

**Manuscript version: Author's Accepted Manuscript**

The version presented in WRAP is the author's accepted manuscript and may differ from the published version or Version of Record.

**Persistent WRAP URL:**

<http://wrap.warwick.ac.uk/145493>

**How to cite:**

Please refer to published version for the most recent bibliographic citation information. If a published version is known of, the repository item page linked to above, will contain details on accessing it.

**Copyright and reuse:**

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

**Publisher's statement:**

Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: [wrap@warwick.ac.uk](mailto:wrap@warwick.ac.uk).

**THE UNKNOWABILITY OF AUTONOMOUS TOOLS AND THE LIMINAL EXPERIENCE OF THEIR  
USE**

**Zhewei Zhang** (University of Warwick)  
Aron Lindberg (Stevens Institute of Technology)  
Kalle Lyytinen (Case Western Reserve University)  
Youngjin Yoo (Case Western Reserve University)

Forthcoming in *Information Systems Research* 2021

## THE UNKNOWABILITY OF AUTONOMOUS TOOLS AND THE LIMINAL EXPERIENCE OF THEIR USE

In the extant theoretical discourse on socio-technical systems, the relationships between inputs and outputs of technologies are assumed to be knowable to human agents, occasionally *ex ante*, but always *ex post*. Recently a new breed of autonomous tools has emerged, which can independently learn and execute novel actions. The input-output-relationships of these tools, however, are unknowable to human agents, both *ex ante* and *ex post*. This calls for analysis of how humans experience the enactment of socio-material agency while interacting with autonomous tools. To this end, we conduct an exploratory, theory-building, comparative case study at one of the world's largest semiconductor manufacturers. We investigate how chip designers interact with two families of design technologies: one following a traditional designer-centric approach where the designer knows what outputs particular inputs to the tools will generate, and another relying on autonomous tools which continually surprise the user. Our inquiry reveals significant differences in designers' experiences of using different tools. When using autonomous tools, designers' experience of enacting socio-material agency becomes *liminal*; a state of continuous emergence, where interactions with the tools are marked by ambiguity, and the design is moved forward along multiple design trajectories in accordance with a multifarious temporality. These insights require us to expand upon several dominant views on the enactment of socio-material agency and necessitate novel thinking on the role and impact of autonomous tools in future work systems as well as on how design and innovation proceeds under such conditions.

**Keywords:** Autonomous Tools, Socio-Material Agency, Liminality, Design, Digital Innovation

## THE UNKNOWABILITY OF AUTONOMOUS TOOLS AND THE LIMINAL EXPERIENCE OF THEIR USE

*Scout, a chip designer working at a major semiconductor design and manufacturing company, is tuning a tool for building the layout of a chip subsection. She attempts to generate three separate chip designs based on three sets of parameters which she has specified, hoping that the tool will produce several new promising designs by Monday morning. As she starts packing up for the weekend, she thinks to herself: “I hope some of these come out alright, but you never know.” On Monday morning, she logs into her workstation and downloads the finished designs from her design repository. One set of parameters failed to converge, thus not producing any results, while another set produced a subpar layout with regards to the speed and electrical interference requirements for her subsection. The third one, however, appeared to be promising. As Scout surveyed the intricate design produced by her tools, she thought to herself: “These designs are truly remarkable, but I honestly can’t make any sense of them—they seem somewhat random to me!” She then began the arduous labor of specifying a new round of runs by modifying the parameters that led to the third design.*

This vignette illustrates how a new breed of digital technologies now support many design and engineering tasks. These technologies operate autonomously while carrying out complete design tasks; they assume high-level input from designers who, in turn, focus on identifying salient design goals and constraints expressed in a set of design parameters (Seidel et al. 2018a; Seidel et al. 2018b). These technologies rely on computation-heavy, self-learning algorithms, which draw upon as well as produce exceptionally large volumes of data. Their use has been made possible by cost-effective access to powerful, distributed computing resources connected through cloud-based infrastructures (Tilson et al. 2010). These technologies, as the vignette shows, are now capable of performing actions that are unanticipated by their users, and even by the tools’ designers. Simply put, the technologies exhibit a new kind of material agency<sup>1</sup>, prompting us to refer to such technologies as *autonomous tools*.

---

<sup>1</sup> We use the terms “material agency,” “socio-material agency,” and “socio-technical systems” as these are established terms in the literature. In this study, however, the term “material” primarily refers to the “technical” aspects of tools, i.e., specific forms of material agency possessed by technologies that rely on computation. We chose to retain these accepted terms to not distract the reader from our core argument.

The emergence of autonomous tools challenges many pivotal assumptions that underpin the studies of the mutual constitution of agency by human agents and technologies acting in a concerted, systemic fashion agency which we, in short, call *socio-material agency*. Past studies have used a multitude of perspectives to account for and characterize the nature of such agency. These include distributed cognition (Hutchins 1995), critical realism (Leonardi 2011), actor-network theory (Latour 2005), and agential realism (Barad 2003). The inquiries that draw upon these positions have galvanized increasingly sophisticated debates concerning the degree to which the agency of technologies extends, connects with, or limits human agency (Leonardi 2011), and explores how socio-material agency emerges from the ways in which human agents understand and interact with the technology and its features (Beane and Orlikowski 2015).

The received views of socio-material agency characterize technologies as having relationships between their inputs (e.g., instructions from a designer) and their outputs (e.g., artifacts generated by a design tool) that may occasionally be unknowable *ex ante*, but will become knowable *ex post*, whether during or after usage. These technologies are assumed to function in ways that the agents using them principally understand and can make sense of based on their use experience. Per these perspectives, human agents' knowledgeability of the input-output relationships of technologies create the necessary epistemic foundation for an agent's ability to enact the tools and assimilate the use experience. Such knowledgeability enables human agents to use technologies in ways that will extend their agency. However, the emergence of autonomous tools shatters most of these assumptions because the input-output relationships of autonomous tools remains fundamentally unknowable to human agents, both *ex ante* and *ex post* (Pasquale 2015). This is not only due to the ambivalent ontology of the digital tools (Kallinikos et al. 2013) or the opacity (Burrell 2016; Turkle 1997) and "black-boxed performance" (Faraj et al. 2018) of complex algorithms and their learning capabilities (MacKenzie 2018; MacKenzie 2019), but also due to the dynamism of the computing environments within which these technologies operate. This fundamental shift in the epistemic foundations of human tool deployment invites us to rethink how the enactment of socio-material agency is experienced. Hence, we ask the following research question:

*How do autonomous tools reshape the human experience of enacting socio-material agency?*

To answer this question, we examine the use of two families of design automation (DA) technologies at ChipCo—a pseudonym for a leading global semiconductor manufacturer. The company has successfully designed and manufactured integrated computer chips for over 40 years and has been involved in DA efforts for the last 30 years. As a result, its design processes are supported by a wide range of sophisticated design technologies that represent, implement, track, validate, and record chip designers' decisions. Overall, the company has digitalized its design tasks in ways that put designers firmly in control of the design process and its outcomes. During our field study, ChipCo started to change its design approach and automation goals by introducing autonomous DA technologies to generate increasing portions of the physical layout of an integrated circuit (IC) chip<sup>2</sup>. During the initial stages of our field study, ChipCo used these technologies only for a few select sections of the chip, while continuing to use traditional DA technologies to design the rest of the chip. The parallel use of two distinct families of DA technologies offered a unique opportunity to conduct a comparative study on the effects of the two families of DA technologies. This allowed us to identify significant differences in the enactments of socio-material agency and related experiences under the two different technological conditions.

## THEORY REVIEW

The concept of agency—an agent's capacity to act within, respond to, and shape its environment (Emirbayer and Mische 1998)—is fundamental to most streams of social theory. Not surprisingly, most debates around the origins and nature of agency focus on *human* agency and deal with issues such as free will, intentionality, rationality, motivation, or the relationships between agency and structure. An excellent review of how human agency has been treated in several streams of sociology, anthropology, and economics is Emirbayer and Mische (1998). In their treatise, human agency manifests an interplay of three temporal dimensions: (a) iterative—a human agent's ability to recall, select, and apply her past experience; (b) projective—a human agent's ability to imagine the possible trajectories of her future actions; and (c)

---

<sup>2</sup> The terms “integrated circuit chip,” “semiconductor chip,” “IC chip,” and simply “chip,” are used interchangeably, as is customary in industry, see, e.g., [https://en.wikipedia.org/wiki/Integrated\\_circuit](https://en.wikipedia.org/wiki/Integrated_circuit)

practical-evaluative—a human agent’s ability to make practical and normative judgments with regards to alternative trajectories of action in the present. In this view, the experience of agency consists of how these temporal dimensions are enacted, emphasized, and interrelated in practice. Human agents can enact temporal structures that tie together the three temporal dimensions of agency in multiple ways. For example, agents can emphasize either the past or future dimension in their action, either through reminiscing about the past or by being strongly projective with regard to the future (Shen et al. 2015). Similarly, agents can experience a tight coupling of the temporal dimensions, and establish stringent causal connections between their past, their current decisions, and their future projections. Such connections, of course, may also be loosened, prompting humans to experience disconnects between their past, the present, and the future, thus producing a disorienting experience of how and why events unfold the way they do.

Going back as far as Marx (1945), several analyses of human agency have examined how, and to what extent, human agency is enabled and conditioned by technologies, and to what extent technologies exhibit an agency of their own, i.e., *material agency*. Though material agency rarely has been examined in the context of three temporal dimensions of human agency, and some scholars have claimed that it lacks inherent reflective capacities to connect the past, the present, and the future (Giddens 1984), it nevertheless exhibits a capacity for taking action in relation to its environment and thereby extending or complementing human agency. The ensuing relationships between human and material agencies have been the subject of a long string of analyses concerning the nature of the socio-material agency of systems which share both technical and human elements that interact while carrying out human tasks (Trist 1981). In such settings, socio-material agency captures the joint capacity of humans and technology elements to interact and make a difference (Latour 2005).

The Information Systems (IS) field, due to its technical origins and organizational focus, has utilized notions of human agency informed by multiple strands of social theory (Bourdieu 1998; Giddens 1984) and has applied these notions while treating information systems as socio-technical systems (Bijker et al. 1987; Hirschheim 1985; Kling and Scacchi 1982; Latour 2005; Pinch and Bijker 1984; Sarker et al. 2019; Trist and Bamforth 1951). Hence, issues concerning socio-material agency have been pivotal to both

the theory and empirics of the field. The inaugural studies of socio-technical systems (for a review, see (Trist 1981) opened this line of inquiry by arguing that while humans create technologies, technologies also shape human agency. Generally, the relationships between technologies and social systems (e.g., tasks, human qualities and traits, organization, social norms, and institutional arrangements) are regarded as mutually dependent and recursively organized (Leavitt et al. 1962).

These studies usually assume that socio-material agency is forged under two conditions: knowability of tools and knowledgeability of human agents, implying that humans skillfully interact with technologies in order to extend their capabilities (Giddens 1984). The deployment of technologies is based on the human knowledge of the expected effects of using technologies under specific conditions, as humans project their plans into the future. This knowledge enables them to exercise the full range of temporal capacities associated with human agency, which, in turn, shapes the subjective, temporal experience of the agents (Emirbayer and Mische 1998). The concrete empirical ways in which human agency becomes interwoven with material agency has been examined in multiple strands of socio-technical studies. We will review some of the more influential strands that are concerned with how humans experience the enactment socio-material agency, especially with regard to the knowability of the input-output relationships of the technologies that are being used.

Building upon these foundations, but augmenting them with a perspective drawn from cognitive science, Hutchins (1995) examined the uses of navigation tools and operations in aircraft cockpits (Hutchins et al. 1996) while formulating his well-known theory of “distributed cognition.” This perspective focuses on how technological artifacts enable and extend human cognition. A stream of studies showed that the agency of such technologies is distributed in time and space, interacting with human agency (in the form of the cognitive tasks of inferring, calculating, and remembering) to constitute a distributed cognitive system. Hence, distributed cognition is a manifestation of socio-material agency formed through an interleaved “computation” process, whereby human actors and technological artifacts interact to produce specific cognitive outcomes (such as navigation decisions) across space and time. Hutchins posits that in concrete work settings, attempts to explain human cognition as occurring solely “within the skull” are futile.



Rather, human agents need to rely on multi-faceted artifacts and their standing capacities to store and propagate representational (cognitive) states (Hutchins 1995, p. 118). His theory also posits that human agents will *knowingly* draw on such specific capacities as they participate in distributed cognition. Essentially, the input-output relationships of technologies need to be knowable both *ex ante* and *ex post* for the distributed cognitive system to work.

Such knowledge empowers human agents in such a way that they experience tighter linkages between the past, their present decision-making, and its future consequences. Agents will draw on the past to inform their current decisions and can reliably project visions, plans, and goals into the future, based on their knowledge of the state of the artifacts (Boland et al. 1994; Mangalaraj et al. 2014; Shaft and Vessey 2006). In line with this, recent studies have applied a distributed cognition to software development (Xiao et al. 2018) and shown how open source software developers draw on well-understood heuristics that knowingly utilize artifacts features to store and transform knowledge representations that move a software designs forward. While doing so, developers tie together the past, the present, and the future in a conscious and controlled manner.

A nascent stream of IS research has drawn on critical realism (Archer et al. 1998) to expand the analyses of socio-material agency by adding concepts that emphasize the contextual and temporally bounded nature of the relationships that underlie socio-material agency. These aspects have been refined in novel concepts such as “affordances” (Gibson 1977; Gibson 1979; Norman 1990), borrowed from ecological psychology, as well as ideas of “imbrication” between human and material agencies (Leonardi 2013; Leonardi 2011; Mutch 2013). Leonardi (2011), for example, posits that while digital technologies partially owe their features to their inherent, standing material characteristics, such features themselves participate contextually in constituting socio-material agency. Hence, socio-material agency only emerges when the features are contextualized as affordances, i.e., when the features have become cognitively mediated, appropriated, and deployed by participating humans. The affordances can only be activated *hic et nunc* during ongoing human-technology interactions. Human agency is always imbricated with the agency of technologies, a process through which humans engage in ongoing interpretative efforts to

understand what technologies can actually do for them.

According to this view, the input-output relationships of technologies will be knowable to participating human agents *ex post*, and they are in many cases also knowable *ex ante*. Such knowledge, however, is always imperfect and evolving. Contextual interactions will, therefore, grow the agent's range of available affordances by increasing the agent's knowledge of technologies (Jung and Lyytinen 2014). Similar to the distributed cognition view, the critical realist perspective places the human agent and her knowledgeability in the foreground when explaining how the enactment of socio-material agency is experienced by participating humans. While the human agent can draw on past events and may project visions into the future, decision-making about technology use in the present remains non-deterministic. Technology use involves engaging with a largely tractable but uncertain reality that exhibits a range of probabilities, the underlying generative mechanisms of which cannot be observed directly. From this viewpoint, the practical-evaluative dimension of human agency is fraught with uncertainty and the space of affordances is unbounded (Fayard and Weeks 2007; Nan and Lu 2014). For example, Leonardi (2011) describes how mechanical engineers working on automobile crash simulations draw on the past by utilizing known affordances of design technologies while projecting into the future by providing constraints that condition the performance of these technologies. While the process is not deterministic, users are assumed to know, even if only nebulously, the impact of affordances and how they constrain future action. Human agents will therefore, at least in principle, draw together the past, the present, and the future as they enact socio-material agency.

