THE UNIVERSITY OF CHICAGO

TRANSACTION AND MESSAGE: FROM DATABASE TO MARKETPLACE, 1970-2000

A DISSERTATION SUBMITTED TO

THE FACULTY OF THE DIVISION OF THE SOCIAL SCIENCES

IN CANDIDACY FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF SOCIOLOGY

BY

MICHAEL C. CASTELLE

CHICAGO, ILLINOIS

DECEMBER 2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACID	Atomicity, Consistency, Isolation, and Durability
API	Application Programming Interface
ARPA	Advanced Research Projects Agency
AMEX	American Stock Exchange
ATS	Alternative Trading System
BBN	Bolt, Beranek, and Newman
CAL	TSS Cal Time-Sharing System
CATS	Computer Assisted Trading System
CCITT	International Telegraph and Telephone Consultative Committee
CICS	Customer Information Control System
CLOB	Centralized Limit Order Book
CMS	Common Message Switch
CORBA	Common Object Request Broker Architecture
CQ	Consolidated Quote
CT	Consolidated Tape/Trade
CTS	Consolidated Tape/Trade System
DEC	Digital Equipment Corporation
DOT	Designated Order Turnaround
DTC	Depository Trust Corporation
DTCC	Depository Trust & Clearing Corporation
ECN	Electronic Communication Network
EDP	Electronic Data Processing
HTML	HyperText Markup System
IMS	Information Management System
INGRES	INteractive Graphics REtrieval System

IPO	Initial Public Offering
IRIA	*Institut de recherche en informatique et en automatique*
ITS	Intermarket Trading System
MDS	Market Data System
MOM	Message-Oriented Middleware
MQ	Message Queue
MTTF	Mean Time to Failure
MTTR	Mean Time to Recovery
MVS	Multiple Virtual Storage
NASD	National Association of Securities Dealers
NASDAQ	National Association of Securities Dealers Automated Quotations
NMS	National Market System
NSCC	National Securities Clearing Corporation
NYSE	New York Stock Exchange
OARS	Opening Automated Report Service
OLTP	On-Line Transaction Processing
OSF	Open Software Foundation
OS/MFT	Operating System / Multiprogramming with a Fixed number of Tasks
OTC	Over-the-Counter
PAMS	Process Activation and Message Support
PARC	Palo Alto Research Center
POSIT	Portfolio System for Institutional Traders
RAND	Research ANd Development
RPC	Remote Procedure Call
SAGE	Semi-Automatic Ground Environment
SDC	System Development Corporation
SEC	Securities and Exchange Commission

SECURE	System to Eliminate and Correct Recurring Errors
SIAC	Securities Industry Automation Corporation
SNA	Systems Network Architecture
SQL	Structured Query Language
SRO	Self-Regulatory Organization
TIB	The Information Bus
TCP/IP	Transmission Control Protocol / Internet Protocol
TMF	Transaction Management Facility
UDP	User Datagram Protocol
USENIX	Unix Users Group
VAX	Virtual Address Extension
VMS	Virtual Memory System

ACKNOWLEDGEMENTS

In the course of this research I received financial support from the University of Chicago Social Sciences Division and the University of Chicago Press Graduate Student Fund. I also received two British Studies Fellowships from the University of Chicago Nicholson Center. In addition, I received the Arthur L. Norberg Travel Grant from the Charles Babbage Institute at the University of Minnesota. I am also grateful to the University of Chicago Department of Sociology for the Robert E. Park Lectureship, which allowed me to test the coherence of my perspectives on unsuspecting and talented undergraduates.

In the Department of Sociology, Ben Merriman always seemed to believe in me even when I didn't, and was a persistent supporter of my work and ideas; I also thank fellow Chicago graduate students Chad Borkenhagen, Jessica Feldman, Peter McMahan, Forest Gregg, Paola Castaño, Monica Lee, and Maude Pugliese for debates and discussion. Outside of Chicago I would also like to thank Etienne Ollion; and I thank Jared Rosoff for being, at the outset of my explorations, particularly open-minded and willing to entertain my questions.

I would like to thank Jeanne Fitzsimmons and the Nicholson Center for British Studies for helping facilitate my two trips to the United Kingdom, where I developed the material for Chapters 4 and 5 of this work alongside Yuval Millo, Emma Uprichard, and other members of both the University of Leicester School of Business and the University of Warwick's Centre for Interdisciplinary Methodologies.

My advisor, Andrew Abbott, has been an invaluable assistance to the direction of my work, both directly and indirectly; his vast and always-extending breadth of interests will always be an inspiration. I am also grateful to the other members of my committee, Andreas Glaeser and

Adrian Johns, for their comments, insights, and encouragement. In addition, I would like to thank Karin Knorr Cetina for her skilled critiques, and to James Evans for his enthusiasm and support after my arrival to his (and Adrian Johns') "Introduction to Science Studies" course as an auditor then in the employ of the Department of Neurology.

Parts of Chapter 2, "Computing in Organizations: Electronic Data Processing and the Relational Model", previously appeared in the online journal *Computational Culture*. Parts of Chapter 5, "Where do Electronic Markets Come From?", previously appeared in *Economy and Society.* And parts of Chapter 6, "The Platform as Exchange", previously appeared in *Economic Sociology: the European Electronic Newsletter*.

ABSTRACT

The term ‘transaction’ is frequently used in discussions of economic activity, but is rarely the direct subject of examination, despite its apparent centrality. This is true whether its role is ascribed to the material mediation of money, to financial capital more generally, or to the rationalized technical activity at the core of large formal organizations. The modern representation of a transaction, however, began as a practical solution to a very specific technical question: how can a computer system successfully handle simultaneous requests contending to rapidly read and write from the same large data resource?

This dissertation makes the case for the sociotechnical formalization of the transaction, along with that of message-oriented data communications, as fundamental prerequisites and facilitators for significant transformations in finance, including the development of electronic securities exchanges. However, these transformations were as political as they were technological. Through an examination of competition and regulation in the securities exchange industry, I show how markets—both in practice and in theory—are redefined by the prevalence and interrelation of reliable distributed platforms for automatically matching buyers and sellers, executing transactions, and broadcasting quotes and trade reports. The resultant redefinition of the exchange as a sociotechnical ‘platform’ then provides a novel framework for the analysis and regulation of contemporary marketplace platforms.

CHAPTER 1

THEORETICAL FOUNDATIONS FOR SOCIAL STUDIES OF COMPUTING

Introduction

This work began as an attempt to answer a relatively simple, if slightly odd, question: what would a 'sociology of databases' look like? From my viewpoint in the early 21st century, it had long appeared to me that once a small number of individuals, even the most otherwise disorganized, devoted themselves to a directed task—such as producing a short film, or hosting a moderately-sized social event—one or more participants would eventually use Microsoft Excel or other spreadsheet software to manage various aspects of the process, sharing and updating their (often quite intricate) tabular data with other members of the group. Meanwhile, I would also occasionally come across (or become temporarily employed by) more-perduring organizations in the sphere of electronic commerce—medium-sized businesses which had grown from smaller mail-order businesses in the 1990s. Each of these consistently ran their operations using a bespoke 'relational' (i.e., tabular) organization of data with open- or closed-source relational database systems, perhaps in concert with older proprietary warehousing software run on mainframe-style systems. But there was never any question that this digital manifestation of their company's operations would be represented in this structured form, and often little awareness that there could be alternatives to doing so.

But with few exceptions, the link between this highly 'tabular' quality of contemporary thought and that of social organization seemed to be rarely a subject of intellectual reflection. What would it mean to seriously consider social structure, as I then put it, "including spreadsheets"—or including the seemingly-ubiquitous relational database? The process of answering this question took me on a journey through the history of some of the most potentially 'boring'-sounding technical formations: database management systems, on-line transaction processing, message-oriented middleware—but taken together, it was precisely these formations which provided the mostly-reliable core of techniques and technologies for all of today's online digital environments, from chatting and socializing to corporate data and communications operations. Moreover, my explorations led me again and again to the sphere of global finance, specifically in their sites of apotheosis—the financial districts of New York and London. It appeared that while many historians of computing had come to understand the relevance of the seemingly quotidian business use of data processing to their field, they had not yet entirely come to terms with how much important conceptual and material change in computing was often driven by the needs of already-existing complex, fast-paced financial systems.

Subsequently, by engaging with the literature on 'market microstructure', an intriguingly materialist sect of economic thought concerned with the technical organization of trading activity, I was struck by the seeming confusion between the practices of financial markets, where buyers can also act as sellers (as in 'flipping' a stock), and the practices of production markets in which buyers are strictly 'downstream' from sellers, as in a supply chain. Basic concepts about markets, like 'supply and demand' or 'competition' or 'liquidity', seemed to take on specific, and often different, meanings and moral values in the financial context. In this case of this conflation of different types of markets, I was able to find existing work in the subfield of economic sociology which supported my suspicions; but those theories of markets had not yet been applied to stock exchanges specifically. Nor had it been applied to a then-emerging domain of business, that of the 'marketplace platform', then (and now) held to be 'disrupting' various service industries. The goal of this work, then, is to begin with the 20th-century predilection for organizing with tables, and to end with a new understanding of these 21st-century platforms, which often appear to unconsciously mimic and reinvent all of the conceptual challenges of the exchanges and the financial markets they produce.

Outline

In this introductory chapter, I set the intellectual stage for understanding the development of (and controversies around) database systems, the formalization and commercialization of transaction processing and message processing techniques, the deployment of those techniques in financial industries in marketization processes, and finally the distributed "scalability" and preponderance of marketplace platforms. Databases were developed during a techno-cultural moment of increased centralization, as terminal devices began to be connected to expensive mainframes; but the distributed systems of the 21st century were instead composed of aggregations of microcomputers, each far more powerful than 1970s mainframes, and each networked to innumerable other devices via the Internet. This is a story, then, about the *sociotechnical structure of computing systems*: how these machines were initially arranged and programmed with respect to existing social structures—such as those of bureaucratic forms of organization, or those of a stock exchange trading floor—and how those arrangements changed over time to coalesce into certain contemporary forms, such as those of commercial digital 'marketplaces'.

My four main chapters will address major concepts:

- Chapter 2: What is *data*?
- Chapter 3: What is a *transaction*?

- Chapter 4: What is a *message*?

- Chapter 5: What is a *market*?

This work, then, also composes a thesis about the *processual ontology of marketization*—a story of how ideas about the social structures known as markets were formalized, materialized, reified, and redeployed. It can, however, be distinguished from a general thesis on the nature of technology or computing technology in that it is focused on the uses of technology for (and by) *organizations*, especially those characterized by a preponderance of *scale* and/or *scope* in their production of goods or services and corresponding volumes of (commercial, financial, accounting) transactions. At the same time, I intend for the discussion of the techniques and technologies at hand to be part of an allegorical mediation of *methodologies* for both social scientists and computing practitioners who might wonder how these broader social, technological, and political transformations might bear on their own techniques and practices.

By the term 'organization' I bring under one umbrella patterns of practice from business, non-profit, governmental, and military forms. The military has sometimes been taken as *the* prominent causal force in the development of techniques and technology: In the 1930s, for example, Lewis Mumford claimed that "[a]t every stage in its modern development it was war rather than industry and trade that showed in complete outline the main features that characterize the machine… [t]he army is in fact the ideal form toward which a purely mechanical system of industry must tend."[1] Later, it became fashionable to take a similar approach with the history of the computer; but the empirically driven interventions of scholars like James Cortada and

---

[1] (Mumford 1934, 89)

Thomas Haigh—as well as prior work by James Chandler and Joanne Yates—exposed a different perspective, in which the interests of *business*, in terms of total human labor, expenditures, and cultural influence, were taken to be a highly underrated influence on changes in computing practices.

And by the term ‘methodologies’ I want to highlight the fact that the issues at hand here—the representation of data for scientific and business applications, the atemporal quality of abstract database models, the unavoidably temporal challenges of distributed systems, and the properties of markets and platforms—all of these bear on questions of practice in both social science and computing, whether one is a statistician or a web developer; a survey researcher or a network engineer. The choice of technical methods in each of these fields will inevitably involve one or more of these topics, and I hope to highlight the deep historical, philosophical and semiotic assumptions and questions upon which everyday sociotechnical practice depends.

