE and a small number of similar programmes, which have fostered a new *hybrid visual language* of motion graphics (Manovich, 2006).

What is the logic of this new hybrid visual language? *This logic is one of remixability: not only of the content of different media or simply their aesthetics, but their fundamental techniques, working methods, and assumptions.* United within the common software environment, cinematography, animation, computer animation, special effects, graphic design, and typography have come to form a new 'metamedium'. A work produced in this new metamedium can use all the techniques which were previously unique to these different media, or any subset of these techniques (Manovich, 2006, p. 10, emphasis in original).

Instead of creating films where an animation sequence was followed by a live action sequence and so on, these various kinds of media content (generated by quite different workflows and raw materials) could all operate as layers within a single overarching timeline, and ultimately begin to interact at a more fundamental level. It is only over time that he belatedly recognised, even as a practitioner, the implications for his own imaginative possibilities for creating media content, as motion graphics began to become the standard for short-form audio-visual content, such as television commercials, and the opening credit sequences to television programmes.

DNLE itself is now typically packaged within a larger ecosystem of production tools, all specialised media editors which are increasingly imagined to operate together to provide a wide spectrum of possibilities for media producers to play within. The emergence of these software-based tools suggests a redefinition of audio-visual practice itself, their collective impact and the emergence of distinctive new conceptual frameworks. Underlying these changes in organising screen content are greater uncertainties concerning the organisation of creative labour itself:

Specifically, digitization has facilitated a collapse and confusion of production workflow and upended traditional labor [sic] hierarchies. Workflow refers to the route that screen content travels through a production organization and its technologies as it moves from the beginning (origination, imaging, recording) to the end (post-production, mastering, duplication, exhibition) of the production/distribution process. [...] In fact, the once linear sequence through which filmed material went before being printed and broadcast has fallen apart. Because of these recent shifts to digital, visualization and effects functions once reserved for post-production now dominate production, and skills once limited to production now percolate through post-production (Caldwell, 2011, p. 293).

A host of software-enabled specialisations, such as colour grading, motion capture, the generation of CGI, and motion graphic techniques, allow for a wider palette of techniques available to media producers. Admittedly, this is a narrative which does not encompass all of audio-visual production; at the opposite end of the filmmaking spectrum are mobile, amateur and networked practices which have reconfigured DNLE in quite different directions (Hight, 2014a, b). Overall, however, the students participating in our research encounter sophisticated, professional-level editing systems with specific conceptualisations embedded in their hierarchy of affordances and in their interfaces. Next, we turn our attention to another ubiquitous software, in this

case used to facilitate the design of engineering, architectural and other physical artefacts in three-dimensional (3D) format. Computer-Aided Design (CAD) is another part of software culture which has had wide-ranging implications for a reimagining of creative practices across a number of related industries.

2.3 The Development of Computer-Aided Design (CAD)

As with the discussion on Digital Non-Linear Editing (DNLE), what follows is necessarily truncated and cursory, as we do not have the space here to delve into the wealth of literature which attempts to analyse and summarise the implications of Computer-Aided Design (CAD) practices. In our own small project we engaged specifically with an engineering discipline, but it is important to note CAD is an aspect of software culture with wide applications within design, architecture and related practices, where it has become a given set of tools with wide-ranging implications for the nature of professional practice.

CAD involves the use of software in the creation, modification, analysis or optimisation of material design (if we define this broadly, to include a range of practices from the design of nuts and bolts, through to more complex forms of mechanical engineering encompassing everything from automotive to bridge design, and ultimately to forms of built environments or architecture). Some of the transformation of material practices associated with this kind of software has been extensively debated, particularly within architectural literature. This befits a field which sees itself as aspects of design practice which transcend the merely functional. In these circles digitised workflows consequently attracted intense debate over the social, cultural and political implications of its outcomes.

CAD arose from a very different institutional environment to DNLE (with the Massachusetts Institute of Technology playing an outside role), but there are some parallels and interesting points of comparison in terms of the significance of a number of new conceptual frameworks which emerging software eventually come to embody and foster. We are concentrating on architecture and engineering in this account, but there are obvious areas now where 3D modelling and media editing software operate together within particular kinds of creative practices (the most recent and celebrated include augmented and virtual reality, but there are deep roots here into forms of computer graphics and game design). As with all software it is increasingly obvious that applications and platforms formed within one sector of human endeavour quickly start to become part of the broader incestuous and prolific combinatorial evolution of software culture (as broadly outlined in Chap. 1).