Actor-Network Theory (Latour 1987; Latour 2012; Latour et al. 1992; Latour 2005) privileges neither the human nor the technology in studying how socio-material agency is formed. Human and material agents<sup>3</sup> make up actor-networks through which the interests of diverse agents become translated and mediated. In this view, socio-material agency emerges in the performance of a network of relationships mediating heterogeneous interests. Generally, such performance is a function of the capacities of the

---

<sup>3</sup> Latour (1987) uses the term "actant" to refer to both human and material agents. We use the term agents to maintain consistency across our argument.

participating agents, their relationships, and the included scripts. ANT is scale-free, in the sense that any agent can be made up of networks of lower-level agents (Latour 2005). Because the internal structure of an agent can be explicated as an actor-network, ANT assumes that the input-output relationships of agents can be known, at least *ex post*, when an action is being performed while activating a particular actor-network.

ANT, and the studies conducted using it, helps us understand how human agents enter into situations where their agency is weak and the events in which they participate are experienced as being exogenously driven. Because of the structural emphasis of ANT analyses, past events, patterns, and tendencies exert significant influence on how events unfold in the present and leave less space for projectivity. Humans experience the enactment of agency as if being carried forth by waves of events within networks nested within networks (Hanseth and Monteiro 1997; Kavanagh and Araujo 1995). For example, Faraj et al. (2004) studied the emergence of web browser technology and examined how technology-related practices were shaped within actor-networks by the processes of inscribing, translating, and framing. The input-output relationships of web browsers were therefore not wholly known by human agents *ex ante* by reference to their material features, but rather emerged through complex interactions within networks. Hence, in ANT, knowledge of input-output relationships is primarily available *ex post*, when a particular actor-network has been performed, stabilized, and made knowledgeable. This implies that the projective dimension is weakened, thus rupturing the tight connections between the past, the present, and the future in the experience of enacting socio-material agency.

Finally, the agential realist view, originally formulated by Barad (2003), suggests that the ontological separation of human and technological agencies is artificial, thus positing “ontological inseparability” (Orlikowski 2007; Suchman 2006). During interactions, the agencies become “inextricably intertwined” and can only be separated by “agential cuts” that temporarily and analytically separate the involved entities and their features from their holistic unity. In a sense, mutually constituted agency is the *only* ontological state of agency. Material agency only becomes knowable *ex post* and is endowed with agency once it is enacted by humans. Hence the agency of the technology cannot be separated from human agency, which itself is always relationally mediated by technologies. Through human enactments,

technologies become knowable through their contingent features-in-use, i.e., actionable technology features emerge only when they are enacted by humans.

Agential realism allows little room to analyze, *ex ante*, how the past and the future are connected in human action. Agency is present in the moment of practice while it is being performed (Introna 2011). For example, Beane and Orlikowski (2015) show how telepresence robots are enacted within a healthcare where the role and agency of the technology emerge as it is being enacted. In agential realism, past “materialized practices” influence and shape future action. Projecting into the future, however, becomes difficult as practice only exists as it is materialized in specific activities or artifacts (Orlikowski and Scott 2014). Hence, the input-output relationships of the technology become knowable primarily *ex post*, while *ex ante* knowledge is fallible and weak, given that it is present only in memory traces of previously materialized practices. The projective dimension of socio-material agency is experienced as weakened.

While each research stream shows significant variations in how the human enactment of socio-material agency is portrayed and experienced, a common assumption across these research streams is that the input-output relationships of technology artifacts become knowable, in some cases, *ex ante* or, at least after deployment, *ex post*. Furthermore, these research streams assume that over time and through repeated use, human knowledgeability increases and therefore allows for a growing mastery of technology, which enables a tighter integration of past, present, and future technology use. This, in turn, suggests that socio-material agency is experienced by human agents in ways that coherently connect the past, the present, and the future (Table 1).

Table 1. Received accounts of socio-material agency and its epistemic foundations					
Perspective	Socio-material agency	Human experience of enacting socio-material agency	Ex ante knowledge	Ex post knowledge	Representative citations
Distributed Cognition	Human agents extend their cognitive capacity and reach through tool use	Tight integration of iterative, evaluative, and projective dimensions of agency	Yes	Yes	Boland et al. 1994; Hutchins 1995; Hutchins et al. 1996; Mangalajaraj et al. 2014; Shaft and Vessey 2006; Xiao et al. 2018
Critical Realism	Human and material agencies become imbricated in that they mutually constitute agency iteratively as new affordances emerge through learning	Non-deterministic integration of iterative, evaluative, and projective dimensions of agency within specific contexts	Partial	Yes	Fayard and Weeks 2007; Jung and Lyytinen 2014; Leonardi 2011; Leonardi 2013; Mutch 2013; Nan and Lu 2014
Actor-Network Theory	Humans and technologies jointly possess capacities to “make a difference” based on their structural arrangements	Agency emerges only in structural arrangements, weakening the projective dimension of human agency while strengthening the iterative dimension.	Partial	Yes	Faraj et al. 2004; Hanseth and Monteiro 1997; Kavanagh and Araujo 1995; Latour 1987; Latour 2005; Latour 2012; Latour et al. 1992
Agential Realism	Material agency does not exist outside human agency and only comes to life when it is mutually constituted during entanglement	Agency emerges <i>hic et nunc</i> through performative enactment and emphasizes the practical-evaluative dimension of agency	Partial	Yes	Barad 2003; Beane and Orlikowski 2015; Introna 2011; Orlikowski 2007; Scott and Orlikowski 2014; Suchman 2006

### **How Autonomous Tools Challenge Received Notions of Socio-Material Agency**

Autonomous tools, as exemplified by technologies such as IBM's Watson (High 2012) and Google's DeepMind (Silver et al. 2016), perform complex cognitive tasks such as performing medical diagnosis, playing complex games, or designing a whole car body, all without the direct and continuous input from human agents. These technologies draw upon advances in artificial intelligence, machine learning, neural networks, and genetic algorithms, which are applied to huge datasets while leveraging powerful computing resources, enabling the performance of cognitively complex, and often creative, tasks such as synthesis, pattern detection, natural language processing, or prediction. These tools do this to such an extent that perceptions of their independent agency arise, i.e., the tools exhibit a form of *computational agency* (Winograd and Flores 1986) such that they have capacity to perform independent actions through computation. TSuch agency can be observed while these technologies perform cognitively complex tasks that string together multiple interdependent actions, often iteratively, to produce nearly complete solutions for given tasks without active human intervention (Tong and Sriram 1992). Moreover, these technologies have a capability to learn from their own actions and improve their "projective" capabilities (Lyytinen et al. 2020).

The complex, equifinal, and dynamic ways of arriving at algorithmic solutions through the use of autonomous tools, as well as the sheer complexity of solutions, severely limits the knowability of the input-output relationships of such tools. Consequently, what such technologies ultimately accomplish based on particular inputs, becomes, in principle, unknowable to participating human agents, both *ex ante* as well as *ex post*. Computational agency becomes opaque (Burrell 2016; Turkle 1997) and inscrutable (Pasquale 2015); technologies appear not only to act on their own, but to also do so in ways that cannot be understood either in an iterative or practical-evaluative sense (Emirbayer and Mische 1998). The material configurations of computational resources (such as storage, processors, or software) that underlie material agency change dynamically in real time and therefore have emergent properties (Kroll 2018). Hence, when using autonomous tools, the projective ability of human agency is severely weakened.

The starting point for understanding the reasons for the unknowability, *ex ante* and *ex post*, of the input-output relationships of autonomous tools is to understand the nature of the procrastinated binding that applies to all performances of computing devices (Yoo et al. 2012). Procrastinated binding refers to the process of dynamically executing the immaterial algorithm expressed in symbolic form on a physical device in a specific space and time continuum. During the algorithm's runtime, the binding ties together symbolic expressions and physical resources to produce a particular computational outcome. The binding is procrastinated in the sense that it does not exist when the algorithm is created; rather, it takes place *every time* the algorithm hits the silicon during runtime. The outcome of the computation does not emerge until the binding takes place and runs its course. Further, autonomous tools draw on algorithms that are non-deterministic and self-learning, such as genetic algorithms (Seidel et al. 2018a), making it difficult, if not impossible, to predict the outcome of each runtime. In addition, autonomous tools implement procrastinated binding in settings that are characterized by distributed and heterogeneous data and a wide array of computing resources distributed and allocated dynamically through cloud-based infrastructures. This renders the associated computational processes and their outcomes highly sensitive to the actual material conditions of runtime, such as the amount of memory or computing resources allocated (Dodgson et al. 2007), or the variety of data pools (Holland et al. 1989). Because of this, autonomous tools have an inherent capacity to produce *de novo* outcomes in each run (Kallinikos et al. 2013).

These characteristics of autonomous tools suggest the unknowability of the relationships between the inputs that human agents supply them with, and the outputs that the tools generate in response, both *ex ante* as well as *ex post*. These limits radically alter the reciprocal epistemic relationships between human agents and their tools, and therefore influence the experience of enacting socio-material agency—an experience which can be characterized as being *liminal*. The concept of liminality originated in anthropology (Turner 1987) as a way of capturing “in-between” states, such as those that take place during coming of age rituals. The concept has been further extended by organizational scholars, who have, for example, used it to capture the status and experiences of temporary workers (Garsten 1999). Within information systems research, scholars have used liminality to describe transitions from old to new roles

and systems, and to the digitally mediated interactions across the boundaries across different cultural environments (Wagner et al. 2012). Henfridsson and Yoo (2014) emphasize the ambiguous aspect of liminality, as experienced by institutional entrepreneurs when multiple innovation trajectories emerge, even though only a few can be executed. Thus, liminality is produced by the co-presence of multiple, distinctively different forces and potentialities that shape human experience, the balance of which is a state of emergence marked by ambiguity and multifariousness. In our study, these forces and potentialities are concerned with what is known and not known about the input-output relationships of participating technologies.

### **A COMPARATIVE CASE STUDY OF PHYSICAL LAYOUT DESIGN AT CHIPCO**

To answer our research question, we empirically explore how the use of autonomous tools reshapes the experience of enacting socio-material agency in the setting where our original research puzzle emerged: the task of designing the physical layout<sup>4</sup> of IC chips—one of the most challenging and complex engineering tasks of today. Semiconductor design in general, and physical layout design in particular, offers a rich context for the study of our research question for several reasons. Semiconductor design is an example of a highly complex design task that, due to its scale and dynamics, have had a deep connection to DA since its inception. As a cognitive task, chip design is highly abstract and complex (Appendix A provides a brief overview of chip design as well as a glossary). Due to miniaturization occurring at an exponential rate, contemporary IC chips contain several billion transistors<sup>5</sup>, where the gate length is 15-20 nanometers and is expected (at the time of the writing of this article) to soon decrease to 10-12 nanometers. To accomplish the complex design goals associated with such tasks, as well as to respond to the exponentially growing scale and complexity of related challenges, chip designers not only have to use a broad suite of DA tools but must also learn new tools and skills every few years. Chip designers frequently interact with tooling

---

<sup>4</sup> While the term “physical” is used here to describe a stage of the IC design, the design task itself is performed in a completely virtual setting. The term “physical”, as opposed “logical,” is used to denote the fact that the design output is a “physical layout” in real, Cartesian, two-dimensional space. The physical layout is the first time in the IC design process where representations of the physical chip, rather than computer code, are produced and analyzed.

<sup>5</sup> Currently the number of transistors on a typical core chip is over 6 billion. For example, Intel’s Core i9 chip has roughly 6.5 billion gates while AMD’s Epyc has a record 19.2 billion transistors.



professionals<sup>6</sup>, and some also build their own tools.

Since the early 80's, the dominant DA approach (i.e., a set of tools and an attendant way of using them to conduct design) to physical layout design has been Structured Digital Design (SDD) where a schematic design of a portion of chip is created aided by computers (Thomas et al. 1983). The designer next focuses on transforming the initial schematic into a physical layout using another set of computer-aided wiring tools. Since the early 2000s, a new generation of design approaches has emerged that use other types of DA tools, often referred to as Physical Synthesis (PS) tools, based on genetic algorithms (Brown and Linden 2009). We will refer to such tools as “autonomous tools.” These tools offer a powerful new way of improving physical design productivity because they autonomously generate full layout solutions for whole sections of a chip.

As the size and the complexity of the chip design continuously grows<sup>7</sup>, ChipCo came to the conclusion that the design approach leveraging traditional tools could not keep up with the pace of complexity in chip design, and therefore was not sustainable. In their search for a solution, they started to experiment with autonomous tools in non-critical areas of the chip. Autonomous tools generate satisficing, “good enough” solutions more quickly, but incur costs in terms of increased uncertainty as the designers have less control over the actual placement process. Over time, the performance and quality of PS tools have improved significantly; newer versions of placement algorithms have better and more efficient heuristics, and hardware performance has improved greatly.

At the time of our study, ChipCo used both these approaches in its physical design process, each supported by a separate suite of tools. In our study setting, the two approaches were applied under identical time-to-market constraints and identical types of requirements with regards to thresholds such as clock speed, heat generation, power consumption, electrical interference, etc. Further, both approaches were at times used simultaneously. While both approaches relied on digital tools, the autonomous tools operated in

---

<sup>6</sup> By tools, we mean software environments for chip design, akin to the integrated development environments (IDEs) commonly used in software development.

<sup>7</sup> As expressed by Moore's law, the complexity of chips double every 18-24 months.

quite different ways from the traditional tools. Hence, our comparative case study offers a fruitful research setting for understanding how autonomous tools reshape the experience of enacting socio-material agency.

### **Data Collection and Analysis**

We conducted a four-year, comparative, theory-generating case study at one of ChipCo's primary design centers. Since our goal was to understand how the enactment of socio-material agency was experienced differently across the two different DA approaches, we followed Yin's (2003) embedded case study approach<sup>8</sup>. During the study period, designers at ChipCo worked primarily on three major IC design projects that covered separate chip generations; we refer to these projects using the pseudonyms Maplewood, Calverton, and Downton. This setting allowed us to collect data from multiple design process episodes following the same DA approach as our embedded case, and then to compare the two design approaches across all projects. This design helped us to better account for potential cumulative learning effects and to avoid "accidental" interactions that might have been included in a smaller sampling window. During the study period, the ratio of sections designed by approaches that leveraged autonomous as opposed to traditional tools increased from roughly 1:1 to 4:1. Most of the sections are now designed using autonomous tools, with traditional tools being applied only to a few highly critical sections that exhibit complex and unusual layout requirements.