Philosophical Interventions

Crucial to this work are a series of philosophical interventions, drawn from a combination of philosophers, historians of philosophy, sociologists, and anthropologists:

- First, we have the notion of *originary technicity*, or technosociality;
- Second, the multivalence of ‘technology’: technology as tool, technique, social practice, and volition (inspired by the work of Carl Mitcham);
- Third, the notions of, and relevance of *indexicality* (drawn from Peirce, and in opposition to the purely *symbolic*); *context* (found in many fields), and

entextualization/recontextualization (terms from linguistic and cultural anthropology).

- Fourth, the fundamental *processuality* and *temporality* of technical action (from Elias, Abbott, and even phenomenology and category theory).

Together, these theories will compose the groundwork on which I will articulate two broad histories of computing and communication, centered around the period from the early 1960s to the early 2000s, focused on the sociotechnical developments which most fundamentally facilitated these marketization trends. In the rest of this introductory chapter, I will introduce and summarize each of these interventions.

Intervention #1: Originary Technicity

In this section, I will first survey the 'sociology of technology' as it existed in the early 20th century, before the foundation of the Society for the History of Technology (SHOT) in 1958. I will then discuss some contemporary developments in continental philosophy which I think are useful for bringing to the fore the notion of *originary technicity*; the idea that all human societies were and are intrinsically technical, and that the social sciences would be much improved if they take technicity as an essential, unalienable element of the social. Finally, I will discuss the case of contemporary organizations and economic sociology as fields which, through the works of Orlikowski, Callon and MacKenzie, have most productively engaged with the essential sociotechnicity of action.

Technik, Technique, and 'Technology' in Early 20th-Century Sociology

Early sociological theory was fundamentally concerned with the technological and the organizational, without explicitly describing itself as such, and thus—especially if one takes, as I will, a perspective on the history of computing closely aligned to concomitant development in organizations—it is not at all inconceivable to link aspects of the foundations of sociology with a computerized contemporaneity.

Specifically, the seminal writings of Max Weber on bureaucracy—portraying formal organizations as an array of social practices, tools, and techniques which support *rational-legal* strategies—have long been an inspiration for those taking seriously the materiality of organizations along with relationships of hierarchies of egos. Marx, on the other hand, with his dubious theory of transduction from labor to value, can nonetheless be held to be strongly concerned with technological change, especially in its historical contingencies and with respect to conditions of labor. Where Weber focuses on "the files" of organizational practices, Marx is concerned with "the machine", which also strives for efficiency, but primarily in the sense of supplanting labor power (MacKenzie 1984).

It was difficult, however, for early English-language sociology to develop a broader 'technological' approach, missing as it was the multivalent term 'technology' as we use it today. For the English word 'technology' in the early 20th century in fact did *not* denote the tools and techniques of the industrial arts, only the *study* thereof (hence *–ology*). It is a challenging but not insurmountable task to determine how the meanings of 'technology' in English evolved; by contrast, today there is *no* single word that indicates "the study of technology" (it would have to be something unwieldy like 'technologyology'). We today have the opposite problem: a frequent awareness of being immersed in tools and techniques, but no named body of knowledge with which to study our condition.

Confusingly, German has the cognate words *Technik* and *technologie*, and French has *technique* and *technologie*; but the meanings of these words as they developed over the last hundred years do not easily map onto English 'technique' and 'technology'. In the late 19th century, the sense of the English word 'technology' was analogous to that of French or German *technologie*, i.e. "the scientific study of the practical or industrial arts" (R. Kline 1995). (Hence, e.g., the "Massachusetts Institute of Technology", founded in 1861.) At the time, the English word "technique" at the time generally referred to skill in the fine arts (e.g. music); but in German in the middle 19th century, *Technik* became progressively came to refer specifically to the industrial arts, and also had the sense of being related to the rules and methods for achieving goals in those arts.[2] This sense of *Technik* led to its being embraced by the emerging engineering profession in Germany.

As Schatzberg (2006) describes, the closeness of the term to engineering topics inspired broad debate over the relationship between *Technik* and *Kultur* (the latter term denoting the intellectual, aesthetic, or religious aspects of civilization); no such contemporaneous debate occurred in the English language. An exemplary discussion occurred in 1910, in Max Weber's response to a lecture by the economist and sociologist Werner Sombart at the first sociology conference in Germany, in which Sombart designated as *primary technology* (or *production technology*, or *economic technology*) "all the procedures that go into the manufacture of instruments generally, including all the goods that go into their manufacture, their production."[3]

Weber was emphatic that the concept of *Technik* be limited to exclude *Kultur*, and to in general limit the volitional qualities of *Technik* to the "manufacturers who make calculations":

> But I… have major concerns about leaving aside such distinctions as those which, in my opinion, Sombart had to make, and *which consider 'technology' as a certain mode of processing*

---

[2] (Schatzberg 2006).

[3] Sombart 1910 [2005], 95.

> *material goods* (incidentally, I do not want to define this concept further here). *If we do not limit the concept of technology in this way, or if the concept is blurred and everything is drawn into the 'spirit' (Geist) and whatever else of the human being (as happened here), then we will be set adrift and never come to an understanding*. Thus, it is not correct for Prof. Staudinger to put forward the following statement (according to the very broad concept that he employs): the meaning of all technology is that the human being foresees the product he or she wants to produce. This notion is somehow in contrast to what technology is not, though I cannot tell from this conception actually and conclusively what that consists of. It applies to going for a walk, eating and to any other possible performance, *but does it really apply, for example, to weavers, spinners and all the unskilled workers in our factories who manipulate a machine without understanding it? It does not apply to them, but only to manufacturers who make calculations.* [emphasis mine] (Max Weber 2005, 27)

Unfortunately for us, this broad diffusion of the meaning of 'technology' in fact *did* occur, and—to take a Whorfian perspective—our capacity thinking about technology and society has been thus potentially limited. For this, it has been argued, we can blame the sociologist Thorstein Veblen, who introduced concepts from German *Technik* to English, integrating its German senses into those of the English word 'technology', instead of emphasizing the (comparatively rare) term *technics* which might have been used for that purpose.[4] After Veblen, the word 'technology' came to denote (as it does today) various objects and aspects of the industrial arts, and not just the study of technical practices.[5]

The sociologists William F. Ogburn and S. Colum Gilfillan brought the topic of (what we would now call) technology into American sociology in the late 1920s and early 1930s, focusing in part on the concept of *inventions* and emphasizing both their social effects and their

---

[4] For details on Veblen's role in altering the senses of 'technology' in English, see Schatzberg (2006) as well as Oldenziel (1999).

[5] In retrospect, Sombart and Weber's notion that technology be limited to 'a certain mode of processing material goods' is highly suggestive, as this is precisely the sphere of affairs for which computing applications first flourished; the historian of computing James Cortada devotes the first 95-page section of his three-volume The Digital Hand to computing in manufacturing industries, highlighting that some of the earliest profitable uses of digital computer applications in the 1950s and early 1960s belonged to the aircraft, automobile, office machine, electrical equipment, steel, and communications equipment industries (Cortada (1996), Cortada (2003)). Charles Babbage's monograph On the Economy of Machinery and Manufactures (Babbage 1832), well-read by contemporaries, also provides an early analogy between the calculations of (at the time, human) computers and the division of labor in manufacturing, which is echoed by Chapter 14 of Marx's first volume of Capital.

temporally interlinked qualities (i.e. inventions' relations to each other).[6] Gilfillan criticized the "popular belief in individual, single, great inventors" (Gilfillan 1935, 75), a theme which is still recognizable today in the history of technology, which dismisses hagiographic narratives of solitary geniuses—and indeed, Ogburn became the first president of the Society for the History of Technology in the late 1950s, shortly before his death.[7] This group and its journal, Technology and Culture, would become the base of any interest in sociocultural aspects of technology. Where the explicit relevance of technology remained at all in sociology in the late 1950s and 1960s, it was in the Human Ecology subfield, as in Duncan's 'P.O.E.T'. typology (Hauser and Duncan 1959).

As the historian of innovation Benoît Godin describes, Ogburn—who taught at Columbia until 1927 before moving to the University of Chicago until 1951—was also interested in explanations of social change which privileged technology over (then-in-vogue) biological and evolutionary metaphors, as well as theories of what we would now call *innovation* and the corresponding *cultural lag* in response to material changes.[8] For Ogburn it was this (often unequally distributed) *diffusion* of innovations, and not their invention, that led to social problems and/or social change. Today, some of these concepts have been revived to the point of saturation in business schools, but typically without reference to Ogburn's influence.

(The Lack of) Ontological Technicity in Contemporary Social Theory

Since the departure of these early sociologists of technology, however, the list of influential social theorists of the latter 20th-century among which the essential technicity of

---

[6] Ogburn and Gilfillan (1933).

[7] Westrum (1991).

[8] Godin (2010).

human sociality and agency is literally or effectively absent is long: Parsons, Giddens, Bourdieu, Luhmann, White. But can *any* ambitious social theory which confidently wanders beyond face-to-face communication and social institutions to the world of large, rational-economic organizations without discussing the role (and substrate) of technological mediation be taken as empirically plausible?[9] While it is of course true that many situations and social phenomena may be profitably studied without a constant awareness of technics, it is the position of this paper that (much like the theories relied upon in economic sociology) any social theory, especially including those concerned with firms and other large organizations, cannot seriously elide the presence, influence, and significance of mediated action (symbolically-oriented or otherwise).[10] Too many important social transformations (especially in the technological scaling of organizational action) have already occurred whose retrospective analysis will depend on a computational-technological awareness not yet made common.

It is necessary, then, to discover those social theorists not yet categorized as such, who have drawn attention to what can be (and has been) called the *originary technicity* (or, if one prefers, *ontological technicity*) of social life: put simply, the fact that there has never been a human society not enmeshed with tools and/or techniques, acquired and used socially, with

---

[9] It should be stated that other theorists, however, which largely restrict themselves to the sociolinguistic and cultural phenomena of face-to-face communication—such as those of Goffman and Garfinkel—will continue to find themselves uniquely reconfigurable to computer-mediated communication, for processes of entextualization and recontextualization are present in both situations (though the entextualization to the symbolic is often stricter in, e.g. social media websites than with other kinds of speech acts). See, for example, the suggestive comments on 'platforms' by Goffman (1983, 7).

[10] I occasionally use the less common English term *technics* as a way to move closer to the French *technique* or German *Technik*; see above on the etymology of *technology* in English. On the absence of technology from mainstream organizational studies, see Orlikowski and Scott (2008).

direct intent (if unpredictable outcomes).[11] This is a position held by the *anthropology of technology*, as well as in much of *science and technology studies* (albeit somewhat unconsciously in the latter).[12] What needs to be explained is why, with some exceptions (e.g. the 1990s-era work of Donald MacKenzie), science and technology studies has provided limited insight into the potential relevance of specific computing technologies for social (and especially organizational) life, and why some of those exceptions (e.g. Bowker and Star (1999)) in fact draw on a rather different tradition—that of *social informatics*, a field which is in turn informed by human-computer interaction and phenomenology.[13] As such I will focus on theories that emphasize originary technicity more broadly over the more common perspectives which either conceive of technology a) only as tools (using terms like 'artifacts' or 'materiality') or b) only as being "socially constructed".

Originary Technicity in French Philosophy

In the space of contemporary continental philosophy, the figure who in recent years has represented the core focus on the fundamental technicity of human society is Bernard Stiegler; while his writings can be somewhat inaccessible, his set of literary inspirations are intriguing,

---

[11] My use of 'ontological' is analogous to that of continental philosophy and emphatically not that of the term 'ontology' in computer science and other sciences; the former concerns beliefs about the nature of being; the latter is a quixotic attempt to formalize the former—to standardize the meaning of texts structured by such beliefs. (For an overview of the latter 'ontology' see Gruber (1993).

[12] With respect to science and technology studies, it must be said that the outsized influence of the topic of *science* (combined with a limited disciplinary overlap with historians of technology) has constrained its insights regarding originary technicity. For the sciences in their contemporary forms involve a rather distinctive array of practices, many involving pipelines of material inscription of a kind not remotely universal to all societies; and in contrast to (e.g.) popular computing practice, the stylistics of scientific practice involve long regimentation and close training. The segregation of science studies from sociology only furthered the limited engagement with less-scientific technological practices (such as the more-quotidian world of business data processing of Yates (2006)) in the former, and the lack of recognition of the universality of technics in the latter.