The development of CAD forms one part of a broader history of engineering and architectural design practice itself, and is associated with a number of transformational milestones in these practices. Some of the earliest *technical* drawings for machines or devices date back to the 14th or 15th century, among the most famous are those produced by Leonardo da Vinci. However, if we were to consider these drawings in a modern context they would be described as *sketches* as they lack dimensions

or scales and often have exhaustive text descriptions to help the viewer understand the intent (Weisberg, 2008, p. 2–1). These early drawings served two purposes: a reference for skilled craftsmen to construct the device depicted and also as a portfolio to present one’s work to a wealthy patron (Lefèvre, 2004). Crucially, at this point in history there was a clear separation in practice between those who offered designs of material objects and those who actually built such things based on those designs. In marked contrast to contemporary practice, this was not a collaborative relationship nor a space where early *architects* were acknowledged as the drivers of projects.

Leon Battista Alberti is invariably credited with inventing modern architecture, in the sense that he exploited the new technologies to insist that the designer was the author of a building and no longer beholden to the craftspeople who actually created a building. Before Alberti, architects had to contend with builders who interpreted their designs according to their own practices and the demands of their local contexts. So the creation of a building was an inherently collective and decentralised process, relying on oral, material and technical traditions outside of the control of the architect (Llach, 2015). The Albertian paradigm is a key reference point for understanding the emergence of CAD. One of the broader ironies of this history is that this software at first seemed to fulfil the promise of the Albertian approach, but has in more recent years gradually undermined it.

Using a new notational system, and exploiting the possibilities for the new technology of print to provide an exact replica of a design, Alberti could insist that the architect was indeed the *author* of a building, not just a starting point for a design which was re-shaped on location by other craftspeople. So in Alberti’s terms, “the *design of the building is the original, and the building is its copy*” (Carpo, 2011, p. 26, emphasis in original). Following Alberti, the notational system of architecture helped to establish a distinct identity for architecture, which in turn eventually helped to set the conceptual stage for the arrival of computers as tools to serve these masters (Llach, 2015).

The specific origins of CAD are typically seen as located within the Massachusetts Institute of Technology (MIT), which produced the first CAD software, Sketchpad, as part of Ivan Sutherland’s doctoral research in 1959,³ building on a variety of earlier work by researchers inside and outside the institution (Cohn, 2010; Llach, 2015). As with early DNLE development, there were very few practitioners who had the resources to commit to investigating the use of the early prototypical and expensive systems. Consequently, the early development of CAD, in an engineering context, was primarily driven by large aerospace and automotive companies. These were companies which were able to afford the expensive computer equipment required and were already engaged in such complex design processes that they were attracted to the possibilities of the reduction of drawing errors, increased reusability of drawings and greater efficiencies promised by CAD. It is important to recognise that the adoption of these systems was driven by a search for greater efficiencies in productivity rather than a design tool. Instead they offered systems to find drawings more quickly,

³Not coincidentally, Sutherland later emerged as a key figure within the history of computer graphics.

simplify modifications of drawings and allow the automation of some parts of drawing practices (CADAZZ, 2004).

As Llach notes, in contrast to the popular conception that Computer-Aided Manufacturing (CAM) is an offspring of Computer-Aided Design (CAD), the opposite is true. Like filmmaking, engineering and architecture were comparatively late adopters of embedding computers into everyday creative practice. CAD developed from experiments to automate manufacturing, and it was only later that the transformational potential for design itself came to be realised (Llach, 2015).

The ethos and vocabulary of manufacturing gave origin to the first CAD systems (Llach, 2015, p. 37), but this was also, unusually for software culture, a highly theorised process. The development of CAD at MIT was complex, and significantly involved a great deal of debate about the nature and desirability of the human-machine hybrid practice which might result. MIT not only developed CAD as a tool, but generated a series of accompanying theoretical reflections that helped to shape assumptions on how it might operate within industry. These debates centred on the use of creatively using computers, the need to divide labour between humans and machines and the implications of re-imagining material design as a kind of data processing (Llach, 2015).

These debates are quite distinctive from those associated with the development of DNLE, as they drew upon a broader caution about the nature and role of computers within material design practice. The term *Computer-Aided Design* itself reflects the demand that computers support human creativity, rather than any sense that there should be a collaboration between human and machine (Llach, 2015). CAD consequently tended largely toward generating efficiencies through the augmentation of pre-existing practices in the 1970s and 1980s (Llach, 2015). For example, engineering within aerospace leader Boeing had an *all CATIA, no paper*⁴ design strategy. This led to a substantial reduction in time to market by safely eliminating the need for physical mock-ups (often required to verify paper designs). The typical impetus for the adoption of CAD was still the quest for workflow efficiencies. In late 2000 automotive manufacturer Ford showed that 3D CAD, with internet enabled product data management (PDM), could cut the concept to shelf time to approximately one third of that required by the more common, non-internet enabled techniques. The primary advantages of the network enabled method were that they allowed viewing and collaboration by geographically dispersed teams on a single digital master, almost eliminating the misfit and mismatch problems often associated with globally dispersed manufacturers and parts suppliers.