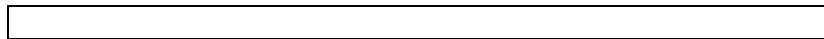
At ChipCo, the design of each generation of chips follows a strict 24-month cycle, divided into four distinct phases. During each phase, the designers need to generate a working physical layout that meets specific requirements set by the management team for that phase. As the project moves from phase to phase across the design cycle, the requirements grow stricter. We treated an individual designer's design process during each such phase as a unit of analysis. We were able to follow the last two phases of the Maplewood design cycle, Calverton's full design cycle, and the first two phases of the Downton design cycle. In total, we collected data from 31 embedded cases, 9 using traditional tools and 22 using autonomous tools (5 of

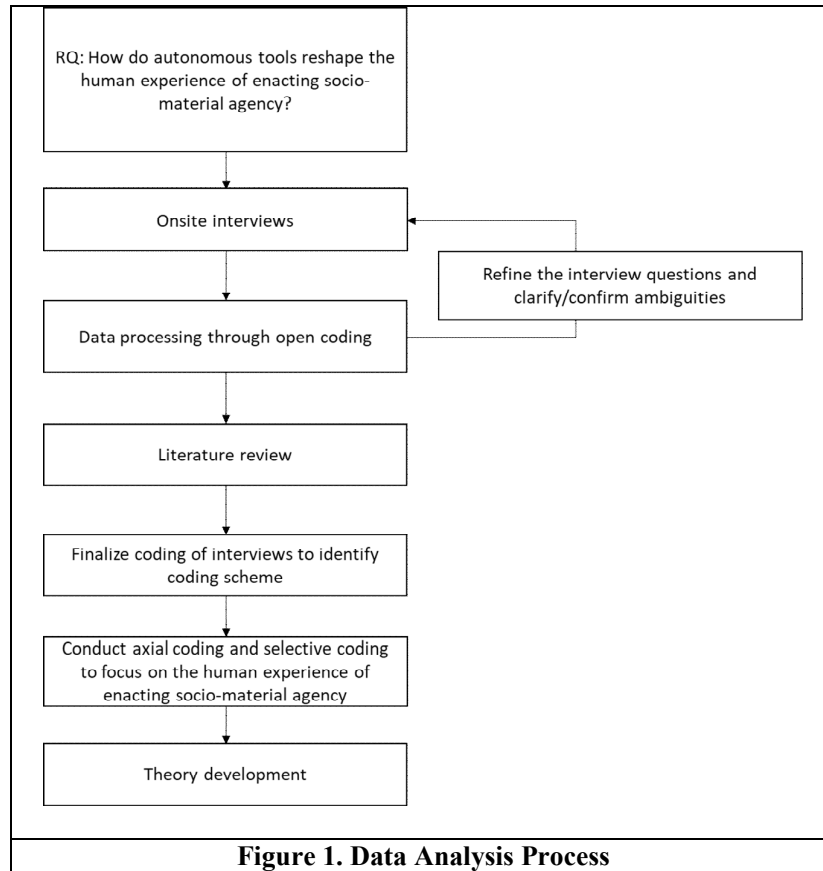
---

<sup>8</sup> We emphasize that this is not a longitudinal study since our research question did not focus on how the two design approaches changed over time.

which were carried out by designers who had also used traditional tools during the course of project Downton).

Given the novelty of the study's context and its comparative nature, we followed grounded theory method for data analysis and theory building (Strauss and Corbin 1998). We relied on interviews as the main data source, not only because of the sensitivity of the highly classified design work conducted at our case site, but also due to the possibility of grasping the *lived experience* of the designers. The confidential nature of the work limited our access to, and presentation of, other sources such as project documentation, project performance data, or direct participant observation. Our choice of interviews as the primary method is in line with Walsham's (1995) argument that interviews offer the "best access" to interpreting the experiences of participants concerning sampled actions and events. In our case, we were concerned with how designers experienced the enactment of design processes leveraging different DA technologies. We adopted an interpretive stance to make sense of the designers' lived experiences and enacted practices. Generally, the approach was guided by a constant cross-checking of emerging concepts against empirical data (Berente and Yoo 2012; Myers 1997). Figure 1 depicts the overall flow of data collection and analysis.





In total, we conducted 9 rounds of on-site interviews, which took place approximately every six months during the four-year research period. The site visits were scheduled to coincide roughly with the end of each main design phase. The interviewees were selected by a “purposeful sampling” strategy (Morse 2007; Strauss and Corbin 1998). We primarily interviewed experienced physical designers who were involved in critical aspects of the physical chip design. To capture various “voices” (Myers and Newman 2007), we also interviewed tool designers and the management team, thereby gaining alternative perspectives. The interviews were conducted in a semi-structured manner with open-ended questions; this gave us and the interviewees flexibility to explore the novel phenomena at hand (Myers and Newman 2007). Appendix B summarizes the interview guide used during our interviews and shows how it was updated as the study progressed. The questions were used as prompts for open-ended conversations. Additional clarifying questions were asked during each interview based on the flow of a particular interview. The

establishment of a baseline of activities allowed us to approach the latter interview stages as “travelers” (Kvale and Brinkmann 2009), thus gaining access to the idiosyncratic experiences and reflections of the designers. Table 2 summarizes the interviews. In total, we conducted 58 interviews.

<b>Table 2. Interviews by project and work domain</b>						
<b>Visit #</b>	<b>Time</b>	<b>Project</b>	<b>Autonomous Tool Designers</b>	<b>Traditional Tool Designers</b>	<b>Others<sup>9</sup></b>	<b>Total</b>
1	2010-11	Maplewood	1 (A <sup>10</sup> )	2 (H, I)	6	9
2	2011-05	Maplewood	3 (A, B, C)	1 (H)	3	7
3	2011-11	Calverton	2 (B, C)	2 (J, K)	1	5
4	2012-05	Calverton	3 (B, C, D)	1 (J)	0	4
5	2012-10	Calverton	2 (B, C)	1 (J)	3	6
6	2013-02	Calverton	2 (B, D)	1 (J)	2	5
7	2013-05	Downton	1 (C)	1 <sup>11</sup> (J)	3	5
8	2014-02	Downton	3 (E, F, G)	3 (J, K, L)	3	9
9	2014-05	Downton	1 (F)	1 (J)	6	8

Overall, we adopted a “dramaturgical model” for our interviews that emphasizes the social interactions between interviewers and interviewees (see Appendix C for details, also (Myers and Newman 2007). At the beginning of each interview, we explained the purpose of our research and told the interviewees that all researchers had signed a non-disclosure agreement, thus ensuring that all conversations would be confidential and the subjects’ anonymity would be protected (Myers and Newman 2007). The interview then moved on to gathering background information about the interviewee, including his or her training and experience. Next we drilled into his or her current work roles, design context, and tasks. We asked the interviewees to take us through their recent design activities, step-by-step, covering the time period between our visits. This allowed us to explore in detail how designers used DA tools in their work and how they collaborated with others. We also asked how physical designers, tool designers, and managers

<sup>9</sup> “Others” includes all other non-physical designers, such as tool designers, architects, and managers.

<sup>10</sup> The letters A to J in the parentheses denote the individual designers we interviewed.

<sup>11</sup> From project Downton and onwards, all the interviewees using traditional tools had switched to using the autonomous tool approach. We have listed them in the autonomous tool designers’ categories as the interviews cover important topics with regard to the difference between the two approaches from the perspective of designers with experience of also using traditional tools.

dealt with the growing complexity of their designs and how this influenced their design practices. All interviews were taped and transcribed verbatim and stored in a central repository for later analysis (Myers and Newman 2007; Yin 2003). To understand the organization of design processes, we traced each designer's design process using graphic representations (Gaskin et al. 2014) and returned to the site to verify the accuracy of our understanding of each design process. We also presented our analyses of design practices and early findings at the site to validate our second-level interpretations. Over the course of the 4 years of the study, and many repeated conversations, we built strong social relationships and a high degree of trust with our interviewees, resulting in genuine and open-ended exchanges (Myers and Newman 2007).

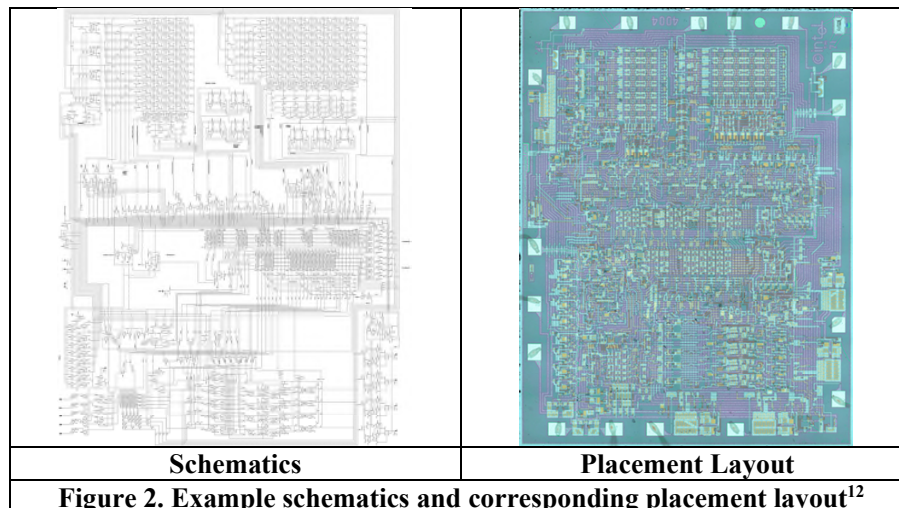
We coded interviews for design practices and the experiences of using tools. In line with this, two of the authors conducted open coding on the transcripts, continuously comparing emerging codes to ensure consistency of the process description (Strauss and Corbin 1998). We followed the three steps adopted by Mazmanian (2012) to ensure the accuracy of our coding. First, both coders read all the transcripts and individually developed two sets of provisional codes (resulting in 592 low-level codes and 34 extensive memos). After this coding was done, we conducted a literature review to search for relevant constructs; for example, it was during the course of this that we came across the concept of "liminality." We found this concept to be useful for characterizing the experiences of the designers we observed, and the concept therefore "earned its way...through demonstrations of its relationship to the phenomenon under investigation" (Corbin and Strauss 1990, p. 9). Second, the two coders, in conjunction with two other co-authors who were not involved in the initial coding, conducted comparative analysis to identify key characteristics of the design practices related to traditional and autonomous tools by detailing how each process unfolded and exploring the designers' experiences of enacting each process. We then forged theoretical categories (e.g., "temporal organization" and "input-output relationships," see Appendix D) to capture how designers experienced the commonalities and differences across the two design approaches. Third, all the transcripts were re-coded according to the consolidated coding scheme for the final round of analysis in which all four co-authors were involved. These accounts were also presented to the designers during the 6<sup>th</sup>, 7<sup>th</sup>, and 8<sup>th</sup> site visits. Based on their feedback, the accounts were revised to ensure they

faithfully reflected the design practices of traditional and autonomous tool designers.

## FINDINGS

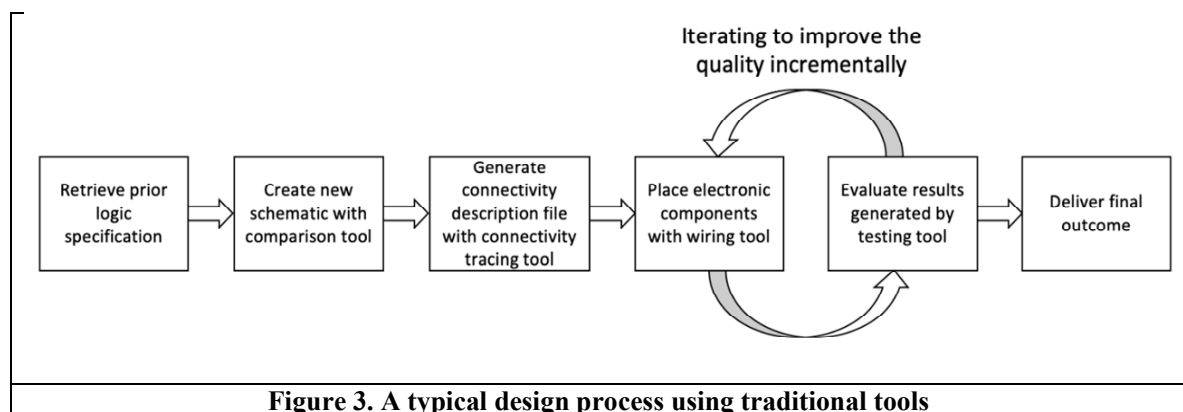
### Design Process Using Traditional Tools

Design processes at ChipCo that leverages traditional tools starts with inheriting an existing design from previous generations of chips. Therefore, a traditional tool designer's first step is to pull the schematic and corresponding logic specification file from the previous generation of chip design and compare it with the new specification received from the logic design team. The designer will attempt to revise and update the old schematic to match the new logic specification files. To produce the new matching schematic, a comparison tool is used to verify the functional equivalence between the schematic and the specification file. It usually takes around a month for the verification process to finalize a new schematic that matches the logic specification file without any issues. This new schematic then becomes the foundation for the initial placement of components on the chip. Figure 2 shows an example of schematics and a corresponding placement layout.



<sup>12</sup> Images are provided as illustrations only. The actual design interfaces may be different in their particulars but similar in their general makeup. The schematics and placement layouts shown were taken from <http://www.4004.com> and are made available under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License.

Once the new schematic is finalized, a connectivity description file<sup>13</sup> can be automatically generated by the connectivity tracing tool. Based on this file, the designer, using traditional wiring tools, begins to create a detailed layout by placing different components on the chip, thereby generating an initial placement. This step is normally rather quick, using a *“fairly tight, tight loop”* (Traditional Tool Designer J) between the connectivity description file on the one hand, and an intermediate “prototype” file<sup>14</sup> on the other hand. This intermediate file contains the physical placement of components on the circuit board and is a more tangible instantiation of the final placement layout compared to the schematic or the connectivity description file. Running the wiring tool is generally quick, about *“a handful of minutes, maybe fifteen minutes”* (Traditional Tool Designer J) but debugging the output and creating new specifications for the tool can take several hours. Once the designer feels satisfied with the placement, she begins to route the wirings between components. This requires her to run timing tests that measure the delays and processing times between distinct parts of the layout, to ensure that the given timing constraints are met. Based on the results generated by the testing tool, the designer revises the layout design on a daily basis by manually modifying the physical component placement or wiring; one designer described this in the following manner: *“rearrange [her] block to put things closer together or line things up for change what metal layer [she is going to be in]”* (Traditional Tool Designer J). Figure 3 shows a typical design process leveraging traditional tools.