[13] Examples of MacKenzie's sociology of computing include MacKenzie (1993), MacKenzie (2001), MacKenzie (2002).

and his core assumptions reasonable. Drawing from a very select and multidisciplinary set of intellectual predecessors (namely, Plato; the former engineer and philosopher Gilbert Simondon; the physical anthropologist André Leroi-Gourhan[14]; the historian of technical systems Bertrand Gille[15]; and his doctoral advisor, the "deconstructionist" Derrida), his 3-volume series Technics and Time begins with the grand, but quasi-reasonable assertion that "at its very origin and up until now, philosophy has repressed technics as an object of thought."[16] As such, it is worth paying some attention to Stiegler, as his project in philosophy is analogous to what must be ours for sociology: an abandonment of the ontological myopia towards technicity.

Moreover, among contemporary philosophers—and in strong contrast to the professors of Anglo-American analytic philosophy who teach aspects of the formal, logical apparatus of computing without ever seeming to consider the operation of such machines by humans—Stiegler is significant in that (in his varied life history) he was once a programmer-analyst for IRIA (now INRIA[17]), and in his writings—unlike past critics of technology like Jacques Ellul or Jean-François Lyotard—his pronouncements on technics are not immediately disconsonant with the more mundane aspects of everyday technical practice. This is something he shares with one of his primary inspirations, Gilbert Simondon, a former high school teacher of physics and philosophy and student of Canguilhem and Merleau-Ponty, under whom he wrote a thesis

---

[14] Leroi-Gourhan's Gesture and Speech (Leroi-Gourhan 1993), originally written in 1964, followed his 1945 book on technology Milieu et technique. In that earlier volume he argued for the universality of *technique*, which led him to considering technics as a significant biological aspect of human evolution (Audoze 2002).

[15] Gille (1986).

[16] Stiegler (1998). The exceptions, as Stiegler later addresses, would be Heidegger and Derrida; Stiegler's argument about the "repression" of technics is analogous to Derrida's argument for a repression of writing (Derrida 1976). For an excellent argument on the relevance of Derrida to science studies, see Lenoir (1998).

[17] *Institut national de recherche en informatique et en automatique.*

entitled On the Mode of Existence of Technical Objects.[18] In that essay, Simondon eloquently describes the need for social scientists who understand technologies and techniques at a deep level—calling for "an organization engineer who is, as it were, a sociologist or psychologist of machines".[19] Stiegler borrows from Derrida the concept of *grammatization*—the process of making the continuous discrete (Stiegler 2012), or more specifically, "the process of description, formalization, and separation of human behaviors in such a way so they can be reproduced"[20]. This concept has elements of *standardization* (Bowker and Star 1999, 135–61) as well as what linguistic anthropologists would call *entextualization*, the transforming of a situation into a text (in this case a strictly *symbolic* text).[21]

Most significantly for sociologists, Stiegler is one of the few contemporary philosophers who directly engages with Max Weber's notion of rationalization and rational accounting.[22] There are close connections between Weberian rationalization and Stiegler's concept of *grammatization*[23]—and so one can think of Stiegler as making Simondon's ideas accessible to sociologists as much as Weberian ideas to philosophers. However, he is concerned that both

---

[18] Du mode d'existence des objets techniques (Simondon 1980). Simondon speaks of a "mode of existence" of technical objects rather than simply "technical objects" because of his distinct emphasis on processual emergence (Hayward and Geoghegan 2012) which we will discuss in more detail below.

[19] "Cette prise de conscience paraîtrait plutôt pouvoir être le fait de l'ingénieur d'organisation qui serait comme le sociologue et le psychologue des machines, vivant au milieu de cette société d'êtres techniques dont il est la conscience responsable et inventive." (Simondon 1958).

[20]Stiegler and Ars Industrialis (2008), translation by Galloway (2010).

[21] "In simple terms, though it is far from simple, [entextualization] is the process of rendering discourse extractable, of making a stretch of linguistic production into a unit—a text—that can be lifted out of its interactional setting. A text, then, from this vantage point, is discourse rendered decontextualizable. Entextualization may well incorporate aspects of context, such that the resultant text carries elements of its history of use within it" (Bauman and Briggs 1990, 73).

[22]Extended passages of his *Mécréance et Discrédit* series (Stiegler 2011 [2004]) carefully explain Weber's Protestant ethic thesis to a somewhat less empirically-minded audience.

[23] Stiegler and Ars Industrialis (2008), translation by Galloway (2010).

Weber and Marx did not possess a coherent theory of *consumption* (or of desire), and so his theories of capitalism diverge in a psychoanalytic direction less amenable to integration with a sociology still averse to such perspectives (but potentially in alignment with, e.g. a Baudrillardian consumption studies); but like a sociologist, he takes rational accounting and its concomitant technologies and techniques as a *prerequisite* for Western society. Here, technics is universal; but the specific techniques for rationalizing economic practices to some symbolic substrate remains a valuable topic of interest.

Intervention #2: Mitcham's Four Senses of 'Technology'

Carl Mitcham's book *Thinking Through Technology* is a meta-analysis of the philosophy of technology (Mitcham 1994), which, in surveying both the "humanities"- and "engineering"-centric philosophies of technology, finds four broad meanings inherent to the use of the term 'technology' in English: technology as *object* (tool), technology as *knowledge* (technique), technology as *activity* (social practice), and technology as *volition* (intention or agentic interest). Once one appreciates that these four definitions cannot be empirically disentangled, the biases of existing theoretical approaches to 'technology' (and their limited utility for a sociology of computing) become apparent.

For example, a theoretical leaning towards *materiality* will inevitably privilege the first meaning, of technology-as-tool; a leaning towards *knowledge*—especially embodied knowledge, as in Mauss (1934)—will privilege the second meaning, of technology-as-technique; a leaning towards *social construction* will privilege the third meaning, of technology-as-social-practice; and a leaning towards *agency* will privilege the fourth meaning, of technology-as-volition. Mitcham cogently argues that none of our occasions to use the word 'technology' can really eliminate the relevance of each aspect, and yet we can group existing sociological studies of technology as typically being skewed along just one or two of these dimensions. What we are *seeking*, however, are those thinkers whose conception of technology *transgresses* these categories, because like many sociotechnical phenomena, 'computing' is a notion which is taken alternately as an artifactual tool (e.g. "ICT" as a catch-all noun for computing artifacts subsuming everything from bureaucratic mainframes to personal computers to networked, distributed systems), as a technique (e.g. "algorithm" as a kind of opaque, mediating filter), as a social practice (e.g. "hacker culture"), and (rarely) as volition. The ineffective poverty of studying computing in these conceptually isolated ways has long been apparent, and to move forward we must focus on those theories of technology which do not necessarily privilege one perspective, and instead take the social and the technological as undeniably one.

Technology as Object, Knowledge, Activity, and Volition in Science and Technology Studies

It should be noted that the existence of Mitcham's four senses of 'technology' have at times been noted in the science and technology studies literature, though usually not all simultaneously. For example, the general introduction to Bijker, Hughes, and Pinch (1987) describes three layers: that of *physical objects* or *artifacts*, of *activities* or *processes*, and of what people *know* (in the sense of "know-how"). These categories correspond to three of Mitcham's senses, with only *volition* absent. But what an absence that became, as the lacunae was filled by the controversial notion of *socio-technical agencements* (Callon 2005)—with the term "agencements" effectively combining the senses of the material and the volitional.[24] But if the

[24]Related to the need to consider volition in technology—with the argument being that technologies are always created or used for some purposive action—is Swedberg's emphasis on *interests*, which he argues is necessary in order to bridge economic sociology and political economy (Swedberg 2007).

notion of originary technicity is taken seriously, the idea that "agency" be ascribed to tools of varying kinds can be seen as not problematic but merely limited: for when all human societies are taken as intrinsically technological, all action in general is intrinsically technological. It may be the case that science and technology studies' emphasis on individual actors (and their prostheses) as opposed to (clearly socio-technically complex) organizations may have limited the uptake of a more holistic view of technology and society. But if one asks the question "how does a firm take action?", by contrast, it would become clear that all four of Mitcham's senses must be taken into consideration and woven together: for organizations indeed do have social structure—but one cannot seriously model it without including, for example, the hundreds and thousands of spreadsheets—and databases, and Enterprise Resource Planning (ERP) systems—by which that structure is intentionally and prosthetically maintained daily.[25]

Technology in Economic Sociology

With respect to the study of organizations, Wanda Orlikowski has been the strongest voice demanding a "deeper and more dialectical understanding of the interaction between technology and organizations" (Orlikowski 1992), and her critiques of the technology and organizations literature are fruitful when read with respect to Mitcham's fourfold typology of technology. She dismisses Blau et al. (1976) for an overemphasis on technology as mere tools and an underemphasis on technologies which mediate social behavior; she critiques Davis and Taylor (1986) for too-strong claims for individual agency in its ability to create rewarding sociotechnical workplaces; and conversely, notes that Barley (1986) tends to assume that tools and techniques are fixed and that actors have no facility for deliberately changing them. In each of Orlikowski's criticisms we can locate a dimension of Mitcham's which she finds

[25]On spreadsheets, see Levy (1991) and Campbell-Kelly (2003). On ERP, see Pollock and Williams (2008).

underemphasized; and in many of these cases, the core assumption that technology and society are ontologically separate can be seen as the cause of her complaints.

The field of economic sociology, in contrast to organizational studies, has been willing to embrace the inextricability of technology and society, in part through the work of Michel Callon, in his “Laws of the Markets” salvo (Callon 1998); Karin Knorr Cetina on the screen worlds of currency traders (Cetina 2003a); Alex Preda, in his discussions of the stock ticker in the history of financial markets (Preda 2006); and Donald MacKenzie, e.g., in his depiction of the performativity of options pricing theory (MacKenzie 2006). Economic sociology’s productive notion of *performativity* is a theory that simultaneously spotlights the artifactuality, technique, social practice, and intentionality of those technologies and techniques which draw from more hermetic formal theories (such as those of neoclassical economics). MacKenzie’s demonstration of the rise and fall of the canonical Black-Scholes model in options pricing is an excellent example of a materially-embodied technique whose success was nevertheless dependent on its surrounding socio-economic context. One can also productively move from the performativity of mathematical models to a ‘lower’ level of computational mediation—that of the transaction-processing systems which now run as the active substrate on all electronic financial exchanges, as well as brokerage firms’ internal matching engines, as we shall see in Chapter 5. These we can see as in continuity with a performativity—and indeed a problematic one, in an age of controversial high-frequency trading (MacKenzie 2014)—of the auction markets first modeled by Léon Walras in the 19th century.

These potential extensions and parallels to the world of software in general are obvious, but have been only begun to be explored within sociology. For example, the case for performativity of analytical models is in fact far stronger (if still potentially quite controversial) in the world of programming languages, the grammatical syntax for which can easily be traced to mathematical disciplines of formal logic and other, more far-flung, theories (such as the "object-oriented" metaphors derived from architecture (Alexander 1999)). And just as with the rational, game-theoretic actors of neoclassical economics, the supposed foundation of computer science lies in an abstract, individualistic, and thus sociologically problematic model—that of the black-box-like, interactionally-isolated, Turing machine, which I will discuss below.

Intervention #3: Indexicality, Entextualization, and Recontextualization

I have argued that social theories of the technological are often problematic, leaning too far towards conceiving of technology as tool, or as technique, or as social practice, or as volition. I now argue that *computing*, taken as a 'genre' of technology, should also be taken to encompass each of Mitcham's four concepts in any investigation. So as with technology in general, analytical perspectives in which computing is seen as "exterior" (i.e. as black-boxed software tools) can obscure aspects of computing technique (e.g. programming know-how), as well as that of social computing-in-practice, and especially the crucial importance of volition—for very little software is labored over, by individuals or groups, without intentional ends.