While MIT was crucial to the broader development of CAD, and succeeded in actively shaping the popular imagination with design fields and wider (Llach, 2015), ultimately how CAD developed diverged from this original role. The longevity of its introduction into everyday practice perhaps aided in this adaptation, as CAD gradually drifted further from how it was conceived by its creators, as it became diffused through architectural and engineering practice. As the software itself became more sophisticated with enhancing computer technologies, there was a gradual shift of

⁴CATIA is a CAD platform.

focus from simply automating the practice of drafting into something more transformative: the emergence of a platform facilitating a comprehensive building (and design) simulation (Llach, 2015). These are all developments which at first glance appear to provide a narrative of inevitable transformation, a confirmation of the claims of technological determinism. Initial CAD programs effectively just translated the blueprinting process onto a digital platform, and it was only as the software increased capability to allow for the techniques of 3D modelling that its broader creative capacities became prevalent.

Modern 3D CAD programs include a variety of sophisticated analysis tools that allow various simulations to be run on the 3D item/structure. This has given rise to the term *virtual product development*, where products are developed and prototyped in an entirely digital medium (CADAZZ, 2004). Today, CAD is used extensively in most activities in the design cycle, everything from recording product data, to allowing for remote collaboration between design teams (Bilalis, 2000). The open co-creation possibilities of CAD software emerged gradually, but also in a highly theorised way, a reflection of the significance of the university environment as a breeding ground for its conception and early prototypes.

Interestingly, Llach's critical perspective draws explicitly from the Software Studies paradigm, arguing that software needs to be examined "as part of the infrastructures that condition the design and production of built environments" (Llach, 2015, p. 23). For commentators such as Llach, what was at stake is the nature of the creative endeavour itself. Before the widespread use of CAD in the education of engineers, there was much greater emphasis on drawing and sketching (Buchal, 2002). Hare (2005) says that sketching is inherently creative, the practicing and sketching frequently leads to more creative thinking; in fact, analog tools, such as pen and paper, are still viewed as more haptic and intuitive. From this perspective, CAD can guide an engineer through technical issues, such as dimensions and scaling, but it does not have the same ability to create quick visualisations like sketching does. Moving to a CAD workflow, then, might mean losing key elements of design practice.

Llach cautions against making generalisations in this area however, as the use of the CAD tool has varied greatly and the use of these systems are deeply informed by practitioners' own position within debates over the role which computer-based practice should play. He cites the example of Frank Gehry, who continued to construct actual models, which were then scanned and inserted into computer form. The process is complex here, as the potential of the software also clearly informed the imagination of architects, allowing them conceptual space to potentially re-imagine the nature of their own practice. The role of the software is still, in everyday disciplines globally, negotiated and framed by broader agendas and localised practices (Llach, 2015).

2.3.1 Implications of CAD

The overall paradigmatic changes associated with CAD workflows have been neither universal nor linear. Initially this software represented a confusion of the Albertian

perspective, and the emergence of a vision of architectural design as data processing (Llach, 2015, p. 66), in the process “revealing software as a territory where the meaning of design itself is negotiated” (Llach, 2015, p. 87). Rather than a slave for the Albertian paradigm, the computer has sparked a profound refashioning of the nature of material design practices, such as engineering, with debates now centred on the nature of the human-machine assemblage that has emerged, and which way development should now progress.

Just as DLNE is now part of a broader production ecology that challenges understandings of what media are (Manovich as cited in Chap. 1), Llach argues that software is a site for competing theorisations about design, and consequently, “the technology project of CAD appears as a disciplining project, not an emancipatory tool, but rather a governing one” (Llach, 2015, p. 102). The broader implications are complex, and there is (again) notably more detailed and extensive theorising about these aspects within discourses surrounding CAD than for DNLE.