<sup>13</sup> This file is called a “netlist,” and specifies the connections between the electronic components (i.e., the network of components, hence the name) described in a given schematic.

<sup>14</sup> This is called “rapid file,” which is a prototype-like layout that allows the designer to test specified requirements.



As the design matures, other requirements, such as power, heat, leakage, and noise are gradually introduced. As a result, it takes longer, anywhere between a couple of hours to half a day, for each iteration to build a new layout. The changes also become subtler as more of the components' placements are fixed: *"In the last two weeks, you fixed sort of the layout and you tried to tweak...like adding in a gauge, changing the size of a gauge and whatnot"* (Traditional Tool Designer J).

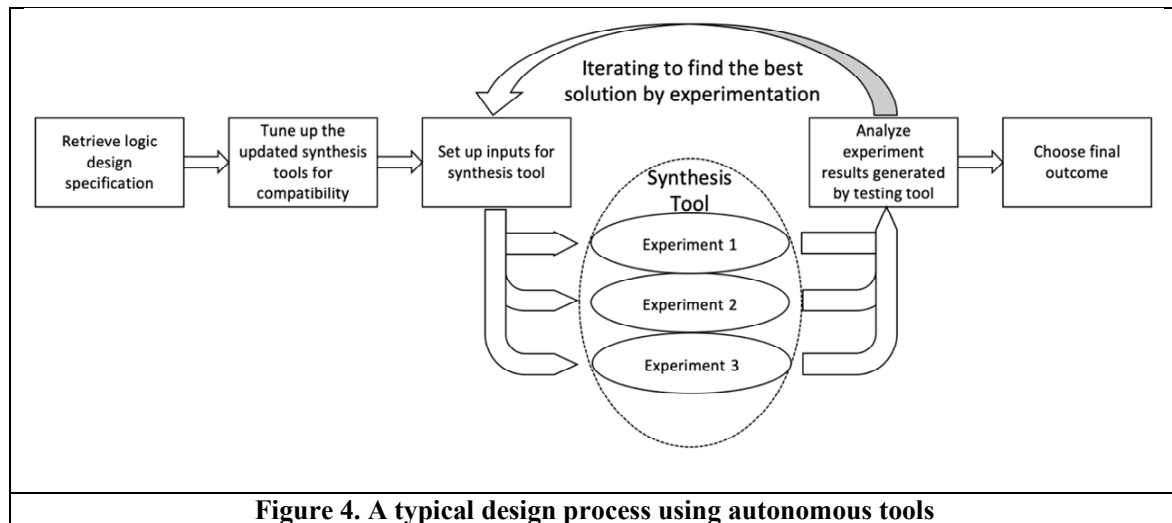
Throughout the design process, designers using traditional tools submit their designs to the validation team every two weeks. The validation team pulls together all the sections of the chip and performs tests to ensure that progress is made with regards to timing and connections between different physical components and areas. As a result of such analyses, it is common for designers to have to go back to the connectivity description file to make some fundamental changes in order to accommodate changes that were separately introduced to a larger section of the chip. Iterations continue until the full chip conforms completely to the logic specification and meets all stated requirements. At this point, the designer hands the final placement design to the integration and validation team.

### **Design Process Using Autonomous Tools**

Just like a traditional tool designer, an autonomous tool designer starts a new project with designs inherited from the previous generation of chips. Since the design process leveraging autonomous tools does not involve creating a schematic, only the previous generation's logic specification files will be retrieved. For designers using autonomous tools, *"the first step is just to get our tools to work"* (Autonomous Tool Designer F). In a design process using autonomous tools, layouts are generated directly from the logic specification. To accomplish this, a synthesis tool, which integrates multiple functionalities, each of which are carried out by individual tools under the traditional DA approach, is used. To facilitate the design of a new chip with increased complexity and constraints, the synthesis tool is continuously improved by adding new features and capabilities—in particular, each generation draws on an improved placement algorithm. Due to these changes, however, it is generally impossible to generate a working layout based on the

inherited logic specification files without crashing the tool a few times. During this time period, the autonomous tool designer works primarily on validating the toolset to solve any compatibility issues and can spend *“a ton of time even just trying to get the tools to start building”* (Autonomous Tool Designer C). The main focus is to create *“cleanup scripts”* to debug the errors caused by incompatibilities between the new version of the tool and the old logic specification files.

After the updated synthesis tool has been tuned and made compatible with the inherited logic specification, the designer is able to start working on generating the physical layout based on the new logic specification. Instead of manually placing electronic components, the autonomous tool designer relies on the tool to synthesize the logic specification directly into a completed placement solution through an iterative search and optimization process. The designer’s primary task is to specify a set of constraints and goals for the tool, which influences the synthesis algorithm’s behavior: *“Work harder on this. Work harder on that. Focus in this area more”* (Autonomous Tool Designer C). Because of the complexity of placing and wiring hundreds of thousands of components, each synthesis run (the designers call it a *“spin”* or *“experiment”*) can take from several days to weeks to finish. Often, such a spin may complete without creating a solution at all. Together with the time required to set up tool constraints and conduct a post-run evaluation, it usually takes about one calendar week to generate a single solution and evaluate its feasibility. To speed up the design process and cut off unfeasible solution trajectories, designers normally run several spins in parallel, each with slightly varied goals and constraints. The basic design principle for processes leveraging the synthesis tool is to *“kick them all off in parallel. When they come back, figure out what the best one is and see if that problem’s been solved”* (Autonomous Tool Designer C). After the synthesis tool generates multiple layout solutions (i.e., multiple versions generated using slightly varying input goals and constraints), each solution will be tested against the design requirements, such as timing constraints, using the same testing tool that is used by traditional tool designers. Based on the test results, the designer chooses the solution that exhibits the best results as the new baseline solution and comes up with more refined and improved constraints for the next round of experiments. Figure 4 shows a typical design process using autonomous tool iterating across three *“experiments.”*



Because of the long running time, designers using autonomous tools normally work on several modular units (called functional unit blocks or FUBs) at once. This stands in contrast to the focus on one or two such units that designers using traditional tools normally maintain. Because autonomous tool designers simultaneously work on multiple units, they tend to stagger their work on each unit in such a way that they can analyze the experimental results and develop the specification for the next experiment of one unit while keeping the other units' experiments running in the background. One designer explained this as such: *"So I own three huge [modular units], and you know I try and stagger them a bit so that there's one finishing you know every other day and then I can look at it, analyze it, figure out what I need to tweak to make it better and then kick it off again before the next [modular unit] finishes"* (Autonomous Tool Designer C).

Comparable to the design process leveraging traditional tools<sup>15</sup>, additional covering tests are gradually introduced into the design process, with timing being the first and most important set of requirements. Once the timing results meet established performance expectations, the designer will begin to include additional tests, such as power, heat, leakage, manufacturability, and noise. The synthesis algorithm runs for increasing spans of time as these additional constraints are introduced. As the project approaches the deadline, a wide range of requirements need to be met. Resynthesizing at this stage can

<sup>15</sup> In fact, the design team is structured based on the chip sections rather than the design approaches. Therefore, a section may be designed by a team of designers using mixed approaches.

cause substantial changes in the layout, thus giving rise to substantial risks and potential delays. To avoid full re-synthesis at these later stages, designers using autonomous tools can switch to “tweaking” the existing design using discrete, manual changes that resemble the hand-editing performed by designers using traditional tools. Just as for the design processes that leverage traditional tools, the validation team will continue to pull intermediate designs from the design database every two weeks to run integration tests on the larger sections of the chip. Once the placement completely matches the logic specification and the design conforms to all requirements, the designer can hand off the final design to the validation team.

### Different Behaviors of Design Tools

As the two illustrations of the design processes indicate, designers using traditional and autonomous tools accomplish similar design goals but follow different processes. These differences are not related to any varying degrees of digitalization since both approaches are fully digitalized. Rather, the differences stem from the underlying logic of how each DA approach allows for framing the design problem, how it guides and supports the search for solutions, and how during this process, human and computational agencies now have different capacities, thus leading to different experiences of enacting socio-material agency. A summary of these differences is provided in Table 3.

<b>Table 3. Differences in design processes leveraging traditional vs. autonomous tools</b>			
<b>Concepts</b>		<b>Traditional tools</b>	<b>Autonomous Tools</b>
<b>Tools’ Behaviors</b>	<b>Computation</b>	Serially executes the designer’s specific wiring and layout commands based on pre-determined computational tasks.	Generates layouts automatically based on the goals and constraints provided by designers.  Non-deterministic and self-learning algorithms featuring built-in randomness.
	<b>Procrastinated binding</b>	Draws upon fixed computational resources to perform pre-determined computational tasks at runtime.  Assumes procrastinated binding but the design outcome is not in principle affected by the availability of computational resources during the runtime due to the simplicity of the computation	Draws upon procrastinated binding of highly distributed and heterogeneous computational resources available at runtime.  The design outcome is significantly affected by the availability of computational resources during the runtime and procrastinated binding
	<b>Runtime</b>	It takes several minutes to update the layout with edits made by the	It takes several days to generate the new layout based on constraints

		designers.	specified by the designers.
	<b>Execution of input</b>	Direct implementation of designers' manual placement and wiring in digital form.	One of many possible solutions found by the algorithm that satisfy the constraints set by the designer.
<b>Designers' Behaviors</b>	<b>Action</b>	Uses the tools to extend cognitive capabilities in ways that help realize the design goals.	Provides goals and constraints to restrict the behavior of the synthesis algorithm.
		The designer directly manipulates the component placement and wiring, with full control over the actual layout.	The designer has no control over the production of the actual layout during solution-generation.
	<b>Design approach</b>	Focus on the recent placement and wiring changes, deciding which exact changes to make, such as the placement of new components or new wiring.	Focus on the set of inputs and constraints that yields the best results from the parallel experiments, while attempting to specify new sets of inputs and constraints that would lead the algorithm to generate a solution that meets all requirements.
	<b>Temporal organization</b>	<i>Iterative:</i> The designer works with iterations that are limited in design scope.  <i>Practical-evaluative:</i> The designer makes decisions with regards to new placements or wirings based on her knowledge of an appropriate solution, which is a direct descendant of the existing layout.  <i>Projective:</i> In each iteration, the designer specifies exactly what changes should be made and expects them to be accurately implemented in the layout.	<i>Iterative:</i> The designer works with iterations that are expansive in design scope.  <i>Practical-evaluative:</i> The designer makes changes by adjusting constraints and then chooses the "best" outcome from multiple layouts generated by previous experiments.  <i>Projective:</i> In each iteration, the designer tunes constraints but cannot fully foresee their effect on the produced layout.
	<b>Input to the tools</b>	Specific placement and wiring of the components.	Specific parameters of the synthesis algorithms.
	<b>Expectation of Input</b>	The components should be placed at the specified locations.	The generation of the layout should follow the rules specified by the input parameters.
	<b>Input-output relationships</b>	Designer understands the tools in terms of their operations and expected effects.	Tools are black-boxed and causally ambiguous to the designer.

Broadly speaking, traditional SDD tools provide capabilities similar to traditional 2D/3D Computer-Aided Design tools (Henderson and Values 1991; Majchrzak et al. 1987). The tools use iconic

representations of a limited set of design elements, which allow designers to place and connect such elements in relation to each other so as to create dependencies between components that are known to produce expected functionalities in the design outcome. Further, traditional tools visualize the physical layout of the IC chip and the connections between the chosen design elements (e.g., gates), and also record and validate designer-initiated changes. Such a style of design is, as one designer expressed it, “*very manual labor-intensive*” (Traditional Tool Designer J).

The traditional tools serially execute commands issued by a designer by producing a corresponding digital representation of the layout. The tools also compute the consequences of such design decisions *vis à vis* given design goals (such as timing). A designer described the work with traditional tools thus: “*instead of the designer going in and grabbing that one and moving it over here and grabbing that, you can give a logical description of, “Grab this group and make it horizontal. And grab this group and make it horizontal. Grab both of those and make them vertical”*” (Tool Designer O). For example, when a designer gives a command to create a path between two components, the traditional design tool will take a few moments to generate the new placement by implementing the change into the existing layout; it will then evaluate the consequences of the action in terms of heat, electrical interference, and timing.

As traditional tools mainly realize designers’ specific commands that produce local changes in the digital layout, the runtimes of manual design tools are relatively short, normally lasting for “*a handful of minutes, maybe fifteen minutes*” (Traditional Tool Designer J). Therefore, traditional tools are less computationally demanding than autonomous tools, and generally work well with available ranges of computing resources.

In contrast, autonomous tools are designed to automatically generate a complete layout of a given design task that “*has all the cells placed, all routed*” (Autonomous Tool Designer F) and are expected to do so each time the tools are run. This task is accomplished through the use of a complex, genetic algorithm<sup>16</sup>. The behavior of the algorithm is guided and restricted by a set of constraints (referred to by

---

<sup>16</sup> A genetic algorithm is a type of heuristic algorithm for which the search process is iterative, based on Darwin’s theory of evolution towards increased fitness (Wang et al. 2009).

autonomous tool designers as “*knobs*”) given by the designer. The search for a feasible solution to the placement of components is a computationally demanding task. In fact, it is theoretically impossible to find the optimal placement within the limited time and resources given to the task<sup>17</sup>. To overcome this theoretical barrier, autonomous tools rely on genetic (heuristic) algorithms that carry out an iterative search process. This process always contains some built-in randomness<sup>18</sup>, and, because the process is self-learning, it takes on evolutionary characteristics<sup>19</sup>:

*“It’s like a genetic algorithm that it’s trying to do. You know like how evolution happens. Like you start throwing... ‘What if I put this gate before here? What if I convert this?’”* (Autonomous Tool Designer F).

These algorithms are currently able to generate “*good enough*” solutions within a reasonable time period (days to weeks) for relatively complex placement problems. The search for a solution, however, may vary greatly from run to run even for the same problem with the same set of constraints. This is because of the underlying, evolutionary computational process, its randomness (such as the initial states that are randomly selected), and the way in which the problem is presented to the algorithm in the form of constraints. Furthermore, each run needs to be supported by extensive computational resources, with the availability of such resources influencing both how long the “spin” runs and how good the solution is. Hence, the search capability of each iteration is constrained by the available resources. For example, the storage required for each autonomous tool run is in the hundreds of gigabytes:

*“you had 16 runs and each one of these runs takes up between 50 gigabytes to 100 to 200 gigabytes of data before you’re done. So you start talking terabytes of data”* (Manager Y).