But is it possible to actually draw boundaries around what constitutes 'computing' and what does not? As we shall see, this is far more difficult than some have imagined. Entire social-scientific fields claiming to this have been built upon conflations of arguably ill-defined terms: the term "ICT" (*information and communication technologies*) in British informatics research is only one offender, conflating the problematic term 'information' with the (arguably significantly different) phenomena of 'communications.'[26] I will argue later that this conflation of what one might characterize as rule-following symbolic transformation (automated "information" processing of varying sorts) with network standards, infrastructures and/or processual, contextual interaction ("communication") would be better seen as a kind of duality; and the first step towards seeing this is to understand 'information' semiotically.

To do so, I will rely on aspects of Charles S. Peirce's theory of signs, whose uptake remains slow in sociology but shows the occasional signs of resurgence (e.g., Swedberg (2011). The currently most astute proponent of Peircean semiotics is the Yale anthropologist Paul Kockelman; his contemporary works range from a position statement on "the semiotic stance" (Kockelman 2005) to an ambitious book blending anthropological and sociological theory to advance the notion of agency in semiotics (Kockelman 2013). Rather than focusing on Peirce's writings by the letter, I will, following Kockelman (2005), assume an understanding of:

- Peirce's tripartite sign-system, composed of a *sign* ("whatever stands for something else"), and *object* ("whatever a sign stands for"), and an *interpretant* ("whatever a sign creates insofar as it stands for an object"—one can think of this as the process, mental or otherwise, by which an interpretation occurs), all in relation with each other (Kockelman 2006).[27]

---

[26]The term 'information' today simultaneously connotes a) a process of *informing* or communicating knowledge (the older sense, dating to the 14th century; with b) Shannon's formalization of information as statistical entropy (Shannon 1948). R. R. Kline (2006) describes how many intellectual groups in the 1970s identified 'information' as the foundation of a technological revolution despite having rather different definitions.

[27]As Kockelman points out, the object need not be 'real': "An object, then, is whatever a signer and interpreter can correspondingly stand in relation to—it need not be continuously present to the senses, taking up volume in space, detachable from context, or 'objective' in any other sense of the word." (Kockelman 2006).

- A component of the sign-system “triangle” can in turn be a component of another sign-system, continually, processually, and/or recursively. Everyday subjective experience involves uncountable chains of such sign-systems.

- The sign itself can be one of potential likeness (a *qualisign*), of actual physical connection (a *sinsign*), or necessarily of arbitrary convention (a *legisign*).[28]

- However, the relationship between the sign and object, given a particular interpretant (thus composing a full sign-system), can be either *iconic* (a relation of likeness, similarity), *indexical* (a relation of causal connection), or *symbolic* (an *arbitrary* relation, analogous to Saussurean semiotics (de Saussure 1916 [1959]). As such, the interpretant may potentially interpret a sign *as an* iconic sign, *as an* indexical sign, or *as a* symbolic sign, regardless of its “actual” type of connection to the object.[29]

- Finally, interpretants can be divided into (iconic-like) *affective interpretants* (e.g. a change in bodily state); (indexical-like) *energetic interpretants* (involving physical or mental effort, e.g. a reflex action or association); and *representational interpretant* (e.g. an assertion, speech-act, or ‘thought’).[30]

---

[28] Fernando Zalamea, one of the few individuals well-versed in both Peircean semiotics and the historical development of contemporary mathematics, summarizes these ternary categories well: “Peirce’s vague categories can be “tinctured” with key-words: (1) Firstness: immediacy, first impression, freshness, sensation, unary predicate, monad, chance, possibility; (2) Secondness: action-reaction, effect, resistance, binary relation, dyad, fact, actuality; (3) Thirdness: mediation, order, law, continuity, knowledge, ternary relation, triad, generality, necessity” (Zalamea 2016).

[29]This is important as it means that a sinsign which “should” be treated indexically—e.g. a weathervane pointed east which “should” signify a eastward wind—may be interpreted iconically by the less directionally inclined (e.g., as merely another weathervane). The word ‘tree’, a legisign which “should” be interpreted as a symbol conventionally referring to trees, may be interpreted differently by an interpretant without knowledge of English (e.g. iconically, as a inscription of non-sense).

[30] (Kockelman 2013). It seems possibly relevant that this latter type of interpretant is closely related to what philosophers call ‘intentionality’.

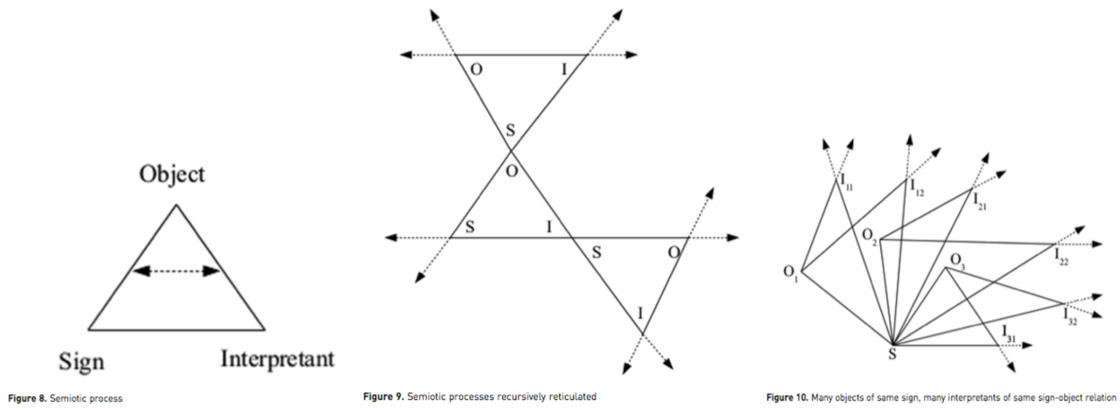

Figure 1. Peirce's sign-systems. At left, a sign-system with object, sign, and interpretant. At center, an example of processuality of sign-systems. At right, examples of varying, parallel sign-systems incident on a single sign standing for varying objects for varying interpretants. (Kockelman 2015, 175–83; © 2015 University of Chicago Press. Reprinted by permission.)

I will argue, minimally, that if we work backwards from contemporary intuitions towards what constitutes a 'computing' practice, we will be forced—except in the special case of *analog* computing—to skew towards (processual chains of) *symbolic* sign-systems: forms of inscription which are interpreted (by human and/or machine processes of interpretation) as having an arbitrary relationship to their object. (This should be more than apparent in those digital cases where the signs held to underlie digital computing practice are referred to a "meaningless" zeros and ones.)[31] There will be, however, be many subtle exceptions to this claim, and teasing these out will allow us to mitigate between these intuitions about the essence of computing.[32]

---

[31] By contrast to this symbolic aspect of digital computing, analog computing is characterized by its indexicality—the inputs and outputs are physically linked, and should be interepreted as such. (As defined by Silverstein, indexes are "those signs where the occurrence of a sign vehicle token bears a connection of understood spatio-temporal contiguity to the occurrence of the entity signaled" (Silverstein 1976, 27)).

[32] By contrast to digital computing, analog computing occurs in an indexical—as such, in direct physical relation to—the input, c.f. the computer scientist Edmund Berkeley:

Just as we cannot extricate tool from technique, technique from social practice, and social practice from volition, we also need to attend to the essential mixture of iconic, indexical, and symbolic signs (and interpretations) which are in play in any computational situation (and indeed any communicative situation). In what follows, I will not necessarily be using the Peircean framework at its most fine-grained levels (e.g., meticulously constructing and empirically arguing for the existence sign/object/interpretant triads in particular situations). Instead I will merely ask the reader to appreciate the close connection between what we would call *context* or *contextualization,* and processual chains of sign-systems involving indexical signs. This latter sort of semiosis, it will be argued, is inevitable in any sociotechnical situation, but this fact is frequently "repressed" among certain actors in the history of computer science; it is instead sometimes presumed that the *entextualization* of contextual phenomena to symbolic inscriptions (as, e.g., required by any program accepting digital input) is unproblematic. Below, Kockelman explains how such an appreciation for contextualization processes brings Peircean semiotics in close alignment with both a *hermeneutic* philosophy of interpretation:

> "[T]o say that one semiotic process *contextualizes* another semiotic process is to say that the meaning of the latter is dependent upon the meaning of the former... Typically, the same semiotic process is contextualized by many other semiotic processes and, in turn, contextualizes many other semiotic processes (any of which can relate to it at various degrees of spatial, temporal, or categorical remove). Such co-contextualizing relations among semiotic processes, and the meaningful coherence they both enable and constrain, is what should be meant by holism in the hermeneutic sense" (Kockelman 2013, 98–99).

Kockelman's reference to *hermeneutics* here denotes an awareness that the meaning of any given activity is a function of various possible subjective interpretations. This will become relevant in Chapter 2's discussions of tabular data, which has a wider variety of possible

---

"Machines that handle information as measurements of physical quantities are called *analogue* machines, because the measurement is *analogous* to, or like, the information" (Berkeley 1949, 65).

interpretations than hierarchically- or network-oriented data; and for when I make claims in Chapter 4 for the *sociality* of message-broadcast systems, whose data-communiqués have the propensity for multiple simultaneous meanings on the part of multiple simultaneous addressees.

Intervention #4: Processuality and Temporality

Throughout this text I will always consider the technical, the historical, the social, and the computational as being intrinsically *processual*. As Abbott describes in his volume Processual Sociology, the term "processual" indicates "an approach that presumes that everything in the social world is continuously in the process of making, remaking, and unmaking itself (and other things), instant by instant" (Abbott 2016, ix). We shall see that to a certain extent this position of "ontological processuality" problematizes otherwise dominant perspectives in each of these spheres.

An oppositional thread that binds subsequent chapters together is the tension between aprocessual and processual perspectives, and the perpetual rediscovery of either perspective in fields/periods in which the other comes to be dominant. The sociologist Richard Ball noted one of these significant intellectual transitions in the rise of General Systems Theory (GST) in the late 1960s and 1970s, explaining that with the influence of Peirce and Whitehead alongside the cyberneticist Norbert Wiener and Von Bertalanffy, "it has been generally conceded that the older classificatory logic must give way to some form of *processual* logic."[33] Systems theory's

[33] (Ball 1978). By "classificatory logic" Ball alludes to a taxonomic impulse, which I would argue is often found in aprocessual thought styles (e.g. the hierarchical organization of Chomsky's syntactic structures).

emphasis on holism and relationships also has its parallels in the symbolic interactionism of Mead, who along with Bergson and Bachelard will motivate our discussion in Chapter 4.[34]

Processual thinking in philosophy is most often held to have the Greek philosopher Heraclitus as its starting point, with his statement “you cannot step into the same river twice”.[35] In our case we will be less directly concerned with process-oriented philosophies like Whitehead’s, and more with those philosophies which (overtly or covertly) reject static representations of information and organization. Specifically, in Chapter 4 we will look to the late-19th and 20th century philosopher Bergson, who contrasted what he called *duration* (as a subjective experience) with the kind of time theorized by physicists of the era, the latter being a “mathematized spatial concept”[36] (as suggested by, e.g., the term ‘timeline’).

Technical Processuality: Simondon and Individuation

Stiegler adopted from Simondon the concept of *individuation*, a term which denotes a process leading to what is called an *individual*. Simondon’s notion of individuation is similar to Norbert Elias’ concept of the *civilizing process* (Elias 1939), a progressive interiorization of social mores (although these theorists do not overtly cross paths, with Elias not being translated into French until 1973); but Simondon generalizes the concept to humans, machines, and their fundamental interrelation, so that one may refer to the individuation process of man as much as one may talk about the individuation process of a machine (which he calls *technical individuation*). One imagines here the process by which the older practice of ‘computing’

---

[34] The ‘symbolic’ of “symbolic interactionism” should not be thought of in the same sense of ‘symbol’ in Peirce (the latter being analogous to Saussure’s signs, and thus having an arbitrary relationship to its referent).

[35] (Nayak 2016).