Robertson and Radcliffe (2009) argue that “there is growing evidence that the ubiquitous CAD tools that design engineers use in their everyday work are influencing their ability to solve engineering problems creatively, in both positive and negative ways” (p. 136). Positive factors include the ability to visualise and *play* with designs, less time spent on detail (potentially allowing more time on being creative), and enhanced communication facilitating *group creativity*. Negative impacts tend to be vaguer, though Robertson and Radcliffe have identified four general categories:

- Enhanced visualisation and communication: there are obvious positive aspects to this category. Negative impacts included having clusters of people crowding around a monitor hampering brainstorming; and the tendency when a detailed CAD model was displayed for it to convey an illusion of completeness and discourage further creativity.
- Circumscribed thinking: this could either be where the functionality of CAD limited solutions (either to what was possible to do in CAD, or perhaps worse, what was easiest to do in CAD); or at the other end of the scale very proficient CAD users using the functionality of the tool to develop unnecessarily complex designs because CAD allowed it rather than because these were the best design solutions.
- Bounded ideation: the notion that using CAD for large portions of a day was not necessarily conducive to creativity (the mundane nature of drafting along with technical problems and software bugs being a distraction from the process of designing).
- Premature fixation: as CAD models became more complex (usually as the design process proceeds) there was greater disincentive to make changes (presumably due to the amount of work that would be required to make these).

As always, debates centred on whether such new human-machine assemblages truly enhance innovative and effective design practice. Some commentators insist on a profound paradigmatic change prompted by CAD, with hints of the technological determinism underlying some writing on software culture more broadly. Carpo writes that the “Albertian paradigm is now being reversed by the digital turn” (Carpo, 2011, p. 27).

The idea that the new digital design tools could also serve to make something else – something that would not otherwise have been possible – may have occurred when architects began to realize that computer-aided design could eliminate many geometrical and notational imitations that were deeply ingrained in the history of architectural design. Almost overnight, a whole new universe of forms opened up to digital designers. Objects that, prior to the introduction of digital technologies, would have been exceedingly difficult to represent geometrically, and could have been produced only by hand, could now be easily designed and machinemade using computers (Carpo, 2011, p. 36).

There is parallelism here with the developments of fully realised CGI-animated film worlds fostered by media production, but prompted also by the influence of postmodern theorists such as Gilles Deleuze. He offered a new language of *folds* in architectural design (helping to prompt the development of algorithmic affordances in CAD platforms which could realise these in virtual form). The fold, “a unifying figure in which different segments and planes are joined and merge in continuous lines and volumes, is both the emblem and the object of Deleuze’s discourse” (Carpo, 2011, p. 86).

And, crucially, the rigour of the Albertian paradigm is much more compromised now within this environment. Instead of a firm commitment to the authorship of the architect, who produced a design and anticipated that it would be exactly replicated in the building itself, and the early CAD phase where the software was used to implement broader assumptions of standardisation and automation, today CAD allows for a more fluid and ever changing re-imagination of the nature of design itself. An architect’s original plan could once again (as in pre-Alberti times) be endlessly reinterpreted using individual explorations at different points in the design process: “In a digital production process, standardization is no longer a money-saver. Likewise, customization is no longer a money-waster” (Carpo, 2011, p. 41).

As CAD became more embedded within material design disciplines, such as engineering, it allowed three-dimensions to become part of the authoring process. As with DNLE the broader ecology of software development has

made it possible to envisage a continuous design and production process where one or more designers may intervene, seamlessly, on a variety of two-dimensional visualizations and three-dimensional representations (or printouts) of the same object, and where all interventions or revisions can be incorporated into the same master file of the project. This way of operating evokes somehow an ideal state of original, autographical, artisanal hand-making, except that in a digitized production chain the primary object of design is now an informational model (Carpo, 2011, p. 33).

A key point was that design representations now became “forms of *building*” structured information, engineered rather than drawn (Llach, 2015, p. 67, emphasis in original). Architects were able to model new constructions in the software itself. A key shift here is toward the term “*modeling*, often used by architects to describe the production of three-dimensional descriptions in software, [which] evokes manual work in a way that other words, such as simulation, do not” (Llach, 2015, p. 100). The CAD process has evolved increasingly into an ever-more data-intensive set of practices, with recent developments of building information modelling (BIM) deeply embedded within automated practices allowing and requiring greater databases of content to be folded into the design process.

The CAD systems currently available for students within disciplines such as engineering (one of the foci of our research in the following chapters), then, are highly sophisticated, densely designed software-based platforms enabling a wide variety of material practices. They have evolved from their origins as tools to serve, support and help implement human creativity, emerging as human-software engines which are challenging for any practitioner to master, and typically offer a daunting software environment for novice users to encounter (as we shall see in Chap. 4).