Therefore, in practice, access to, and the nature of, the computational resources available for generating the

---

<sup>17</sup> Computational wiring is a problem known to be NP-hard (only solvable in non-deterministic polynomial time). This means that, while it is solvable, there are no known algorithms that can find a solution in polynomial time, given the number of inputs, i.e., the number of components and the number of wires between them and their expected length (timing). See <https://en.wikipedia.org/wiki/NP-hardness>

<sup>18</sup> One designer said: “*I always joke that the R in [their abbreviation for the autonomous tool approach] stands for ‘random.’*”

<sup>19</sup> The actual algorithm used by the autonomous tools is proprietary and is therefore kept anonymous.

solution will play a significant role in the effectiveness of the solution search. As one autonomous tool designer noted to us:

*“They found that their tools couldn’t handle it [a modular unit] being that big. It was just too big of a runtime. Some of the tools would just crash because they’d run out of memory trying to design it, and so the tools chopped the thing in half right down the middle, but in a terrible way. The [modular unit] wasn’t designed to be two pieces, and then they cut it down the middle, they had way too many wires crossing that interface”* (Autonomous Tool Designer D).

### **Different Behaviors of Designers**

Traditional tools extend the agency of human designers by providing visualization aids and offering direct manipulation commands for placing components or wiring. The internal operation of the tool is transparent since the tools directly implement the requested command and show the result in a visible manner that is understood by the designer. Designers using traditional tools exercise a large degree of control over how the tool behaves. As one tool designer commented, *“It’s still fully specified by the designer”* (Tool Designer O).

Consequently, the tool maintains a tight relationship with the designer; the tool continuously receives inputs from the designer, prompting specific actions that help the designer to reach her design goal. Designers using traditional tools work in small iterations, each of which lasts only a few minutes. Within each iteration, the designer posits and evaluates possible solutions based on her knowledge and decides which series of commands will best instruct the tools to move towards a solution. Because the commands are strictly implemented and immediately reflected in the layout, a designer using traditional tools is able to predict what outputs will be generated and how the design will be improved. One traditional tool designer noted: *“if you change something, it’s going be more or less the same thing, plus your change”* (Traditional Tool Designer H).

As designers manipulate the layout and search for a plausible solution, each stepwise manipulation is informed by her evolving understanding of the design problem, her past design experiences, and the



current design goals. Because of bounded cognition, designers using traditional tools work only on a limited set of alternatives for an existing layout. Designers extend their capacity by implementing laborious and error-prone design steps using well-established design rules hardwired into the tools. Designers control the design in that they follow a continuous “trajectory” towards a solution, where each subsequent design action builds on the cumulative effects of past decisions:

*“I mean it seems the experience [using traditional tools] seems like I have a better understanding of the details of it, and with such large partitions [using autonomous tools] you have to abstract to a higher level, and you only get to see your work on parts of that if there’s a problem with it. If there’s no problem, then you just move on. Now it seems in the structured part [using traditional tools] it was a little more predictable, or a little easier to see ‘I am here and I need to get here’; and at least for the [autonomous tool] part of it, it’s hard to see, at least for me, ‘This is where I am at and where I need to get and what it will take to get there.’ [That] isn’t as clear.”* (Traditional Tool Designer K)

In contrast, designers using autonomous tools deliver new, incrementally modified constraints to the autonomous tools and expect that the tool will accommodate those modifications to move the layout towards their design goals. An autonomous tool designer expressed this as follows: *“For [autonomous tools], we’re capturing inputs and constraints and the tool goes off and synthesizes it. So, for [autonomous tools], you want to take those inputs and constraints and everything that you feed into the machine and move them forward to the new process”* (Tool Designer O). The tool runs autonomously by generating a candidate layout. Designers have a relatively low degree of insight into the computational process that undergirds the generation of this layout. One autonomous tool designer bluntly said, *“it is logically very much a black box to me”* (Autonomous Tool Designer D). Therefore, each time the autonomous tools are run, the designers expect the generation of a single layout from a set of the many possible layouts that would satisfy the specified constraints.

As it generally takes several days for autonomous tools to generate a complete solution, the design process is characterized by longer iterations of wider scope (i.e., covering a complete modular unit). One

autonomous tool designer commented that *“the whole automated flow is about four days, five days”* (Autonomous Tool Designer C). The built-in randomness of the algorithm combined with the available computational resources means that designers using autonomous tools cannot directly manipulate individual design elements: *“We said we wanted ‘This constraint and output, this side driver and do this’, and it [the tool] said, ‘No. I’ll put something else there,’ and did things like that”* (Tool Designer O). Given the unpredictability of the generated placement solution, designers using autonomous tools typically experiment with multiple, parallel solutions by varying and tweaking the design parameters (i.e., the specified constraints) and then choosing the “best” result as the starting point for the next iteration: *“we kick them all off in parallel. When they come back, figure out what the best one is and see if that problem’s been solved”* (Autonomous Tool Designer C).

Since the algorithms have the capacity to handle an extremely large number of gates and wirings, the granularity of the solution space is no longer bounded by the designer’s cognitive capabilities. In addition, autonomous tools generate layouts that bear little similarity to existing human layouts. In fact, the layouts tend to baffle and surprise human designers. Therefore, the design process no longer follows a singular design trajectory and becomes, rather, a family of discrete and independent design choices and design trajectories, which the designer traverses jointly with the tool. During the course of this process the tool provides evaluative information as to whether a particular trajectory is worth pursuing. One autonomous tool engineer noted:

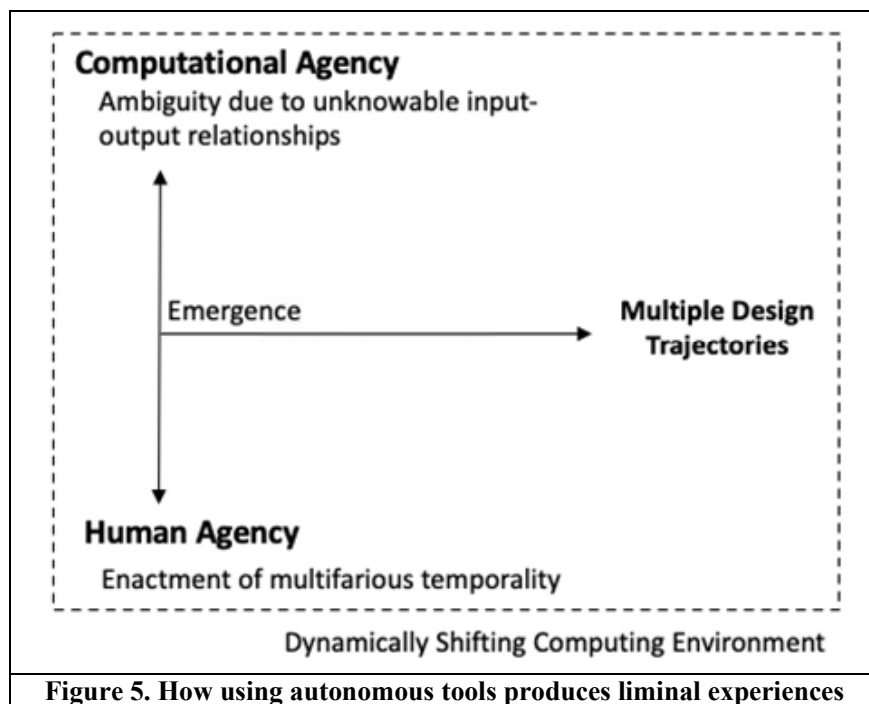
*“I don’t understand what my [section] does nearly as well from a logic standpoint as I used to for my other [sections]. I used to understand. Even [subsections produced by autonomous tools] ten years ago, I knew what they did. I had looked at the logic specification and understood all the different blocks of the specification and what it did and I could probably make hand edits to it myself. Nowadays, no, I don’t. I only know at a high level what it does. I don’t know what each of the different modules actually produces. I don’t know what each of the [sections] is storing... Every time you run the [autonomous tools], it completes the cells in completely different locations. So it can vary from run to run, but even let’s say like it does exactly the same thing and it places two cells really far*

*apart that needed to be closer together” (Autonomous Tool Designer C).*

In sum, the two DA approaches imbue chip design with alternative design logics and lead designers to experience the enactment of socio-material agency differently. Next, we will theorize our empirical findings to better understand these differences.

## DISCUSSION

When designers use autonomous tools, they enact socio-material agency, the experience of which can be characterized as being liminal, i.e., *a state of emergence marked by ambiguity and multifariousness* faced by designers when multiple possible design trajectories co-exist and are continuously being revised and rewoven to move a design artifact forward using tools with unknowable input-output relationships. We summarize our theoretical development in Figure 5 below.



### Emergence

The experience of liminality can be understood as a state of emergence in two interrelated but distinct ways. First, the threshold between human and computational agencies has become fleeting and

nebulous, and therefore the effects of various actions becomes emergent, rather than linear. As expressed by Latour (2005, p. 53), the figurations of agencies, i.e., “the process by which agencies take on observable properties” continuously changes. Specifically, the input-output relationships of the autonomous tool are, in principle, unknowable *ex ante* and *ex post*. As an example, consider the following: when a designer feeds an instruction to an autonomous tool, she is, in principle, unable to predict the exact outcomes of this particular set of instructions. To the autonomous tool, the designers’ instructions are simply one out of many variables that influence the emergence of a particular outcome. Indeed, the self-learning, genetic algorithms powering autonomous tools have built-in randomness, compounded by ongoing interactions with a complex and dynamically shifting computing environment, producing a procrastinated binding that is highly dynamic both in process and outcomes.

Second, the fleeting threshold between human and computational agencies prompts the emergence of multiple potential trajectories (i.e., how the design artifact evolves over time). Multiple such trajectories can and will exist simultaneously and are continuously being revised and rewoven. It becomes difficult to determine why the socio-material agency, as a whole, is enacted the way it is or to what agencies (human or computational) specific actions or outcomes should be attributed. This means that the evolutionary trajectories of the design artifact, i.e., the semiconductor chip that is being designed, continuously exist in a state of emergence (Henfridsson and Yoo 2014). Hence, the overall design trajectory is not formed by the consecutive sediments of past design artifacts, therefore rejecting the idea of strict path dependence of design trajectories (David 1994; David 2001; Liebowitz and Margolis 1995; Van Driel and Dolfsma 2009).

### **Ambiguity**

The ambiguity of the liminal experience faced by designers using autonomous tools is captured by the assessment of designers that it was difficult, if not outright impossible, to directly connect their inputs to the outputs generated by their tools (Reed and DeFillippi 1990). This perception is rooted in the dynamic forms of procrastinated binding performed by such tools (Yoo et al. 2012). Autonomous tools exhibit high degrees of dynamism because of a) the unique, non-deterministic and self-learning nature of the algorithms they run, and b) the dynamically shifting nature of the computing environment within which the tools are

used. First, metaheuristic search, such as the use of the genetic algorithms embedded in autonomous tools, is not fixed during algorithmic execution (Kallinikos et al. 2013). Consequently, algorithmic behavior remains unpredictable. The self-learning nature of such algorithms introduces additional randomness, making it more or less impossible to predict outputs other than in a probabilistic sense. Second, autonomous tools are “computation-heavy” connect with vast, shifting, computing environments. Autonomous tools perform procrastinated binding dynamically by seeking to utilize various amounts of computing resources as well as the latest versions of adjacent artifacts (e.g., the surrounding subsections of a chip) each time they are run. Therefore, each time the algorithm is performed, it potentially faces a different environment and is therefore likely to produce a different result even when the parameter settings are nearly identical.

The high degree of dynamism exhibited by autonomous tools forms the basis of the unknowability of their input-output relationships. As a parallel, consider how (Pickering 1993) described how particle physicists utilize a bubble chamber as an observation instrument. The precise material configuration of the chamber is never fully known to the scientist, but only “temporally emerges in the real time of practice” (p. 575). The resistances of the chamber “appeared as if by *chance – they just happened*. It just happened that when [the scientist] configured his instrument this way (or this, or this) it did not produce tracks, but when he configured it that way, it did” (p. 576, emphasis original). For a scientist the performance of a bubble chamber can only be observed in *real time*. Similarly, the unknowability inherent not only to the computational agency of autonomous tools, but also to the computing environments within which they operate, is experienced by designers in real time.

The computational agency of autonomous tools differs, however, in one significant way from the material agency and its unknowability described by Pickering. Pickering notes that the scientist will eventually gain a deeper understanding through use of the chamber, thus enabling her to precisely account for how and why the chosen material configuration of the chamber worked, i.e., her knowledge of the material agency that she interacts with will grow, at least *ex post*. The designers using autonomous tools, in contrast, will, at most, gain a hazy appreciation of how and why the tool behaves as it does. Because the input-output relationships of such tools are unknowable, both *ex ante* as well as *ex post*, designers will

never become fully knowledgeable with regards to the tool's behaviors.

### **Multifariousness**

The multifariousness of the liminal experience faced by designers using autonomous tools is rooted in the disintegrating connections across the three temporal dimensions of human agency (the past, present, and future). When using autonomous tools, it is harder for humans to project into the future because they cannot reliably predict the trajectories that will be chosen by the autonomous tools. It is also more difficult for them to draw upon the past, because there is less assurance that the past will be iterated upon or inherited from in a predictable manner. This then changes the ways in which action is practically evaluated and executed in the present, i.e., it changes the enactments of human designers.

Ultimately, designers were compelled to “[subject] their own agentic orientations to imaginative recomposition and critical judgment” (Emirbayer and Mische 1998, p. 1010). For example, designers had to constantly shift through multiple past experiments and identify suitable candidate histories that would help them evaluate their present decisions, while projecting appropriate hypotheticals for testing the next round of alternatives. Through experimentation they sought to continuously adapt (Eisenberg 1990) to the dynamic and causally ambiguous behavior of autonomous tools. Designers treated their experiments as “alternative courses of actions [being] tentatively enacted” (Emirbayer and Mische p. 988), indicating that multiple parallel design trajectories existed simultaneously.

Hence, when human agents use autonomous tools, they are likely to experience the enactment of temporality as being *multifarious*. This is because agents using autonomous tools face what Pickering (1993) refers to as the temporally emergent structure of a problem space where the “contours of agency” are unknowable *ex ante*, prompting human agents to continually seek to understand the workings of computational agency and the outputs that are generated, even if such outputs are only temporarily stabilized (Pickering 1993, p. 564). Working under such conditions, human agents can never reasonably connect the past to the future so as to evaluate the progress of their task. Rather, the past becomes a contested string of multiple possible parallel histories, each with its own set of evaluative criteria, and each offering a partial and sometimes fleeting insight into the structure of the overall solution space. This historicity

connects the past, present, and future along multiple trajectories. The human agent's temporal orientation is conditioned and dominated by the need to make sense of the unpredictable computational agency of autonomous tools and their parallel histories. The process and interactions between the human designer and the autonomous tool are woven together from multiple discrete present moments that are rarely directly connected. Each moment holds multiple alternative conjectures with regard to how the design process may unfold, only to disappear into the background as new alternative histories are discovered. Since the human agent can never fully comprehend the overwhelmingly large solution spaces, the manner in which autonomous tools move each design trajectory forward appears to the human agents as a random walk. Each successive iteration is associated with the possibility of new probes, each of which may yield a sliver of understanding when it is combined with the already existing fount of prior trajectories.