[36] (Rescher 1996, 17).

without ‘computers’—of repetitive calculation techniques enacted by humans with more conventional inscription devices—becomes “the computer”, a commodity artifact which can perform those same calculations with some degree of reproducibility. A more contemporary example of extreme technical individuation might be the process by which that latter machine becomes a stable, reliable and transposable—and, today, virtualized—unit of so-called “compute time” in “the cloud”.[37]

Moreover, Simondon takes technical individuation as “the essential condition for technical progress” (Simondon (1980 [1958], 49). This move, in a way, cleverly brings the processual and sociological logic of Elias to bear on the interests of Ogburn and Gilfillan. Just as the individuation of humans occurs in a dialectic with the context of their social institutions, the individuation of the technical object is impossible without a comparative stabilization of its environment (and potential for reproduction thereof). One can think of the hermetic (and thus ‘individuated’) space of the late-20th-century “server room”—under lock and key, temperature-controlled, vacuumed free from dust, attended to and backed up by systems administrators—as a precondition for the consistency of any organization’s dependence on computers.[38]

Mathematical Processuality

Concerned as we are with computational techniques, it would also be valuable to closely examine the history of mathematical techniques (and debates thereof) to see where an awareness of ‘process’ exists or does not exist—though to do so is indeed its own full-scale project. In our

---

[37] Such ‘cloud servers’ are today colloquially conceived of almost entirely in terms of ‘compute time’ and its corresponding monetary cost; companies like Amazon have even set up secondary marketplaces for customers to offload unused time (the ‘Reserved Instance Marketplace’).

[38] Increases in scope of data collection of the 21st century have led to the exteriorization of the “server room” into the even more physically restricted “data center”, typically located in industrial areas with excellent network connectivity, and representing an abstract and virtualized commodification of the affordances of the server room.

case it will be most instructive, at a minimum, to understand the emergence (in the early 20th century) and subsequent practical and pedagogical popularity of a *set-theoretic perspective*, as it is this view which will turn out to inform the relational database model of Chapter 2. It will also be in general important to be aware of the broader history of *mathematical logic*, which also proved thoroughly influential (though not always formally fundamental) to many aspects of computational practice. Briefly, however, I will claim that the emergence of set theory can be associated with a tendency towards *Platonism* (i.e., the belief that mathematics is concerned with the study of some externally existing objects); that processual perspectives are strongly in conflict with a specific *Set-theoretic Platonism* which sees the axioms of set theory as fundamental to all of mathematics; and that this conflict will cause controversies in the conceptual and material histories described in succeeding chapters.

Mac Lane (1986) carefully distinguishes between *methodological Platonism* (a recognition that mathematical practice involves the contemplation of abstractions such as infinite sets of objects); *ontological Platonism* (which ascribes a greater 'reality' to these objects; in the face of set theory, Mac Lane finds this position somewhat more absurd); *epistemological Platonism* (involving "some sort of direct acquaintance with Mathematical objects"; also dismissed by Mac Lane, but perhaps more important in a materialist computing); and he suggests that the attempt to build all of mathematics out of set theory leads to its own "*Set-theoretic Platonism*". In the end of this passage, he concludes that "all the variants of Platonism shatter on the actual practice of mathematics." (Mac Lane 1986, 449).

Philosophies of mathematics which are held to be in contrast to Platonism (Set-theoretic or otherwise) include *intuitionism*, *constructivism*, and *finitism*, each of which (implicitly or explicitly) have aspects of a processual perspective. As the historian of mathematics van

Heijenoort describes, the intuitionist mathematician L.E.J. Brouwer (in his work demonstrating that every function defined on the closed interval [0,1] is uniformly continuous) substitutes (1) the notion of an arbitrary sequence as being composed of an *infinity of choices* for each term with (2) the perspective that such a sequence is made up of "an *infinitely proceeding sequence of choices* that is such that at the *nth* choice one could restrict one's freedom as to future choices by laying down some (not necessarily deterministic) law. This is presented as a process in time…" (Heijenoort 1967, 446) [emphasis added][39]. This decision to substitute an unfolding process for an unrealizable infinity of combinations is characteristic of the intuitionistic approach, and may represent a precursor to my attempt in Chapter 4 to consider processually unfolding 'data' as a distinct epistemic object.

Turing's Cyborg Machines

In the late 19th and early 20th century electrical tabulator machines, deriving from the device Herman Hollerith built for the 1890 census, became more frequently used for counting data encoded on sorted stacks of standardized punched cards; this lineage of technology is a direct predecessor for early commercial computers.[40] From about 1912 Hollerith marketed replaceable, wired 'plugboards' which connected card readers and tabulators; and in the 1930s, efforts by the mathematician Leslie J. Comrie at the Nautical Almanac Office in Greenwich, as well as Wallace Eckert at Columbia University, had shown the promise of 'programming' electronic tabulators with plugboards for the purposes of scientific calculation.[41] But

---

[39] The philosopher Lieven Decock has also made a strong argument for a correspondence between Brouwer's intuititionism and process philosophy (Decock 2005).

[40] (Haigh 2011), (Austrian 1982). For the history of punched-card tabulators in the late 19th and early 20th century, see Heide (2009a).

[41](Priestley 2011). See also Agar (2006) on the role and influence of tabulators in government and scientific computing.

concurrently, and to a large degree independently, from these practical developments, a revolution was taking place in the field known as mathematical logic.

Where we have been speaking of 'formal' symbolic systems, it is possible to do so because of the innovations of the University of Jena mathematician/logician/philosopher Gottlob Frege (Frege 1879), who in his attempt to develop a system of logic sufficient for mathematical reasoning and proof, developed universal quantification (∀), existential quantification (∃), truth-functional conditional (→ or ⊃), logical negation (~), although some of these were developed independently by Peirce and others.[42] This formalization of mathematical inference provoked the German mathematician David Hilbert's programme for formalizing all of mathematics (and later led to immense positivist projects like Russell and Whitehead's *Principia Mathematica*). In 1928, Hilbert posed the *Entscheidungsproblem* (or "decision problem" (Hilbert and Ackermann (1928, 73–74)), which was to find a mechanical procedure which would discover, for a given formal system, whether a mathematical expression in the notation of the system is valid.[43]

In 1935, a 22-year-old Cambridge student, Alan Turing, was introduced to Hilbert's *Entscheidungsproblem* by Professor M.H.A. Newman, and returned one year later with a solution, showing that there can be no solution to Hilbert's *Entscheidungsproblem*. To do this, Turing devised a mathematical model, later dubbed a *Turing machine*, that could—in the form of a process, no less—perform analogous operations to a *human* computer (then considered as a human armed with pen and paper, or a desk calculator)[44]:

---

[42]Specifically, Alonzo Church credited Peirce with the introduction of quantifiers (Brady 2000, 6).

[43](Church 1936).

[44](Soare 1996). Note that Turing's notion of human calculation is intrinsically technological.

> We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions $q_1, q_2, \ldots, q_R$ … The machine is supplied with a ’tape’ (the analogue of paper) running through it, and divided into sections (called ’squares’) each capable of bearing a ’symbol’... (Turing 1936)

What seems more significant is that Turing’s abstract ‘machine’ is based on a ‘human computer’ not in isolation, but armed with *inscriptional and/or calculative technology*. It may seem a trivial point, but in the (many) successive arguments regarding artificial intelligence (inspired by Turing’s later article, “Computing Machinery and Intelligence” (A. M. Turing 1950)) it is not always appreciated that at the (supposed) theoretical core of computer science is not a model of human computation, but *cyborg* computation. Moreover, as pen-and-paper calculations certainly comprise three out of four of Mitcham’s aspects (tool, technique, social practice) it is the last, volition, which we should address. It is easy to imagine a human computer following instructions while also being aware of the goals of their calculations; but it is equally easy to imagine the case where they are *not* aware of the intended goal, and are merely following rules (then dubbed a “programme”). In this case, we must locate volition not in the sociotechnical system composed solely of the human computer, but in a larger *organization* which employs the labor of that (human) computer. (This nearly returns us to Weber and Sombart’s conception of *Technik* as something done by “manufacturers who make calculations”.) In any event, it is worth appreciating that the perspective inspired by Weber, Stiegler, and Mitcham can open up a world of such social suppositions underlying what is supposedly a ‘formal’ mathematical model.

Processuality and Data: The finite vs. the potentially infinite

My eventual approach in this work will be to characterize technological developments regarding *data* in terms of a particular, arguably ontological distinction: that of the comparably static and finite and that of the *potentially infinite* (which I will dub *codata*).[45] This dichotomy is literally ancient; as J.R. Lucas explains in his *Conceptual Roots of Mathematics*, Aristotle (in the *Physics*, *Metaphysics*, and *De Caelo*) "argued against the 'actual' infinite and was willing to allow only the 'potential' infinite."[46] Lucas also mentions the dichotomy in the context of Brouwerian intuitionism (mentioned above) which took on this position in response to Cantor's radical cardinalities of infinities in the early 20th century.[47] To the extent that this dichotomy, in the case of data, is less well-known or even controversial, I think one cannot underestimate the profound pedagogical influence of the aforementioned set-theoretic mathematical foundations which, in many respects, tend to deny temporality, flux, and chaos even in the obvious presence of process-like behavior. One such influence—with strong implications for understanding the modern world of data, as well as for understanding much of the quantitative social sciences—is the connection between set-theoretic mathematical 'relations' and the representation of data in the form of inert *tables*; and it is this conceptual development that is the subject of the next chapter.

---

[45] As I will note later in Chapter 4, the term 'codata' dates to the work of (Hagino 1987a).

[46] Lucas (2000); In the works of Aristotle, see Physics III, §4-§8, 202b30 − 208, esp. 207b27 − 34; Metaphysics, K, 10, 1066a35 − 1067a37, De Caelo, I, §5-§7, 271b − 275b.

[47] Brouwer's position was directly influenced by Kant, who argued that both space and time could not be composed of mere points and instants, that:

> Points and instants are only limits, that is, mere positions which limit space and time. But positions always presuppose the intuitions which they limit or are intended to limit; and out of mere positions, viewed as constituents capable of being given prior to space or time, neither space nor time can be constructed. Such magnitudes may also be called *flowing*, since the synthesis of productive imagination involved in their production is a progression in time, and the continuity of time is ordinarily designated by the term flowing or flowing away (Kant (1965), A169–70/B211–12).

See also Posy (2008) on Brouwer and infinity; see Tieszen (2008) on the connection between Brouwer and Husserl and phenomenology. See van Atten (2012) also on the difference between Kant and Brouwer's perspectives. On Brouwer and Peirce, see Mayo-Wilson (2011).

CHAPTER 2

COMPUTING IN ORGANIZATIONS: ELECTRONIC DATA PROCESSING TO THE RELATIONAL MODEL

Introduction

This chapter addresses data management practices of the late 20th century, specifically in the context of their use by large formal organizations. The most significant processual development in the field of data management in this period is the rise and subsequent dominance of systems using the so-called relational model.[48] Such relational database systems—which organize information conceptually in a similar manner to the tabular layout of a spreadsheet, but with concurrent access, transactional reliability, and a flexible querying interface—ultimately became the dominant technology for storage and retrieval of structured data in commercial businesses and the de facto standard on the web.[49] They are also firmly at the technical core of a vast, global transformation of enterprise data processing and management that has taken place in recent years.

The relational model was far from an overnight success. Between its introduction to the nascent database research community in 1970 and the vibrant competition of functional implementations in the 1980s and 1990s lies a period of great argument and discussion over what amounts to a question worthy of the grandest philosophy: what is the most appropriate

[48] As I will explain below, the so-called 'relational' model of databases was introduced by IBM researcher E.F. Codd in his now-classic 1970 paper, "A Relational Model of Data for Large Shared Data Banks" (Codd 1970); popular present-day databases founded on the relational model include the open-source MySQL and PostgreSQL, Oracle SQL Server, and Microsoft SQL Server. As their names indicate, each uses a variant of the relational-based query language SQL, which originated in work by Donald Chamberlin and Raymond Boyce, also at IBM.

[49] Relational database systems—both open-source and commercial— underlie most contemporary content management systems like Wordpress; collaborative editing sites like Wikipedia; application frameworks like Rails and Django; transactional e-commerce systems; and enterprise resource planning systems like SAP's R/3.

representation of entities and their relationships in the world? This intellectual debate differed from others in that the answer was primarily determined by the interests of large multidivisional businesses, whose organizational difficulties a variety of electromechanical punched card and computing firms, led by IBM, had long been dedicated to resolve. The goal of this chapter is to connect bureaucratic practices (as initially theorized by Max Weber) to the distinctive properties of the relational database model, which are contrasted to previous (hierarchical and/or network) forms of data representation and management. This genealogy will familiarize the reader with the technology which would later be the site of development for the formalization of the transaction concept.

