

control or measurable extents, and they have a complexity much greater than the sum of their parts.

Of course, not all social interactions with software transduce code/spaces. Although much spatiality is dependent on software, software merely augments its transduction in other cases. We term such cases coded spaces—spaces where software makes a difference to the transduction of spatiality but the relationship between code and space is not mutually constituted. For example, a presentation to an audience using PowerPoint slides might be considered a coded space. The digital projection of the slides makes a difference to the spatiality of the lecture theater, influencing the performance of the speaker and the ability of the audience to understand the talk. However, if the computer crashes, the speaker can still deliver the rest of the lecture, but perhaps not as efficiently or effectively as when the software worked.

In other words, the distinction between coded space and code/space is not a matter of the amount of code (in terms of the number of lines of code or the density of software systems). Rather a code/space is dependent on the dyadic relationship between code and space. This relationship is so all embracing that if half of the dyad is put out of action, the intended code/space is not produced: the check-in area at the airport does not facilitate travel; the store does not operate as a store. Here, "software quite literally conditions . . . existence" (Thrift and French 2002, 312). In coded space, software matters to the production and functioning of a space, but if the code fails, the space continues to function as intended, although not necessarily as efficiently or cost efficiently, or safely. Here, the role of code is often one of augmentation, facilitation, monitoring, and so on rather than control and regulation.

As we detail in depth in chapter 4, it is important to note that the relationship between software and space is neither deterministic (that is, code determines in absolute, nonnegotiable means the production of space and the sociospatial interactions that occur within them) nor universal (that such determinations occur in all such spaces and at all times in a simple cause-and-effect manner). Rather, how code/space emerges through practice is contingent, relational, and context dependent. Code/space unfolds in multifarious and imperfect ways, embodied through the performance and often unpredictable interactions of the people within the space (between people and between people and code). Code/space is thus inconsistently transduced; it is never manufactured and experienced in the same way.

Discursive Regimes Underpinning Code/Space

The adoption of software and digital technologies, and the systems, networks, and ways of doing they underpin, have been complemented by a broad set of discursive regimes that have sought to justify their development and naturalize their use. For Foucault (1977), a discursive regime is a set of interlocking discourses that sustain and

every software/dig tech is underpinned by its own distinctive discursive regime.

reproduce, through processes of definition and exclusion, intelligibility and legitimacy, a particular set of sociospatial conditions. Such a regime provides the rationale for how sociospatial relations are predominantly produced, legitimating the use of discursive and material practices that shape their production.

As we discuss in chapter 5 and illustrate in subsequent chapters on travel, home, and consumption, the development and employment of different types of software and digital technologies are underpinned by their own particular, distinctive discursive regime. That said, they usually consist of an amalgam of a number of common discourses: safety, security, efficiency, antifraud, empowerment, productivity, reliability, flexibility, economic rationality, and competitive advantage. In other words, they argue that the deployment of software will improve the safety of individuals and society more broadly; make society and travel more secure; make government or business more efficient; make the fight against crime more effective; empower people to be more creative and innovative; and so on. These discourses are often promoted by government in tandem with business, driven by the interests of capitalism and, increasingly, the agenda of neoliberalism focused on the delivery of social services for profit within a target-driven culture.

The constituent elements of a discursive regime work to promote and make commonsense their message, but also to condition and discipline. Their power is persuading people to their logic—to believe and act in relation to this logic. As Foucault (1977, 1978) noted, however, a discursive regime does not operate solely from the top downward, but through diffused microcircuits of power, the outcome of processes of regulation, self-regulation, and localized resistance. As such, people are not simply passive subjects, disciplined and interpellated in linear and unproblematic ways by discursive regimes. Rather, as with the technologies themselves, such discourses are open to rupture: subversion, denial, and transgression by flourishing software hacking communities, anticorporate web sites, online activist networks, legal challenges to security and surveillance, and campaigns concerning privacy and confidentiality, for example. In this sense, power is not captive, purely in the hands of an unseen elite, although the discursive regime operates—in conjunction with the operation of code/space—to try to (re)produce such a hegemonic order. As such, code/spaces and their discursive regimes work to reinforce and deepen their logic and reproduction, at the same time as others seek to undermine, resist, and transform their hegemonic status. Software opens up new spaces as much as it closes existing ones. Accordingly, Amin and Thrift (2002, 128) argue that because "the networks of control that snake their way through cities are necessarily oligoptic, not panoptic: they do not fit together. They will produce various spaces and times, but they cannot fill out the whole space of the city—in part because they cannot reach everywhere, in part because they therefore cannot know all spaces and times, and in part because many new spaces and times remain to be invented."