### **THEORETICAL IMPLICATIONS**

Our findings have several implications for how to think about the impacts of increased autonomy of digital technologies, for studies of design and digital innovation, as well how we conceive of the broader usage of autonomous tools in our contemporary world.

Contemporary theorizing around the use of digital artifacts has identified several characteristics that digital artifacts share but which are lacking in physical artifacts. These include ontological ambivalence (Kallinikos et al. 2013), generativity (Zittrain 2006), re-programmability (Yoo et al. 2010), communicability (Yoo 2010), and so forth. Increasingly, technologies such as Apple's Siri, Amazon's Alexa, and Google Now are used to enhance experiences of using consumer products (such as cars, TV, etc.). The emergence of these services points to an urgent need to understand the process and consequences of digital innovations that feature a growing number of autonomous capabilities. Scholars have begun to explore the level and nature of the autonomy afforded by artificial intelligence technologies that now act as components in many technological systems, as well as within the context of designing digital artifacts (Callon and Muniesa 2005; Mackenzie 2006; Orlikowski and Scott 2015). Exploring the liminal experiences of enacting socio-material agency can add to the research on digital innovation by articulating

more clearly the emergent properties of interactions across data, algorithms, computing infrastructures, and attendant human practices.

The liminal experience of enacting socio-material agency associated with autonomous tools suggests that we may need to expand our ontological and epistemological perspectives to understand and account for the independent runtime actions of autonomous tools imbued with artificial intelligence and machine learning. If we are to understand the complex structures and dynamics that emerge from autonomous systems that operate under conditions of dynamic, procrastinated binding, a multiplicity of perspectives is required. We may, for example, draw on recent advances in assemblage theory (DeLanda 2013; DeLanda 2016; Müller 2015) that represent a renewed interest in materialism as a philosophical perspective, drawing heavily on theories of evolution and complex adaptive systems. Applying such a perspective on socio-material agency may help forge a more nuanced theory of procrastinated binding that explicates how various elements of data, algorithms, and processing capacities interact within dynamic computing environments to produce emergent outcomes.

Further, our finding that autonomous tools approach design problems in a non-linear way represents a challenge to received notions of design, such as the dominant idea of linear search and its associated costs (Simon 1996). Simon's original theory neither assumed nor discussed "jumps" or "indeterminacy" in theorizing how design solutions are searched for and "computed." Nor did he discuss the possibility of parallelism in the search for a solution. Our findings suggest a renewed focus within IS and organizational studies on the role of computational technologies in accomplishing complex cognitive activities, such as those found in design processes. The use and command of autonomous tools are becoming increasingly common across design professions and they will challenge the inherited conceptual and theoretical frameworks we have developed to understand their effects (Seidel et al 2018a). These technologies fundamentally disrupt the idea of design, leaving us with a wide array of unanswered questions. For example, we need to ask: What is the nature of the loose couplings embedded in consecutive generations of a design artifact? What is the role of the cognitive strategies and heuristics that designers use to specify experiments while learning simultaneously from multiple search histories? In sum, this suggests that stage



models (Royce 1970) are insufficient to capture the nature and temporality of design driven by autonomous tools, no matter how refined the model (Boehm 1988) and irrespective of the degree to which it can account for the presence of iteration in a given trajectory (Lyytinen and Berente 2017), its rhythm (McGrath and Kelly 1986), or its speed (Leonardi 2011). Therefore, if we wish to grasp how design changes when autonomous tools are used to support design tasks, further research is required.

Additionally, traditional conceptions of design are based on the idea that the primary design agency lies with human designers. Designers draw upon “kernel theories” (Gregor and Jones 2007; Walls et al. 1992) that help scaffold the mental imagery of the preferred design conditions—what the artifact should be and do in its finished state—and the means for achieving them. This gives rise to a convergent process in which the preferred solution is incrementally crafted within the problem space through designer-controlled moves. This idea is firmly embedded in the idea of the purified agency of the designer, who uses technologies as extensions of herself and therefore sees technologies as “present-at-hand<sup>20</sup>” (Heidegger 1962; Sennett 2008).

The design processes supported by autonomous tools unfold quite differently. Rather than starting out with a kernel theory, a designer may start by outlining the task’s goals and constraints before engaging with varying outputs of autonomous tools driven by their generalized search capabilities. The autonomous tool, in a sense, continuously abduces (Paavola 2005), i.e., induces and formulates reasonable working hypotheses with regards to the solution. In doing so, the tool may well discover new variants of the kernel theory, but this occurs through a process that is black boxed from the perspective of the designer. The search is composed of a continuous series of discontinuous adjustments between the designer, the tool, and the solution space. As a result, designers will increasingly have to move away from their traditional role of artisans endowed with deep knowledge of their tools and materials (Sennett 2008) through which they realize specific “preferred conditions” (Simon 1996). Instead, designers must become experimentalists, who probe an uncertain, dynamic world, creating, in the process, provisional theories about how both their

---

<sup>20</sup> Present-at-hand, *Vorhandenheit* in German, refers to a mode of being of an agent who is merely looking at or observing something. It reflects the disembodied understanding of an object through a theoretical gaze.

tools and the artifacts they design will work. This suggests that designers are not being replaced by autonomous tools, but rather that the role of the designer *qua* designer is changing. Hence, we are observing a fundamental shift with regard to how design and innovation work is organized which prompts us to reconsider received theories of design. Recently, Seidel et al. (2018a) suggested that designers need to approach this type of computational agency via multiple interrelated knowledge practices, namely: parameterization; process analysis; and the constant modification of algorithms. It is clear that more research is needed to understand how designers contend with the liminal experiences of enacting socio-material agency engendered by autonomous tools, where the tools are seen as “ready-to-hand”<sup>21</sup> (Heidegger 1962; Sennett 2008).

Our insights with regard to the liminal experiences of enacting socio-material agency while using autonomous tools extend beyond the domains of design and digital innovation. An increasing number of applications are imbued with autonomous features that may lead to the types of liminal experiences that we captured in our case study. For example, consider a situation of riding in an autonomous car, or using driver assistance technologies. In such a situation the liminal experience of enacting socio-material agency is brought to the forefront—why does the car (i.e., the technical system) behave the way that it does? Why does the autonomous driving system react in particular ways to certain human inputs? These are the types of questions that we will increasingly need to ask as broader ranges of human practices are mediated by autonomous tools. Indeed, the liminal experience of using such tools forces us to confront issues such as technology-based trust (Jarvenpaa et al. 1998; Jarvenpaa and Leidner 1999) and ethics (Banerjee et al. 1998; Smith and Hasnas 1999). This suggests that we need to find ways of making autonomous tools more transparent and understandable as has been the goal within “explainable AI” (Hagras 2018; Pasquale 2015; Zhu et al. 2018).

---

<sup>21</sup> Ready-to-hand, *Zuhandenheit* in German, refers to a mode of being of an agent involved in the world, acting on an object to achieve something. An object that is ready-to-hand comes into being when it fits into a meaningful part of a purposeful action.

## BOUNDARY CONDITIONS

Our study is a single-site, comparative case study, and therefore confers limited generalizability (Lee and Baskerville 2003) on to our findings. It will be important for the IS field to encourage similar studies in other settings and to examine the degree to which the results may be generalized across contexts and technologies, as well as to identify important contextual elements and boundary conditions. For example, our study focused on the design of semiconductor chips, which as a design activity resides at the more complex end of the spectrum while also assuming well-defined and measurable design goals. This task is also highly amenable to complex algorithmic solutions. This may, potentially, decrease the degree of knowability that designers experience, compared to design contexts that are simpler. It would therefore be interesting to understand how autonomous tools interact with human agency in other contexts, such as the AI-driven design of websites or self-driving cars, drones, and unmanned vehicles.

An important intersection between autonomous tools and related data-generating capacities is emerging in the form of online crowds (Orlikowski and Scott 2015). The vast datasets that crowds generate have been crucial for training machine-learning algorithms, enabling them to make fine-grained and individualized predictions as exemplified by the recommendation engines used by the likes of Amazon and Netflix. Because of the nature of the context we examined in this paper, we cannot account for such interactions. It is, however, clearly an important factor that exacerbates the dynamic and unpredictable nature of computation, and we urge scholars to examine the interactions between autonomous tools and crowd-generated data.

Autonomous tools and the liminal experiences of enacting socio-material agency will force us to reconsider how complex knowledge work is organized (Puranam et al. 2014). For example, in our study we did not analyze the effects of liminality on the coordination and composition of (design) routines (Gaskin et al. 2014; Lindberg et al. 2016)—two crucial aspects of organizing design and other professional work teams. We also need to consider issues of trust (Jarvenpaa et al. 1998; Jarvenpaa and Leidner 1999), power (Levina and Arriaga 2014), privacy and transparency (Pasquale 2015), as well as ethics (Smith and Hasnas 1999) as they relate to use of autonomous tools, since each of these focal phenomena will be irrevocably

changed as liminality enters into domains that have traditionally been dominated by tools with knowable input-output relationships. The insights provided herein open the door to a host of studies that have the potential to change the way we think about organizing when using autonomous tools.

### CONCLUSION

As autonomous tools move digitalization beyond “paving the cow paths” of design work, future designers will need to drop their identification as artisans (Sennett 2008) and become more akin to laboratory scientists, who explore multiple, diffuse design trajectories and related working hypotheses (Pickering 1993). The lack of knowability of autonomous tools pulls us out of the convenient and familiar industrial-era assumptions through which we understand the composition and workings of most socio-technical systems and propels us into unknown territory. It is now incumbent on both scholars and practitioners to make significant efforts to understand how to live and work in a world where the enactment of socio-material agency will increasingly be experienced as liminal.

### REFERENCES

- Archer, M. S., Bhaskar, R., Collier, A., Lawson, T., and Norrie, A. 1998. *Critical Realism: Essential Readings*. Psychology Press.
- Banerjee, D., Cronan, T. P., and Jones, T. W. 1998. "Modeling It Ethics: A Study in Situational Ethics," *Mis Quarterly*), pp. 31-60.
- Barad, K. 2003. "Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter," *Signs: Journal of women in culture and society* (28:3), pp. 801-831.
- Beane, M., and Orlikowski, W. J. 2015. "What Difference Does a Robot Make? The Material Enactment of Distributed Coordination," *Organization Science* (26:6), pp. 1553-1573.
- Berente, N., and Yoo, Y. 2012. "Institutional Contradictions and Loose Coupling: Postimplementation of Nasa's Enterprise Information System," *Information Systems Research* (23:2), pp. 376-396.
- Bijker, W. E., Hughes, T. P., and Pinch, T. J. 1987. *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*. MIT press.
- Boehm, B. W. 1988. "A Spiral Model of Software Development and Enhancement," *Computer* (21:5), pp. 61-72.
- Boland, R. J., Tenkasi, R. V., and Te'eni, D. 1994. "Designing Information Technology to Support Distributed Cognition," *Organization Science* (5:3), pp. 456-475.
- Bourdieu, P. 1998. *Practical Reason: On the Theory of Action*. Stanford University Press.
- Brown, C., and Linden, G. 2009. *Chips and Change: How Crisis Reshapes the Semiconductor Industry*. MIT Press.
- Burrell, J. 2016. "How the Machine ‘Thinks’: Understanding Opacity in Machine Learning Algorithms," *Big Data & Society* (3:1), p. 2053951715622512.

- Callon, M., and Muniesa, F. J. O. s. 2005. "Peripheral Vision: Economic Markets as Calculative Collective Devices," (26:8), pp. 1229-1250.
- Corbin, J. M., and Strauss, A. 1990. "Grounded Theory Research: Procedures, Canons, and Evaluative Criteria," *Qualitative sociology* (13:1), pp. 3-21.
- David, P. A. 1994. "Why Are Institutions the 'Carriers of History'? Path Dependence and the Evolution of Conventions, Organizations and Institutions," *Structural Change and Economic Dynamics* (5:2), pp. 205-220.
- David, P. A. 2001. "Path Dependence, Its Critics and the Quest for 'Historical Economics'," *Evolution and path dependence in economic ideas: Past and present* (15), p. 40.
- DeLanda, M. 2013. *Intensive Science and Virtual Philosophy*. Bloomsbury Publishing.
- DeLanda, M. 2016. *Assemblage Theory*. Edinburgh University Press.
- Denning, P. J., and Lewis, T. G. J. A. S. 2017. "Computers That Can Run Backwards," (105:5), p. 270.
- Dodgson, M., Gann, D. M., and Salter, A. 2007. "'In Case of Fire, Please Use the Elevator': Simulation Technology and Organization in Fire Engineering," *Organization Science* (18:5), pp. 849-864.
- Eisenberg, E. M. J. C. R. 1990. "Jamming: Transcendence through Organizing," (17:2), pp. 139-164.
- Emirbayer, M., and Mische, A. 1998. "What Is Agency?," *American Journal of Sociology* (103:4), pp. 962-1023.
- Faraj, S., Kwon, D., and Watts, S. 2004. "Contested Artifact: Technology Sensemaking, Actor Networks, and the Shaping of the Web Browser," *Information Technology & People*.
- Faraj, S., Pachidi, S., and Sayegh, K. 2018. "Working and Organizing in the Age of the Learning Algorithm," *Information and Organization* (28:1), pp. 62-70.
- Fayard, A.-L., and Weeks, J. 2007. "Photocopiers and Water-Coolers: The Affordances of Informal Interaction," *Organization studies* (28:5), pp. 605-634.
- Garsten, C. 1999. "Betwixt and Between: Temporary Employees as Liminal Subjects in Flexible Organizations," *Organization studies* (20:4), pp. 601-617.
- Gaskin, J. E., Berente, N., Lyytinen, K., and Yoo, Y. 2014. "Toward Generalizable Sociomaterial Inquiry: A Computational Approach for Zooming in and out of Sociomaterial Routines.," *MIS Quarterly* (38), pp. 849-871.
- Gibson, J. J. 1977. "*The Theory of Affordances*," in *Perceiving, Acting, and Knowing. Towards an Ecological Psychology*. Hoboken, NJ: John Wiley & Sons Inc.
- Gibson, J. J. 1979. *The Ecological Approach to Visual Perception*. Psychology Press.
- Giddens, A. 1984. *The Constitution of Society: Outline of the Theory of Structuration*. University of California Press.
- Gregor, S., and Jones, D. J. J. o. t. A. f. I. s. 2007. "The Anatomy of a Design Theory," (8:5).
- Hagras, H. 2018. "Toward Human-Understandable, Explainable Ai," *Computer* (51:9), pp. 28-36.
- Hanseth, O., and Monteiro, E. 1997. "Inscribing Behaviour in Information Infrastructure Standards," *Accounting, management and information technologies* (7:4), pp. 183-211.
- Heidegger, M. 1962. *Being and Time*. New York: Harper & Row.
- Henderson, K. J. S., Technology, and Values, H. 1991. "Flexible Sketches and Inflexible Data Bases: Visual Communication, Conscription Devices, and Boundary Objects in Design Engineering," (16:4), pp. 448-473.
- Henfridsson, O., and Yoo, Y. 2014. "The Liminality of Trajectory Shifts in Institutional Entrepreneurship," *Organization Science* (25:3), pp. 932-950.