Relevant Aspects of Weberian Bureaucracy

A cornerstone of classical sociological theory, Max Weber's comprehensive generalization of bureaucracy in his *Economy and Society*[50] has influenced generations of scholars in the study of this most organizationally complex and technologically intricate of social structures. In his emphasis on the fundamental role of efficiency (in stark contrast to the already-predominant pejorative characterization of 'bureaucracy' as inefficient), Weber wove together a variety of seemingly-unrelated practices as aspects of a singular strategy which he called *rational-legal domination*.[51] While the subsequent hundred years have brought a variety of confrontations and disputations of the relevance of Weber's "ideal type" of bureaucracy, from

---

[50] (Max Weber 1922, 956–1005).

[51] This term is, specifically, in contrast to both traditional authority (e.g. patriarchialism, feudalism) and (individualistic) charismatic authority. (Max Weber 1922, 212–301).

empirically-based "tests"[52] to more contemporary theoretical revisions[53], few have attempted to supplant more than a small subset of his typology at a time. But in our case, we shall only be concerned with the aspects of Weber's theory of bureaucracy which can be fruitfully applied to the historical development of electronic data processing and database management systems.

In an effort to be simultaneously general and specific, Weber addresses multiple overlapping aspects of bureaucracy without ever giving primacy to one or another: it is a condition simultaneously (but not preferentially) characterized by jurisdiction, exhaustive rules, hierarchical authority and management, and the maintenance and control of 'the files'. For my purposes, the truly radical element of Weber's conception of bureaucracy is this singular emphasis on the importance of written documents, for it is in the development of database management systems that the use of "written documents"—today "written", of course, among and along an unfathomable variety of technological strata, including (at the lowest levels) disk drives, memory stores, and computer networks—achieves an organizational coextensivity unimaginable in Weber's lifetime. But even at the time, Weber saw the 'office' (*Bureau*)—"the combination of written documents and a continuous operation by officials"—as "the central focus of all types of modern organized action".[54]

In the course of his argument, Weber discusses at length the bureaucracy's fixed-salary, full-time official positions; notes the necessary presumption of a monetary economy; observes the dramatically increased scale of administrative tasks (in part via increased means of

---

[52] e.g., Arthur Stinchcombe's case study of construction management.; or Richard H. Hall's quantification of Weber's bureaucratic "dimensions" (Hall 1963).

[53] See Perrow (1986) for subsequent developments in the organizational sociology mainstream.

[54] (Max Weber 1922, 219).

communication); and also considers at length what he calls the “technical advantages” of bureaucratic organization, which will be especially important for us. These technical advantages include:

> Precision, speed, unambiguity, knowledge of the files, continuity, discretion, unity, strict subordination, reduction of friction and of material and personal costs—these are raised to the optimum point in the strictly bureaucratic administration, and especially in its monocratic form.[55]

It should be apparent to anyone familiar with the centrality and feature set of database technologies for all large organizations in the late 20th and early 21st century that we are dealing here with the first legitimate theorist of “data society”; for Weber’s list of aspects of the highly rationalized organization can also be plausibly read as a promotional list of features of a present-day commercial database.[56]

A further relevant topic regarding bureaucracy is its remarkable indestructability: “Once it is fully established,” Weber says, “bureaucracy is among those social structures which are the hardest to destroy.”[57] For him this is a direct result of the apparatus of authority combined with the functional specialization necessary at every position; he does not explicitly refer to the concentrated *bureau* of files as contributing to the formal organization’s improbably longevity. We can see further than he, however, and note the advances in data replication and storage which (with sufficiently continuous technical maintenance) gives the files an even greater (and more valuable) stability.

---

[55] (Max Weber 1922, 973).

[56] The term *file*, in the present-day computing sense, has its obvious origin in the bureau; before the advent of magnetic tape and disk drives, the computer ‘file’ was a stack of punch cards, each of which might have represented one entity (a supplier, say); additionally, the stack may have been pre-sorted in some way amenable to the computation at hand.

[57] (Max Weber 1922, 987).

Therefore, we can conclude by enumerating the aspects of bureaucracy which are likely to be most relevant for understanding the both early electronic data processing practices and also those of the database management systems which followed them in the latter half of the 20th century:

1. The unification and centralization of records.[58]

2. The efficient querying and processing of records.

To this we will add a third aspect which for Weber is not a generic property of all bureaucracy, but only for those organizations involved in what he called "rational economic profit-making"—the practice of *capital accounting*[59]*:*

3. The reliable and secure tracking of internal and external economic transactions.

Keeping these three particular volitional goals for the enterprise in mind, we can address the corresponding benefits (or drawbacks) of relational (and non-relational) databases as they emerged, dominated or receded—and in some cases, returned—beginning with the stirrings of high-profile debate in the (primarily Northern Californian) database research community of the 1970s. But as with any technology or set of technologies, the sociotechnical landscape of database management (much less *data* management, which would include the storage of less explicitly structured content like email) is not one of constant, linear progress but of a dynamic heterogeneity.[60] While managerial literature past and present may offer the illusory vision of a

---

[58] Note that centralization of *records* is not the same thing as the so-called centralization of the *organization*; as we shall see, centralization of the records in fact allows for greater decentralization of other aspects of the organization.

[59] (Max Weber 1922, 90-100,160-164).

[60] The discussion regarding technological heterogeneity in (Bijker, Hughes, and Pinch 1987, 191–94) refers to heterogeneities of use among groups of practitioners (of the 'same' technologies); I would also consider 'temporal'

business world progressively converging on the "best practices" of novel technologies, the reality is that—as seen in the 'Y2K' scare—some systems, developed at great cost in the mainframe era, continue to serve their purpose today, on updated (or simulated) operating systems and hardware (though many have undoubtedly been supplanted by enterprise resource planning systems). The result is that a single database management system rarely completely prevails in any large organization; and many contemporaneous smaller firms continue to make do with older functioning systems; or—as in the case of many 21st-century startups—adopt more recently-developed database technologies at the outset.[61] As such, one might consider this chapter a contribution to the social history of "popular" database culture, which considers the relational systems which generated the economic success of firms like Oracle and SAP at the expense of many smaller software companies and a plethora of less-widespread information retrieval applications.

The Role of Computing for Bureaucracy

Following the lead of James Cortada[62], we will consider the history of computing not as the history of one monolithic tool considered independently from its various manifestations and the uses to which they were put, but as a history of a heterogeneity of *applications* (by which we mean the social use of various tools and techniques—or the interactive embodiment of such practices as 'software'—for specific goals). We can then distinguish between the developments in the broader history of computing wrought specifically by scientific and engineering

---

heterogeneities in which older technological applications and newer technological applications are used concurrently, sometimes among identical practitioners.

[61] We shall see in the chapter 4 (on message-oriented 'middleware') that the centralized database itself was not sufficient to address this heterogeneity.

[62] (Cortada 1993, 132–33).

applications—including the first electronic, stored-program computers—and those changes driven by business and government applications with very different organizational goals, which began with simple efficiency of data processing but came to include the need for higher-level programming languages, larger data storage, and resistance to operational failure. While the scientific precursors have, within the history of computing, traditionally been given precedence of importance over business-centric perspectives[63], recent work has emphasized the sensible idea that the computer industry (led over a long period of time by IBM) should be seen primarily as in direct continuity with the office equipment industry (led over a long period of time by IBM).[64] While a strict logical separation of scientific/engineering applications and business/government applications in the case of data processing is not fully warranted—innovations on either side often cross-fertilized, especially during wartime and within the research departments of IBM (which, by the 1960s, was directly catering to both markets with related products)—the quantities of data storage used by managers and administrators traditionally outstripped those of scientific datasets.[65]

It is therefore of great interest that much of the early technological change within commercial and government bureaucracies—the widespread adoption of keypunches, electric tabulators, printers and sorters in the punched-card lineage of Herman Hollerith's innovation[66]—

---

[63] (Norberg 1990).

[64] (Haigh 2001b). Haigh uses the term "administrative computing" instead of "bureaucratic computing"; the latter is preferred here in part to revive Weber, but primarily because the former phrase has been recently adopted by a plethora of university IT departments.

[65] For more on the contrast in scale of prewar administrative and scientific use of calculating tasks, see Cortada (1993, 128–36). Lars Heide comments that for punched-card applications in the 1930s, a large scientific calculation might have used "several hundred thousand punched cards… in contrast to the several tens of millions of cards consumed four times a year by the Social Security Administration" (Heide 2009b, 250–51). For a dual timeline of developments in scientific and commercial computing from the 1950s onward, see Nolan (2000).

[66] See Norberg (1990).

occurred contemporaneously with an absolutely radical transformation of commercial organization, a story vividly told in the historian Alfred Chandler's influential book, *The Visible Hand*.[67] Chandler's history explains the rise of the large multidivisional business enterprise—which, by 1950, had become the standard form of what he calls "managerial capitalism"—from its origins in the small, traditional, family-owned firms that defined U.S. commerce at the beginning of the 19th century. He does not explicitly focus on developments in data processing and management; he is, instead, explaining the very foundations and processual creation of what we now call 'data processing' and 'management'. He describes how in 1840 "the most advanced accounting methods… were still those of [14th-century] Italian double-entry bookkeeping", and that "nowhere was the press of business enough to cause a merchant to delegate any of his tasks."[68] But such was the rise in complexity, communication, and safety concerns that came with industrial development—and railroad transport especially—that in the following decades, new levels of delegation and control became necessary. Chandler describes one pioneering railroad superintendent who in 1853 not only initiates impressively Weberesque principles of administration and subordination, but also puts into place a system of regular hourly and daily reports from engineers and rail agents via telegraph.[69] (Notice here that an industry predicated on physical decentralization appears to simultaneously demand a greater logical centralization of knowledge, management and control.)

Two computing-minded business historians subsequently influenced by Chandler's suggestive analysis are JoAnne Yates, whose monographs examined the transformations in

---

67 (A. D. Chandler 1977).

68 (A. D. Chandler 1977, 37–38).

69 (A. D. Chandler 1977, 101–3).

internal communication technologies and techniques through 1920 via case studies of the Illinois Central Railroad, Scovill, and Du Pont[70], as well as the insurance industry[71]; and the former IBM researcher James Cortada, with his books on the data-processing technology of the early office appliance industry[72] as well as a three-volume series on the adoption and use of computing technologies in various organizations and commercial industries.[73] Thomas Haigh has published an important series of articles that also adopts the data-processing perspective, and (most relevantly) builds up to a history of database systems.[74] The historian Michael Mahoney describes these developments in business-oriented computing history as helping to lead away from a history of the computer and toward a history of software.[75]

We can thus see evidence, in the last two decades, for a trend against a historiographic bias for 'electronic computing' and a corresponding trend towards 'data management and processing'—and my argument would be that favoring the former lineage necessarily leads to an overemphasis on the centrality of the Von Neumann stored-program architectures (and the related formal model by Turing), and a corresponding underemphasis on the Hollerith/IBM lineage of record-oriented data processing for commercial businesses which preceded and accompanied those developments.[76] For just as Shannon's formalization of 'information' was

---

[70] (JoAnne Yates 1989).

[71] (JoAnne Yates 2008).

[72] (Cortada 1993).

[73] (Cortada 2003); (Cortada 2005); (Cortada 2007).

[74] (Haigh 2001b); (Haigh 2006); (Haigh 2009).

[75] (Mahoney 2005).

[76] It is of interest that this science-and-mathematics vs. bureaucracy dichotomy has its analogy in the history of statistics as well, which also can be traced back to two domains with quite differing procedures (one a branch of mathematics, and the other an administrative activity) which—as described by (Desrosières 1998)—only began to

unconsciously adopted by generations of popular appropriators, we can see that Turing's formalization of 'computing' as the dynamic execution of machine instructions—symbolically encoded alongside non-executable data on a hypothetical infinite tape—itself leads to a notion of computers which emphasizes (the theoretically formalizable) programming and algorithms over the management and procedural processing of data.[77] Neither side represents the 'essence' of computation because there is none—computation (like all technologies) has always been a heterogeneous mix of tools, techniques, social formations, and volitions[78], and one can claim to find its origins in the astronomical tables which inspired Charles Babbage, or the statistical tables which inspired Herman Hollerith. What seems significant here is not what computing "really is", but instead the fact that both paths lead back to *tables*.