2.4 Summary

This chapter has scoped the genealogy and development of two distinctive forms of software—DNLE and CAD—commonly taken up within the professional fields of media studies and engineering today. Obviously, it is not enough to provide such broad histories, as they reveal little but generalisations and theorising extemporised from exemplars and case studies at hand. What is required from this point is more detailed explorations of how and the extent to which these might play out within specific institutional contexts, whether these broader patterns and generalisations hold true across different practices, deployed by distinct practitioners, within institutional variations, and any number of other factors. We turn to this task in the next two chapters.

References

- Bilalis, N. (2000). *Computer aided design CAD*. Report produced for *INNOREGIO: Dissemination of innovation and knowledge management techniques*. Retrieved from http://www.adi.pt/docs/innoregio_cad-en.pdf.
- Buchal, R. O. (2002). Sketching and computer-aided conceptual design. In *7th International Conference in Computer Supported Cooperative Work* (pp. 112–119). Design, Brazil: IEEE.
- CADAZZ. (2004). CAD software—history of CAD CAM. Retrieved from <http://www.cadazz.com/cad-software-history.htm>.
- Caldwell, J. T. (2011). Worker blowback: User-generated, worker-generated, and producer-generated content within collapsing production workflows. In J. Bennett & N. Strange (Eds.), *Television as digital media* (pp. 283–310). Durham, NC: Duke University Press.
- Carmo, M. (2011). *The alphabet and the algorithm*. Cambridge, MA: MIT Press.
- Cohn, D. (2010). *Evolution of computer-aided design, digital engineering* (December 2010). Retrieved from <http://www.digitaleng.news/de/evolution-of-computer-aided-design/>.
- Dancyger, K. (2011). *The technique of film & video editing: History, theory, and practice* (5th ed.). Burlington, VT: Elsevier.
- Ellis, J. (2012). *Documentary: Witness and self-revelation*. London, UK: Routledge.
- Hare, R. (2005). *The act of sketching in learning and teaching the design of environments: A total skill for complex expression*. Retrieved from <http://www.lboro.ac.uk/microsites/sota/tracey/journal/edu/hare.html>.

- Hight, C. (2014a). Shoot, edit, share: Cultural software and user-generated documentary practice. In K. Nash, C. Hight, & C. Summerhayes (Eds.), *New documentary ecologies: Emerging platforms, practices and discourses* (pp. 219–236). Houndmills, UK: Palgrave Macmillan.
- Hight, C. (2014b). Automation within digital videography: From the Ken Burns effect to ‘meaning-making’ engines’. *Studies in Documentary Film*, 8(3), 235–250.
- Koppelman, C. (2005). *Behind the seen: How Walter Murch edited Cold Mountain using Apple’s Final Cut Pro and what this means for cinema*. Berkeley, CA: New Riders.
- Lefèvre, W. (2004). *Picturing machines 1400–1700*. Cambridge, MA: MIT Press.
- Llach, D. C. (2015). *Builders of the vision: Software and the imagination of design*. New York, NY: Routledge.
- Manovich, L. (2006). After effects or the velvet revolution. *Millennium Film Journal*, 45(46), 5–19.
- Minghella, A. (2003). *Cold Mountain*. Miramax. DVD.
- Murch, W. (2001). *In the blink of an eye: A perspective on film editing* (2nd ed.). Beverly Hills, CA: Silman-James Press.
- Ohanian, T. A. (1998). *Digital nonlinear editing: Editing film and video on the desktop*. Boston, MA: Focal Press.
- Robertson, B. F., & Radcliffe, D. F. (2009). Impact of CAD tools on creative problem solving in engineering design. *Computer-Aided Design*, 41(3), 136–146.
- Rubin, M. (2000). *Nonlinear: A field guide to digital video and film editing* (4th ed.). Gainesville, FL: Triad Publishing Company.
- Thompson, C. (1994). *Non-linear editing: A survey*. New Technology and Training Series. London, UK: Skillset, The Industry Training Organisation for Broadcast, Film & Video and British Film Institute.
- Weisberg, D. E. (2008). A brief overview of the history of CAD. In *The engineering design revolution: The people, companies and computer systems that changed forever the practice of engineering*, (pp. 2–1). Available at <http://www.cadhistory.net/02%20Brief%20Overview.pdf>.



<http://www.springer.com/978-981-10-7058-7>

Software Literacy

Education and Beyond

Khoo, E.; Hight, C.; Torrens, R.; Cowie, B.

2017, XIII, 101 p. 6 illus., Softcover

ISBN: 978-981-10-7058-7