- High, R. 2012. "The Era of Cognitive Systems: An inside Look at Ibm Watson and How It Works.," *IBM Corporation, Redbooks*).
- Hirschheim, R. 1985. "Information Systems Epistemology: An Historical Perspective," *Research methods in information systems*), pp. 13-35.
- Holland, J. H., Holyoak, K. J., Nisbett, R. E., and Thagard, P. R. 1989. *Induction: Processes of Inference, Learning, and Discovery*. MIT press.
- Hutchins, E. 1995. *Cognition in the Wild*. MIT press Cambridge, MA.
- Hutchins, E., Klausen, T. J. C., and work, c. a. 1996. "Distributed Cognition in an Airline Cockpit,"), pp. 15-34.
- Introna, L. D. 2011. "The Enframing of Code: Agency, Originality and the Plagiarist," *Theory, Culture & Society* (28:6), pp. 113-141.
- Jarvenpaa, S. L., Knoll, K., and Leidner, D. E. 1998. "Is Anybody out There? Antecedents of Trust in Global Virtual Teams," *Journal of management information systems* (14:4), pp. 29-64.
- Jarvenpaa, S. L., and Leidner, D. E. 1999. "Communication and Trust in Global Virtual Teams," *Organization science* (10:6), pp. 791-815.
- Jung, Y., and Lyytinen, K. 2014. "Towards an Ecological Account of Media Choice: A Case Study on Pluralistic Reasoning While Choosing Email," *Information Systems Journal* (24:3), pp. 271-293.
- Kallinikos, J., Aaltonen, A., and Marton, A. J. M. Q. 2013. "The Ambivalent Ontology of Digital Artifacts,"), pp. 357-370.
- Kavanagh, D., and Araujo, L. 1995. "Chronigami: Folding and Unfolding Time," *Accounting, Management and Information Technologies* (5:2), pp. 103-121.
- Kling, R., and Scacchi, W. 1982. "The Web of Computing: Computer Technology as Social Organization," in *Advances in Computers*. Elsevier, pp. 1-90.
- Kroll, J. A. 2018. "The Fallacy of Inscrutability," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* (376:2133), p. 20180084.
- Kvale, S., and Brinkmann, S. 2009. *Interviews: Learning the Craft of Qualitative Research Interviewing*. Sage.
- Latour, B. 1987. *Science in Action: How to Follow Scientists and Engineers through Society*. Harvard university press.
- Latour, B. 2012. *We Have Never Been Modern*. Harvard university press.
- Latour, B., Mauguin, P., and Teil, G. J. S. S. o. S. 1992. "A Note on Socio-Technical Graphs," (22:1), pp. 33-57.
- Latour, B. J. P. y. S. 2005. "Reassembling the Social," (43:3), pp. 127-130.
- Leavitt, H. J., March, J. G., and March, J. G. 1962. *Applied Organizational Change in Industry: Structural, Technological and Humanistic Approaches*. Pittsburgh: Carnegie Institute of Technology, Graduate School of Industrial Administration.
- Lee, A. S., and Baskerville, R. L. 2003. "Generalizing Generalizability in Information Systems Research," *Information systems research* (14:3), pp. 221-243.
- Leonardi, P. 2013. "Theoretical Foundations for the Study of Sociomateriality," *Information and organization* (23:2), pp. 59-76.
- Leonardi, P. M. 2011. "When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Agencies," *MIS quarterly*), pp. 147-167.
- Levina, N., and Arriaga, M. 2014. "Distinction and Status Production on User-Generated Content Platforms: Using Bourdieu's Theory of Cultural Production to Understand Social

- Dynamics in Online Fields," *Information Systems Research* (25:3), pp. 468-488.
- Liebowitz, S. J., and Margolis, S. E. 1995. "Path Dependence, Lock-in, and History," *Journal of Law, Economics, & Organization*, pp. 205-226.
- Lindberg, A., Berente, N., Gaskin, J., and Lyytinen, K. 2016. "Coordinating Interdependencies in Online Communities: A Study of an Open Source Software Project," *Information Systems Research* (27:4), pp. 751-772.
- Lyytinen, K., and Berente, N. 2017. "Iteration in Systems Analysis and Design: Cognitive Processes and Representational Artifacts," in *Systems Analysis and Design: Techniques, Methodologies, Approaches, and Architecture*. Routledge, pp. 51-71.
- Lyytinen, K., Nickerson, J. V., and King, J. L. 2020. "Metahuman Systems = Humans + Machines That Learn," *Journal of Information Technology* (0:0), p. 0268396220915917.
- Mackenzie, A. 2006. *Cutting Code: Software and Sociality*. Peter Lang.
- MacKenzie, D. 2018. "Material Signals: A Historical Sociology of High-Frequency Trading," *American Journal of Sociology* (123:6), pp. 1635-1683.
- MacKenzie, D. 2019. "How Algorithms Interact: Goffman's 'Interaction Order' in Automated Trading," *Theory, Culture & Society* (36:2), pp. 39-59.
- Majchrzak, A., Chang, T.-C., Barfield, W., Eberts, R., and Salvendy, G. 1987. *Human Aspects of Computer-Aided Design*. Bristol, PA, USA: Taylor & Francis/Hemisphere.
- Mangalaraj, G., Nerur, S., Mahapatra, R., and Price, K. H. 2014. "Distributed Cognition in Software Design: An Experimental Investigation of the Role of Design Patterns and Collaboration," *MIS Quarterly* (38:1), pp. 249-274.
- Marx, K. 1945. "Capital: A Critique of Political Economy.,").
- Mazmanian, M. 2012. "Avoiding the Trap of Constant Connectivity: When Congruent Frames Allow for Heterogeneous Practices," *Academy of Management Journal*.
- McGrath, J. E., and Kelly, J. R. 1986. *Time and Human Interaction: Toward a Social Psychology of Time*. Guilford Press.
- Morse, J. M. 2007. "Sampling in Grounded Theory," in *The Sage Handbook of Grounded Theory*, A. Bryant and K. Charmaz (eds.). SAGE Publications Ltd, pp. 229-244.
- Müller, M. 2015. "Assemblages and Actor-Networks: Rethinking Socio-Material Power, Politics and Space," *Geography Compass* (9:1), pp. 27-41.
- Mutch, A. 2013. "Sociomateriality—Taking the Wrong Turning?," *Information and Organization* (23:1), pp. 28-40.
- Myers, M. D. 1997. "Qualitative Research in Information Systems," *Management Information Systems Quarterly* (21), pp. 241-242.
- Myers, M. D., and Newman, M. 2007. "The Qualitative Interview in IS Research: Examining the Craft," (17:1), pp. 2-26.
- Nan, N., and Lu, Y. 2014. "Harnessing the Power of Self-Organization in an Online Community During Organizational Crisis," *MIS Quarterly* (38:4), pp. 1135-1158.
- Norman, D. A. 1990. *The Design of Everyday Things*. New York: Doubleday.
- Orlikowski, W., and Scott, S. V. 2015. "The Algorithm and the Crowd: Considering the Materiality of Service Innovation,").
- Orlikowski, W. J. 2007. "Sociomaterial Practices: Exploring Technology at Work," *Organization studies* (28:9), pp. 1435-1448.
- Orlikowski, W. J., and Scott, S. V. 2014. "What Happens When Evaluation Goes Online? Exploring Apparatuses of Valuation in the Travel Sector," *Organization Science* (25:3), pp. 868-891.

- Paavola, S. 2005. "Peircean Abduction: Instinct or Inference?," *Semiotica* (2005:153-1/4), pp. 131-154.
- Pasquale, F. 2015. *The Black Box Society*. Harvard University Press.
- Pickering, A. 1993. "The Mangle of Practice: Agency and Emergence in the Sociology of Science," *American journal of sociology* (99:3), pp. 559-589.
- Pinch, T. J., and Bijker, W. E. 1984. "The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other," *Social Studies of Science* (14:3), pp. 399-441.
- Puranam, P., Alexy, O., and Reitzig, M. 2014. "What's "New" About New Forms of Organizing?," *Academy of Management Review* (39:2), pp. 162-180.
- Reed, R., and DeFillippi, R. J. 1990. "Causal Ambiguity, Barriers to Imitation, and Sustainable Competitive Advantage," *Academy of management review* (15:1), pp. 88-102.
- Royce, W. W. 1970. "Managing the Development of Large Software Systems: Concepts and Techniques," *Proceedings of the 9th international conference on Software Engineering*, pp. 328-338.
- Sarker, S., Chatterjee, S., Xiao, X., and Elbanna, A. 2019. "The Sociotechnical Axis of Cohesion for the Is Discipline: Its Historical Legacy and Its Continued Relevance," *Mis Quarterly* (43:3), pp. 695-720.
- Seidel, S., Berente, N., Lindberg, A., Lyytinen, K., and Nickerson, J. V. 2018a. "Autonomous Tools and Design: A Triple-Loop Approach to Human-Machine Learning," *Communications of the ACM* (62:1), pp. 50-57.
- Seidel, S., Berente, N., Martinez, B., Lindberg, A., Lyytinen, K., and Nickerson, J. V. 2018b. "Autonomous Tools in System Design: Reflective Practice in Ubisoft's Ghost Recon Wildlands Project," *Computer* (51:10), pp. 16-23.
- Sennett, R. 2008. *The Craftsman*. New Haven, Conn.: Yale University Press.
- Shaft, T. M., and Vessey, I. 2006. "The Role of Cognitive Fit in the Relationship between Software Comprehension and Modification," *Mis Quarterly*, pp. 29-55.
- Shen, Z., Lyytinen, K., and Yoo, Y. 2015. "Time and Information Technology in Teams: A Review of Empirical Research and Future Research Directions," *European Journal of Information Systems* (24:5), pp. 492-518.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., and Lanctot, M. J. n. 2016. "Mastering the Game of Go with Deep Neural Networks and Tree Search," (529:7587), p. 484.
- Simon, H. A. 1996. *The Sciences of the Artificial*. Boston, MA: MIT Press.
- Smith, H. J., and Hasnas, J. 1999. "Ethics and Information Systems: The Corporate Domain," *Mis Quarterly*, pp. 109-127.
- Strauss, A., and Corbin, J. 1998. "Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory," 2<sup>nd</sup> Ed).
- Suchman, L. A. 2006. *Human-Machine Reconfigurations*, (2 edition ed.). Cambridge ; New York: Cambridge University Press.
- Thomas, D. E., Hitchcock Iii, C. Y., Kowalski, T. J., Rajan, J. V., and Walker, R. A. 1983. "Automatic Data Path Synthesis," *IEEE Computer* (16:12), pp. 59-70.
- Tilson, D., Lyytinen, K., and Sørensen, C. J. I. s. r. 2010. "Research Commentary—Digital Infrastructures: The Missing Is Research Agenda," (21:4), pp. 748-759.
- Tong, C., and Sriram, D. 1992. *Artificial Intelligence in Engineering Design: Volume Iii: Knowledge Acquisition, Commercial Systems, and Integrated Environments*. Elsevier.



- Trist, E. 1981. "The Socio-Technical Perspective. The Evolution of Socio-Technical Systems as a Conceptual Framework and as an Action Research Program," in *Occasional Paper*. John Wiley & Sons, pp. 49-75.
- Trist, E. L., and Bamforth, K. W. 1951. "Some Social and Psychological Consequences of the Longwall Method of Coal-Getting: An Examination of the Psychological Situation and Defences of a Work Group in Relation to the Social Structure and Technological Content of the Work System," *Human relations* (4:1), pp. 3-38.
- Turkle, S. 1997. "Seeing through Computers," *The American Prospect* (8:31), pp. 76-82.
- Turner, V. 1987. "Betwixt and Between: The Liminal Period in Rites of Passage," *Betwixt and between: Patterns of masculine and feminine initiation*, pp. 3-19.
- Van Driel, H., and Dolfma, W. 2009. "Path Dependence, Initial Conditions, and Routines in Organizations: The Toyota Production System Re-Examined," *Journal of Organizational Change Management* (22:1), pp. 49-72.
- Wagner, E. L., Newell, S., and Kay, W. 2012. "Enterprise Systems Projects: The Role of Liminal Space in Enterprise Systems Implementation," *Journal of Information Technology* (27:4), pp. 259-269.
- Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. J. I. s. r. 1992. "Building an Information System Design Theory for Vigilant Eis," (3:1), pp. 36-59.
- Walsham, G. J. E. J. o. i. s. 1995. "Interpretive Case Studies in Is Research: Nature and Method," (4:2), pp. 74-81.
- Wang, L.-T., Chang, Y.-W., and Cheng, K.-T. T. 2009. *Electronic Design Automation: Synthesis, Verification, and Test*. Morgan Kaufmann.
- Winograd, T., and Flores, F. 1986. *Understanding Computers and Cognition: A New Foundation for Design*. Intellect Books.
- Xiao, X., Lindberg, A., Hansen, S., and Lyytinen, K. J. J. o. t. A. f. I. S. 2018. "" Computing" Requirements for Open Source Software: A Distributed Cognitive Approach," (19:12), pp. 1217-1252.
- Yin, R. K. 2003. *Case Study Research: Design and Methods*. SAGE.
- Yoo, Y. 2010. "Computing in Everyday Life: A Call for Research on Experiential Computing," *Mis Quarterly* (34:2), pp. 213-231.
- Yoo, Y., Boland Jr, R. J., Lyytinen, K., and Majchrzak, A. 2012. "Organizing for Innovation in the Digitized World," *Organization science* (23:5), pp. 1398-1408.
- Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. "Research Commentary-the New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information Systems Research* (21), pp. 724-735.
- Zhu, J., Liapis, A., Risi, S., Bidarra, R., and Youngblood, G. M. 2018. "Explainable Ai for Designers: A Human-Centered Perspective on Mixed-Initiative Co-Creation," *2018 IEEE Conference on Computational Intelligence and Games (CIG)*: IEEE, pp. 1-8.
- Zittrain, J. L. 2006. "The Generative Internet," *Harvard Law Review*, pp. 1974-2040.