The Relational Model in Context

For the managers of large U.S. firms, the decade of the 1960s could be characterized as a time of great, and unfulfilled, technical promise.[79] As Thomas Haigh describes[80], a powerful mythology among managers of the day was the idea of a *Management Information System* (MIS) which would centralize all information then being acquired and processed independently in separate divisions of the organization; this dream was largely not achievable using the

---

merge in the 1940s. Sociologists today at times seem only dimly aware of the genealogical contradiction lying within this supposedly unitary discipline which mashes together state politics and mathematical probabilities.

[77] The innovation of Turing's machine abstraction, arguably, was to formalize symbolic transformation as a *process*—a term used frequently in his famous paper on computable numbers (Alan M. Turing 1936)—although this ontologically processual aspect is downplayed somewhat in Turing's post-hoc transposition to the foundations of computer science (Haigh 2014).

[78] (Mitcham 1994).

[79] For a discussion of the state of data management *before* the 1960s, see Haigh (2006).

[80] (Haigh 2001a).

technology and techniques of the era, although some complex proprietary data management systems had begun to be developed within individual large businesses. Examples include IBM's own Information Management System (IMS), developed with North American Aviation (with initial representatives from Caterpillar Tractor) to handle the hundreds of thousands of parts in the bill of material for the Apollo program's spacecraft[81]; and the Integrated Data Store (IDS), developed by Charles Bachman for General Electric's myriad manufacturing operations and first used in 1964.[82] The former, IMS (still in use today, for reasons we will explain later), became known as the exemplar of the hierarchical model of database systems, due to its required representation of entities in a tree-like manner branching from a root. The latter, Bachman's IDS, was the starting point for what became known as the network model, named as such because entities could be related in graph-like directed configurations, and not just hierarchically.

It is not insignificant that these early database systems were predicated on large-scale manufacturing operations;[83] the interest in databases was originally a symptom of only the most complex organizations, as they struggled from the weight of the subset of reality they sought to control, and searched for a stable ground (or "base") upon which it could support interrelated activities. Even before the possibility of integrating data from multiple departments (operations, accounts payable/receivable, payroll, billing, etc.)—and certainly before the concept of integrating across multiple geographic divisions—single departments had already reached the limits of existing systems.

---

[81] (Pugh, Johnson, and Palmer 1986a, 589–91). North American Aviation later became North American Rockwell and Rockwell International.

[82] (Bachman 2009).

[83] For those seeking an answer to why database systems were invented in the United States specifically, the country's dominance in manufacturing sectors in the 1950s and 1960s—see Cortada (2003, 66–88).

While this close connection between large commercial manufacturing firms and the development of database technology has presumably been long apparent to specialists, it bears mentioning just how closely related the concerns were. (Obviously, companies like IBM, RCA and Honeywell were also *manufacturing* computers and were thus doubly concerned.) The main computing consortium of the 1960s, Codasyl (Conference on Data Systems Languages), had an Executive Committee predominantly populated by represented by commercial businesses as opposed to academic institutions, with representatives from all the main computer firms[84] as well as representatives from metals (U.S. Steel), chemicals (DuPont, B.F. Goodrich), insurance (Prudential Life), and transportation (General Motors). Codasyl's Programming Language Committee had been responsible for the formulation (in 1959-1960) of the widely used Cobol, a "Common Business-Oriented Language" whose syntax was intended to be more readable to managers than scientific programming languages like Fortran.[85] It was their Data Base Task Group (DBTG) subcommittee—founded and initially chaired by W.G. Simmons of U.S. Steel—that was tasked with designing a specification for general database access; the DBTG's specifications in 1969 and 1971 were influenced strongly by both Cobol[86] and committee member Bachman's IDS. However, the Codasyl DBTG specification was not universally

---

[84] IBM and its competing so-called "seven dwarves"—Burroughs, UNIVAC, NCR, Control Data Corporation, Honeywell, RCA, and General Electric.

[85] Nathan Ensmenger, *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise* (MIT Press, 2012), 93-101.

[86] The COBOL influence was significant in, for example, their separation of the Data Definition Language (DDL) and Data Manipulation Language (DML), corresponding to Cobol's 'DATA DIVISION' and 'PROCEDURE DIVISION'.

acclaimed, especially by those who noted its complexity and, especially, its insufficient separation of programmer-level operations and the structuring of data on the physical device.[87]

It is in this context of this building of commercial demand—and correspondingly imperfect data management solutions (as well as a formal specification met with a not-insignificant amount of skepticism)—that Edgar F. "Ted" Codd's influential article, "A Relational Model of Data for Large Shared Data Banks", was published in the Association for Computing Machinery's flagship journal *Communications of the ACM*. Although the idea of the relational model took some time to catch on, as the decade progressed Codd's proposal came to occupy a prominent role at the controversial forefront of database research, despite not yet being implemented, commercially or otherwise. Codd, employed at IBM's San Jose Research Laboratory, had previously worked both inside and outside IBM on a variety of projects ranging from electronic calculation, data processing, and multiprogramming (the ability to run multiple jobs simultaneously on a single computer).

But unlike many IBM researchers who had been trained primarily in engineering, Codd had studied mathematics and chemistry at Oxford, and had recently completed a Ph.D. in Communication Sciences at the University of Michigan, working under Arthur W. Burks on cellular automata. Burks was a professor of philosophy who had previously worked at the University of Pennsylvania on the ENIAC, the world's first general-purpose computer; rather distinctively, Burks helped found Michigan's computer science program in the mid-1950s—then the 'Logic of Computers' program—while simultaneously editing the collected papers of Charles

[87] An example critique of the DBTG specification is (Engles 1971).

S. Peirce.[88] It would seem that Codd's immersion in the intricacies of both computer engineering and symbolic logic allowed him to view the pragmatic idea of data modeling in a distinctive way; but the resulting formal style in his 1970 paper also slowed uptake among less-mathematically-inclined researchers. The story of database research in the subsequent decade of the 1970s can be summarized in the gradual diffusion of the idea of the relational model and its (eventual) implementations; and eventually, the relational model made aspects of the long-fantasized "total information system"—which took one of its contemporary forms as the centralized Enterprise Resource Planning (ERP) installation—possible.[89] However, one can also be more detailed about the source of its transformative power.

I will argue that the success of the relational model can be understood on three more-or-less independent axes, which I will label *semiotic*, *bureaucratic*, and *transactional*:

1. The relational model differs primarily in its largely symbolic and tabular representation to the user, as opposed to the explicitly encoded referential relations of the hierarchical and network models to which it was opposed. This fundamental semiotic difference produces a highly valued effect recognized more typically as 'data independence'.

2. Somewhat surprisingly, the relational model mapped onto the cognitive practices of traditional administrative batch processing better than the IMS-style hierarchical model or the DBTG-style network model (which were both, in part, explicitly trying to preserve the batch processing "record-at-a-time" logic).

---

[88] The fact that one of Codd's advisors was a Peirce expert may make it not much of a stretch, then, to attempt to trace Codd's 'relational model' innovation to Peirce's explorations of applying predicate logic to real-world entities, which he called the 'logic of relatives'.

[89] It does not seem unreasonable to call it "revolutionary" in the Kuhnian sense, as its opposition (in the form of the Codasyl DBTG specification) is today almost completely forgotten among computing professionals.

3. The relational model, because of its symbolic representation of entity relations, was also highly amenable to the formalization of concurrent transactions, a technology which emerged and improved contemporaneously in the late 70s and early 80s. The ability to process atomic transactions at high speed while maintaining a consistent state was an enormous selling point for database management systems regardless of whether or not they used a relational model.[90]

The Semiotic Aspect of the Relational Model

Codd's 1970 paper begins with the statement:

> Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation).

In this statement Codd is addressing the call for a greater separation between what was at the time called *the logical* and *the physical*. The *logical* corresponded to the programmer (or "user")-level perspective of the data; the *physical* corresponded to some metaphorically 'lower' level of internal hardware (for example, specific locations on a disk drive). Before IBM's unbundling and the emergence of "software" as a commercial industry distinct from computer manufacture, the boundaries between one and the other could be fairly loose, and certainly varied impressively from computer to computer and system to system in the 1960s. But the history of revolutionary computer applications is, in large part, a history of techniques which successfully conceptually compartmentalize symbolic or communicative layers: such methods today (in software engineering) variously go under the name of *abstraction*, of *encapsulation*, of

[90] For those not familiar with the relational model, I refer the reader to (Darwen 2012); on the relational language SQL, see D. Chamberlin (2012). It is important to understand, at the very least, how the set-theoretic term 'relation' is unrelated to (and arguably in opposition to) its sociological sense.

protection. In 1970 Codd spoke of “data independence” (i.e., keeping the representation of the data separate from the underlying structure on disk), and in the 1960s a notable issue was of “machine independence” (developing applications which could function not just on IBM’s computers, but on those of other manufacturers). For example, the “high-level language” of Cobol allows a programmer to modify records encoded on punched cards without specifically acknowledging the underlying operating system or hardware.[91] Slightly predating such terms (but again based on this notion of independence between artifactual layers) was the notion of “generalization”[92], the idea that one could develop methods for processing data—sorting, for example—which did not depend explicitly on the particular deck of punched cards one was dealing with.[93]

Codd’s relational model was very distinctive from then-existing database models in having only one formal user-level conceptual data type: the *relation*, which can be thought of as a simple table of records (where the ordering of rows is unimportant). This allowed Codd to adopt certain algebraic operators as a starting point for a hypothetical high-level, interactive data-retrieval language. So, for example, the algebraic *projection* operator (of dimensions in a relation) became analogous to *selection* (of columns in a table); and the *natural join* operator could be used to connect tables into new tables, based on the shared values or one or more

---

[91] Similarly, the introduction of IBM’s System/360 in the mid-1960s was important because it offered a platform on which one could execute applications developed for smaller systems. One can still run applications developed for this platform today on contemporary IBM System z mainframes.

[92] On early generalized file processing, see W. M. McGee (1959).

[93] These continued (metaphorically vertical) divisions of symbolic strata eventually produced a conception of a layer-cake division of labor of computing systems, with electrical engineers tasked to the ‘bottom’ or ‘lower-most’ level, systems programmers above them, and so on to the interactional ‘top’ of user interface design; applications (or developers) which bridge sufficiently multiple strata are today referred to as “full stack”.

columns.[94] Codd noticed that the use of this so-called *relational algebra* provided a potentially vast simplification over the often-complex programmatic traversal of data structures necessary to retrieve data (at the user level) in existing hierarchical- and network-based systems; and in contemporaneous and subsequent papers, Codd developed the idea of using these operators to interface with a proposed ‘casual user’ of databases.[95] (This hypothetical query language would ultimately be incompletely realized in the now-ubiquitous *SQL*.)

But how did Codd’s proposal of representing both entities and relations as tables help with the data independence problem? In order to understand this, we can look to the ACM SIGFIDET[96] conference of May 1-3, 1974 in Ann Arbor, where the differences between the relational model and the network model from the DBTG network model were most explicitly presented, in the form of a gentleman’s duel between Codd and Charles Bachman. The two had publically clashed in the previous year’s SHARE conference in Montreal, where Bachman presented a draft version of his ACM Turing Award lecture.[97] Entitled “The Programmer as Navigator”, Bachman’s idealized interlocutor for database interaction is not at all the ‘casual user’ that Codd envisioned, but of an advanced programmer who “picks his way through the data to resolve an inquiry or to complete an update.”[98] At the end of the talk, Codd rushed to the

---

[94] The projection operator in the relational algebra is represented by the symbol π; the natural join operator is the symbol $\bowtie$.

[95] Codd combined these operators with expressions from his predicate-logic-based relational calculus, which he had shown to be reducible to relational algebra. See Codd (1972).

[96] SIGFIDET stands for the Special Interest Group on File Description and Translation; in 1975, it became SIGMOD (Management of Data).

[97] The proceedings for the 1973 SHARE Conference is in (Jardine 1974), and the final version of Bachman’s lecture is in (Bachman 1973).