## **APPENDIX A: A BRIEF OVERVIEW OF CHIP DESIGN**

Since the invention of the IC in the early 60's, chip performance has grown exponentially (Denning and Lewis 2017). Semiconductor manufacturers have had to cope with the growing complexity of ICs through continuous advancement in DA. Chip design improvements have, for the past 30 years, been driven largely by DA innovations that have improved designers' productivity to reach the levels necessary to produce increasingly complex chips without significantly increasing the number of engineers. During this period, the scope of DA has shifted from supporting a single design step (such as placing a gate or a set of gates, each carrying a specific Boolean function) and connecting those gates to implement a given set of Boolean functions to support an expansive set of design tasks, such as generating schematic scaffolds for rough gate placement, validating support for logic and power, supporting the overall workflow, and assisting with project coordination. The recent trend has been to turn previously independent tasks that were initially integrated by the wits of a designer into a single computational task where the tool independently lays out and validates a fully specified functional logic for a given design feature, such as USB functionality or memory caching, on an allocated physical area.

IC chip design is roughly divided into four tasks: specification; logic design; physical design; and validation. Specification determines how the chip is expected to behave logically, and also specifies its physical performance requirements, such as clock speed, power consumption, or instruction fetch time. Architects, high-level physical designers, and logic programmers develop such specifications early on during the chip design. During the logic design stage, logic programmers generate specifications that detail the logic functions that each section of an IC is expected to perform. The specification is expressed in terms of Boolean logic functions and timing requirements. The outcome of the logic design is a logic specification file formulated in a machine-readable programming language such as RTL (Register Transfer Level). During the physical design stage, physical designers (whom we focus on in this study) implement the given RTL specifications by placing circuit gates and related logics on an allocated physical space of the IC. The physical design implements, in silicon, the behavior expressed in the given RTL code, while trying to meet other physical constraints set up for the chip that might include timing, power, heat, dissipation (leakage of

electrons), noise, and manufacturability. Finally, during validation, physical engineers and validation specialists test whether the physical chip performance meets the specification.

The use of DA during the physical design phase has often been regarded as the most critical factor in improving design productivity. It is the longest stage of the chip design process and is also the most error prone<sup>22</sup>. At the same time, the effort necessary to carry out the physical design has increased by a steady factor of two on a biannual basis, thus following Moore’s law (Denning and Lewis 2017). Therefore, close to 100% of the design work during physical design has been digitalized and heavily automated to keep up with the continuous need for productivity improvements.

In Table 4 below we explain the technical terms used throughout the manuscript. Note that some of the names of technologies have been changed to similar-sounding names to protect the confidentiality of our interviewees.

<b>Table 4. Glossary</b>	
<b>Gates</b>	Short for “logic gate” or “electronic gate”—the basic component of chip design. It performs one or several Boolean functions.
<b>Wiring</b>	Electronic connections between gates.
<b>FUB</b>	“Functional Unit Block”—a region of a chip often allocated to a single designer.
<b>Layout</b>	The representation of a physical placement of electronic components that matches the logic specification and meets specified engineering requirements, such as timing or noise-related requirements.
<b>Netlist</b>	A file containing the list of functional components and connections between these components based on the schematic. Note however that these components are not “physically” placed but are instead simulated during layout design.
<b>Rapid Files</b>	An intermediary layout created based on a netlist. The rapid file is a prototype-like-layout that allows the designer to test the given requirements.
<b>Schematics</b>	A graphic diagram representing an abstract and rough design based on the logic specification. Schematics do not necessarily fulfill all the detailed engineering requirements but are rather an abstract representation of how components relate on a chip.

---

<sup>22</sup> In our interactions with the leaders and design managers at ChipCo they often highlighted the special challenges related to physical layout.

## APPENDIX B: INTERVIEW GUIDE

As our interviews were carried out over 4 years spanning 9 rounds, we adapted our interview questions over time. During our first site visit, we focused on attaining a comprehensive overview of the chip design process. We therefore interviewed a large number of designers that included team leads, tool designers, section managers, and project managers. Based on these initial interviews and our emerging theoretical insights, we constantly updated our interview questions as presented in Table 5 below. Primary questions (questions asked at all visits) were repeated in each interview round. “Questions no longer asked in later visits” were only asked during the first two visits to help us gain a broad understanding of the overall design practices at ChipCo. “Questions added in later visits” were added during our last three visits as we focused more on the designers’ relationship with their tools.

<b>Table 5. Interview Guide</b>			
<b>Designers</b>	<b>Tool Designers</b>	<b>Management</b>	<b>Questions asked at all visits</b>
Yes	Yes	Yes	Please begin by giving me a short history of your own career and how you came to work with your present organization.
Yes	Yes	Yes	We are interested in various forms of information technologies that you use in your design project. What are the key digital tools that your company uses to support design projects? Can you tell us what specific digital tools your organization has adopted recently?
Yes	Yes	Yes	We are interested in studying if and how the design practices and information technology use of your organization have changed based on your adoption of the tools you mentioned above.
Yes	Yes	Yes	How has the nature of design tasks in your organization changed over the years during your time here?
Yes	No	No	Can you describe how the design processes and digital tools are embedded in each step and phase of the task in the example you mentioned? What are the reasons for using these tools in each task?
Yes	No	No	In particular, can you describe your current design practices, standardized methods, and the specific ways in which you determine requirements, manage them, and how they interact with design decisions.
<b>Questions no longer asked in later visits</b>			
Yes	No	No	We are interested in how these digital tools relate to conventional tools (non-digital) for design work.
Yes	No	No	What has been the relationship between the use of digital and non-digital tools in the example project you mentioned above?

Yes	No	No	What has been the relationship between the use of digital and non-digital tools in the example project you mentioned above?
No	Yes	Yes	How did you come to adopt these tools? How did you come to adopt the design platforms that you mention above? Where did the requirements emerge for these systems—from technology opportunities or from learning from your clients, markets, or internal experiences?
No	Yes	Yes	What were the main barriers in adopting these tools among different work groups at the different sites involved?
No	Yes	Yes	Were there differences in the ways in which each group or individual had to work?
			<b>Questions added in later visits</b>
Yes	No	No	How is the use of digital tools related to the time-space distribution of your design practices?
Yes	No	No	How has your relationship with the tools changed over the life of the tool, the individual task, and over the life of projects?
No	Yes	Yes	How has the nature of collaboration in your organization changed over the years during your time here?
No	Yes	Yes	Have you or your organization discontinued use of any of these collaborative tools? If so, what were the main reasons for choosing to do so? If not, why?
No	Yes	Yes	Please explain how the use of one of these collaborative tools has changed over the life of the tool and over the life of the project.

## APPENDIX C: DRAMATURGICAL MODEL OF INTERVIEWS

We followed the guidelines proposed by Myers and Newman (2007) to conduct interviews according to the “dramaturgical model” (see Table 6 below). By doing so, we were able to get our interviewees to talk freely about what they actually did. Our interviewees tended to open up and the setting became a safe space where interviewees “ventilated” through talking about shared experiences, frustrations, feelings, difficulties, challenges, and perceived puzzles with regards to the work processes that they were carrying out.

<b>Table 6. Dramaturgical model of interviews</b>	
<b>Guidelines</b>	<b>Implementations in our study</b>
Situating the researcher as an actor	We started our interviews by briefly introducing ourselves and providing an overview of the research. We also asked questions related to interviewees’ background and experience before moving on to the research questions.
Minimize social dissonance	We explained that the researchers had signed a non-disclosure agreement, i.e., that all the conversations would be kept confidential and the subjects would be kept anonymous. We also built trusting relationships with our interviewees through repeated conversations and also by interacting with them socially during coffee breaks and luncheons.
Represent various “voices”	We interviewed tool designers, architects, management teams, and several designers at different time points to capture alternative perspectives and experiences of how the chip design progressed using the two approaches.
Everyone is an interpreter	All interviews were taped, transcribed verbatim, and stored in a central repository for comparison. We also verified and discussed our interpretations with the interviewees in the next rounds of interviews.
Use mirroring in questions and answers.	All interviewers were familiar with the technical terminology used in the chip design, and subsequent questions were constructed using the interviewee’s language. We were given several tutorials during each visit on how the technology had changed and also on different aspects of chip design such as layout, tool designs, architectures, logic design, power, and manufacturing problems.
Flexibility	We adopted a semi-structured interview guide using open-ended questions. Additional clarifying questions, prompted by an individual interview, were asked as the need emerged during each interview.
Confidentiality of disclosures	We signed non-disclosure agreements with the company and all the interview recordings and transcripts were securely stored.

## APPENDIX D: CODING TREE OF INTERVIEW DATA

Table 7. Coding Tree			
Representative Quotation	First order	Second order	Third order
<p><b>Traditional Tools:</b>  <i>“[Designer] can give a logical description of, “Grab this group and make it horizontal. And grab this group and make it horizontal. Grab both of those and make them vertical” (Tool Designer O)</i></p> <p><b>Autonomous Tools:</b>  <i>“So one of main inputs is gonna be the [constraint A] and one of the inputs is gonna be the [constraint B] and the [constraint C]” (Manager Y)</i></p>	Inputs	Computation	Computational agency
<p><b>Traditional Tools:</b>  <i>“It’s still fully specified by the designer, which is why it’s more intensive and still custom, but we allow the designer to manipulate the design easier... in the design space, we’re writing those tools to allow the designer to manipulate it more efficiently” (Tool Designer O)</i></p> <p><b>Autonomous Tools:</b>  <i>“So the algorithm is random because at the beginning it’s random. You don’t know where you’re going to start. So imagine the air. You throw a ball somewhere in air, and it’s random and then it tries to do its best to find the minimum solution around that there.” (Autonomous Tool Designer F)</i></p>	Execution of Input		
<p><b>Traditional Tools:</b>  <i>“you can make changes in [tool A] to define the relative placement of the pieces based off the net list, and so you save your results into a rapid file and then you iterate upon” (Traditional Tool Designer J)</i></p> <p><b>Autonomous Tools:</b>  <i>“In the design space, you change the inputs and then you kick it off to the tool and then the tool changes the design on its own and spits out results” (Tool Designer O )</i></p>	Outputs		
<p><b>Traditional Tools:</b>  <i>“it’s still automated from the standpoint of what you’re doing is telling it where to place different design elements, so you can control where things get placed and then you capture that in a recipe</i></p>	Input-Output Relationships		

<p>so you can repeat it and then we can use a router, an auto router to connect it, or you can even direct to the router as well. You know as much control as you need, but what's different, we used to draw all that by hand." (Manger T)</p> <p><b>Autonomous Tools:</b>          "I don't understand what my [section] does nearly as well from a logic standpoint as I used to for my other [sections]. I used to understand. Even [subsections produced using autonomous tools] ten years ago, I knew what they did. I had looked at the logic specification and understood all the different blocks of the specification and what it did and I could probably make hand edits to it myself. Nowadays, no, I don't. I only know at a high level what it does. I don't know what each of the different modules actually produces. I don't know what each of the [sections] is storing..." (Autonomous Tool Designer C)</p>			
<p><b>Traditional Tools:</b>          "all the tools are generally in-house and have been home-grown over the last 15 to 20 years" (Tool Designer O)</p> <p><b>Autonomous Tools:</b>          "[Autonomous] tools, we tend to buy from third-party companies like [company A] and [company B] and we bring those tools in and we customize them." (Tool Designer O)</p>	Computational resources	Procrastinated binding	
<p><b>Traditional Tools:</b>          "Now we want to capture all that in a tool so if something has to move, you change the recipe and then tools just redraw and regenerate it for you" (Manger S)</p> <p><b>Autonomous Tools:</b>          "Some of the tools would just crash 'cause they'd run out of memory trying to design it, and so they chopped the thing in half right down the middle, but in a terrible way. The FUB wasn't designed to be two pieces, and when they cut it down the middle, they had way too many wires crossing that interface." (Autonomous Tool Designer C)</p>	Impact on design		
<p><b>Traditional Tools:</b>          "Yeah, I mean it's a handful of minutes, maybe 15 minutes....so it is pretty fast." (Traditional Tool Designer J)</p> <p><b>Autonomous Tools:</b></p>	Runtime		



<p><i>“Okay, so this sequence here on our large FUBs takes nine days. So just doing that first step of is sometimes six days, and doing all the analyses, the machines run for two to three days.”</i> (Autonomous Tool Designer B)</p>			
<p><b>Traditional Tools:</b> <i>“[We] read the RTL, hand draw the schematics, hand do the placement.”</i> (Traditional Tool Designer J)</p> <p><b>Autonomous Tools:</b> <i>“because in [Autonomous Design] you like to do a lot of different experiments where you try different knobs or different settings”</i> (Autonomous Tool Designer G)</p>	Design approach	Action	Human agency
<p><b>Traditional Tools:</b> <i>“So, we’ve done some things to say, ‘Take this group of logic and array it out horizontally. And then take these other cells and group them up and array them out horizontally.’”</i> (Traditional Tool Designer H)</p> <p><b>Autonomous Tools:</b> <i>“There’s a lot of configuration you can do to the [tool], so there are a lot of tweaks you can do, different knobs you can turn to tell it ‘Work harder on this. Work harder on that. Focus in this area more.’ So there’s a lot of knobs that you can do internally other than moving black boxes. You can force the placement into different areas to reduce congestion and things like that. So there are a lot of internal knobs.”</i> (Tool Designer P)</p>	Input to the tools		
<p><b>Traditional Tools:</b> <i>“if you change something, it’s going be more or less the same thing, plus your change”</i> (Traditional Tool Designer H).</p> <p><b>Autonomous Tools:</b> <i>“You couldn’t control it. It would fix not quite the way you would actually want it. So if you really wanted things done the way you want it and it was important, then you had to do it by hand.”</i> (Autonomous Tool Designer D)</p>	Expectation of input		
<p><b>Traditional Tools:</b> <i>“Yeah. I mean probably a couple times [per day]... there’s more debug time, so being able to figure out what needs to be changed sometimes took longer”</i> (Traditional Tool Designer J)</p>	Iterative dimension	Temporal organization	

<p><b>Autonomous Tools:</b>  <i>"They iterate every week for years. For more than two years."</i> (Manager T)</p>			
<p><b>Traditional Tools:</b>  <i>"I mean it is much more predictable every time you redo it. You may get a little slight difference in the routing from the auto router"</i> (Traditional Tool Designer J)</p> <p><b>Autonomous Tools:</b>  <i>"You can't predict where the gates are going to be, where the latches are going to be. You can't predict any of that."</i> (Autonomous Tool Designer D)</p>	Projective dimension		
<p><b>Traditional Tools:</b>  <i>"my experiences with it is that you can get a pretty good idea on it. You see what the width of the [gate] is and things involved"</i> (Traditional Tool Designer J)</p> <p><b>Autonomous Tools:</b>  <i>"kick them all off in parallel. When they come back, figure out what the best one is and see if that problem's been solved"</i> (Autonomous Tool Designer D)</p>	Practical-evaluative dimension		