[98] Ibid., p. 656.

microphone[99] and, while being relatively civil, admitted to "disagreeing entirely" with Bachman's vision. Instead, Codd said—referring to his own relational model—"we now have an opportunity to reverse the trend of making the logical view more and more complicated".

For the 1974 Ann Arbor conference, both Codd and Bachman would be giving presentations, alongside papers by supporters of their respective approaches. Codd worked with Chris Date (an early, and subsequently life-long, disciple of the relational model who worked at IBM Hursley)[100] to draft two papers: one—primarily by Codd—describing the relational approach's potentially superior support for non-programmers[101] and the other (primarily by Date) describing its benefits for programmers[102]. Codd's paper emphasized the relational model's goals as: 1) simplifying the logical data structures (i.e. using only tables as opposed to the cornucopia of data types in the network model), 2) allowing both programmers and non-programmers "to store and retrieve target data <u>without having to "navigate" to the target</u>" (emphasis in the original); 3) to introduce an English-language interface for truly casual users; and 4) to provide rules about authorized access (i.e. which user can see what data) and integrity (e.g. keeping the data in an appropriately consistent state) which can be specified separately from the logical view. While Codd was somewhat misguided about the near-term possibility of implementing a responsive natural-language interface (today, those working in business intelligence still need to learn either SQL or a clunky interface which can generate it), each of his

---

99 According to Chris Date in (Haigh 2007, 21).

100 IBM Hursley was (and is) a corporate campus in England in the countryside outside of Southampton, where researchers in the 1960s and 1970s worked on the programming language PL/1, and later the transaction monitor CICS (continuously developed and maintained there to this day).

101 (Codd and Date 1974).

102 (Date and Codd 1974).

points was a direct attack on a problematic aspect of the network model with respect to data independence; Bachman's side simply did not have a comparable vision for the use of the network model by "casual", non-programming users.

Date's paper, on the other hand, used a novel kind of diagrammatic rhetoric to contrast the two approaches, as seen in Fig. 2. By comparing the simplicity of the logical views of the relational model to the complexity of the network model, Date vividly demonstrates significant differences, which Bachman would often try to play down by suggesting that the two models could somehow be reconciled.[103] Indeed, I would argue that the diagrams in Fig. 2 can provide an immediate understanding, even to the non-specialist, of what makes the two approaches fundamentally different. As can be seen in the top half of Fig. 2, the relational model does not have, or explicitly require, any referential relations between the part (represented by table *p*) and the supplier (represented by table *s*). Instead, their relationship is represented by another table (labeled *sp*). With this third table, the association between attributes of *s* and *p* could be generated dynamically, as needed by the user, as opposed to existing as pointers at the logical level or (as in the computer programming sense of 'pointer'[104]) at the physical level.

---

[103] Over the following decade, Date would publish successive editions of the textbook *An Introduction to Database Systems* (Date 1975), which would make heavy rhetorical use of similar diagrams to help argue for the superiority of the relational model. Codd had previously used illustrations to compare the two models, albeit with more limited graphic skill, in (Codd 1971).

[104] A 'pointer' in computer programming is a value which represents a specific location in memory instead of some other data type, like an integer or character. If pointers with an invalid value are followed (or *dereferenced*) during the execution of a program, an error will result. Similar to the 'low-level' characterization of the network database model, programming languages which require extensive use of pointers (such as C) are considered to be 'lower-level' than those do not allow for explicit use of pointers (such as Java). In Peirce's terms, then, a pointer is a *symbol* which is interpreted *indexically* by some ongoing computational process.

S

S#	SNAME	STATUS	CITY
S1	SMITH	20	LONDON
S2	JONES	10	PARIS
S3	BLAKE	30	PARIS
S4	CLARK	20	LONDON
S5	ADAMS	30	ATHENS

P

P#	PNAME	COLOR	WEIGHT
P1	NUT	RED	12
P2	BOLT	GREEN	17
P3	SCREW	BLUE	17
P4	SCREW	RED	14
P5	CAM	BLUE	12
P6	COG	RED	19

SP

S#	P#	QTY
S1	P1	3
S1	P2	2
S1	P3	4
S1	P4	2
S1	P5	1
S1	P6	1
S2	P1	3
S2	P2	4
S3	P3	4
S3	P5	2
S4	P2	2
S4	P4	3
S4	P5	4
S5	P5	5

Figure 1.1.1: the suppliers-and-parts data model (relational approach)

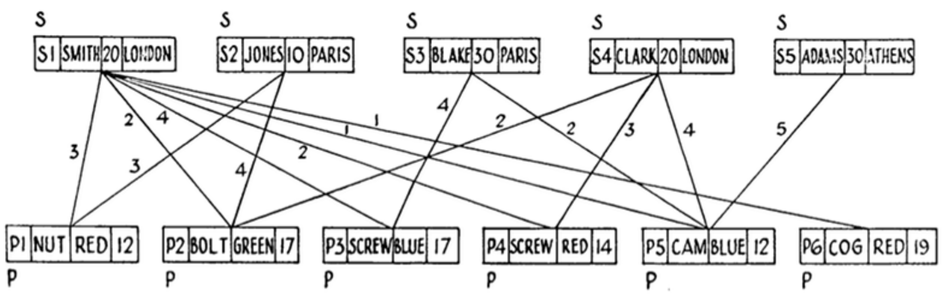

Figure 1.1.3: the suppliers-and-parts data model (network approach)

Figure 2**:** Date and Codd's diagrammatic comparison of the logical views of a relational database (top) and of a network database (bottom). From "The Relational and Network Approaches: Comparison of the Application Programming Interfaces" (Date and Codd 1974, 89–91; © 1974 Association for Computing Machinery, Inc. Reprinted by permission.)

One can compare this situation to the diagram of the network model (the bottom half of Fig. 2). Here, it is made clear that a network-based system of absolutely minimal complexity looks—on paper anyway—like something of a mess. Here, every supplier appears to have a direct, referential connection to each of the parts it supplies, and those referential connections themselves need to store extra information (in this case the number of parts supplied). But in fact, the situation is even worse than this. Date also provided diagrams showing the user's view of a DBTG implementation of the network model, as shown in Fig. 3. On the left side of Fig. 3, we see that in order to be able to relate parts to suppliers in both directions in a DBTG system, we need to explicitly create two objects to connect the records, *S-SP* and *P-SP* (called 'owner-coupled sets' in the DBTG specification). On the right side of Fig. 3, we see not the implementation of this model, but the actual programmer-level logical perspective (although one can imagine the physical level looking nearly identical—this was one of Codd's complaints with respect to the network model's lack of data independence). Date does not even have room to show all the links necessary for just these few records. Imagine adding a third entity to the table—e.g., including the warehouses in which the part is stored; an uncomfortably intricate spaghetti emerges. This complexity is what needs to be 'navigated' by the programmer, to use Bachman's term. By comparison, imagine adding a third entity to the relational model: one would add a table *W* listing each warehouse's attributes, and a table *WP* which related warehouses to the parts they stored, producing five tables in total—a comparative simplicity, and still readable at the logical level.

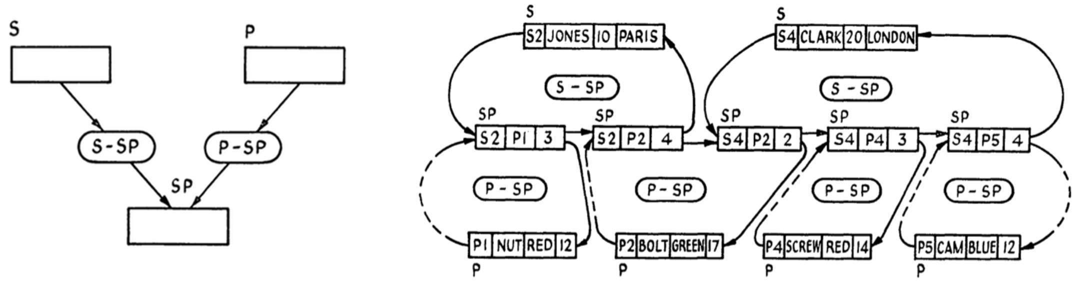

Figure 1.1.4: suppliers-and-parts data structure diagram

Figure 1.1.5: the suppliers-and-parts data model (DBTG part only)

Figure 3. Date's depiction of the Codasyl DBTG network model. From "The Relational and Network Approaches: Comparison of the Application Programming Interfaces" (Date and Codd 1974, 92–93; © 1974 Association for Computing Machinery, Inc. Reprinted by permission.)

Moreover, it should be clear that while in both model's representations, the parts and suppliers have as an attribute a unique, identifying symbol (*S1*, *S2*, *P1*, *P2*, etc.), it is only the relational model which takes explicit advantage of this identifier to represent relationships. A fundamental advantage of the relational model can be located, and this is it: the *referential* relationship of part and supplier is instead represented using *symbols*.[105] The distinction between the two models is thus fundamentally *semiotic*; where the network model's pointers mimic the indexical real-world physical relationship between part and supplier, the relational model represents that relationship in an explicitly symbolic, tabular form.[106] This tabular representation

[105] It should be noted that pointers are of course also encoded as symbols (as every value in a computer is); however, unlike a unique identifier like *S1* above, for a pointer to be useful it must be interpreted indexically (in the Peircean sense), i.e. as a direct reference to some other location in memory. This is to say, a 32-bit pointer with the hexadecimal value 0xefff00a0 is a useful value if it refers to a location in memory with relevant data; its meaning when interpreted symbolically—for example as the integer 4,026,466,464—is much less useful.

[106] Some researchers seemed to understand aspects of this better than others: in a 1974 paper presented at an international conference, G. M. Nijssen stated: "The relational model does not encompass the concept of predefined

is by no means “natural”, but it might be said that it is “natural” within a certain type of social structure—such as a commercial or administrative bureaucracy—which (as we know from the business historians) had been working and thinking with records and tables for many decades.[107] I would argue that this focal difference—of transducing reference to symbol—is at the heart of “data independence” in all of its forms.

So the crucial difference between the network model and the relational model should be clear. Where the network model enforces referential (i.e., pointing) links between entities at the logical level, the relational model enforces the *absence* of such reference. In this way, we can say that what the relational model allows for is a sort of freedom in recontextualization of the (entextualized) database artifact.[108] This freedom is realized in the so-called “expressiveness” of relational query languages like SQL, which (given an appropriately ‘normalized’ design)[109] allow one to relate — via joins and projections — new entities with every interaction. From a naïve perspective, this freedom would appear to be significantly less efficient (and such a complaint was taken quite seriously when computers were as comparatively slow as they were in the 1970s); it would seem that if I want to, for example, find the color of all of the parts supplied by supplier #2, I must exhaustively search the (unordered) part table *P* to locate each row based

---

navigational routes. *Either one may say that there exist no predefined navigational routes in the relational data model, or that all possible routes are dynamically materialized.*” [emphasis added] (Nijssen 1974).

[107] Another, possibly more radical, position is that the network model’s explicitly referential representation of relationships is no less artificial than a tabular one, with the additional detriment of being—in its typically “hairball”-esque way—more difficult to interpret, using computers or otherwise. Such an argument would have great implications for present-day trends in computational sociology as well as in the emerging digital humanities.

[108] The term *entextualization* derives from work in the anthropological study of performance, and is defined as “the process of rendering discourse extractable, of making a stretch of linguistic production into a unit—a text—that can be lifted out of its interactional setting.” From (Bauman and Briggs 1990, 73).

[109] A ‘normalized’ relational database is one which reduces the various represented objects (and relations between objects) into the simplest possible table, and was described at the outset in (Codd 1970). (A ‘fully denormalized’ relational database would be a single, very large table with plenty of missing values.)

on the part identifier, as opposed to simply following a pointer/link. The answer is that such searches, while definitely slower than following a pointer value, can be quite efficient; the technology which allows it is relatively independent from Codd's contribution, as it was already well known, having been used from the early days of print. This technology is known as an *index*.

Bureaucratic Aspects of the Relational Model

It would be difficult (if not pointless) to attempt to undertake a history of the origin of the practice of assigning individual numbers to observed entities in the world, for this technique seems rather natural to the numerate and is thus likely to have been independently re-discovered in a variety of periods and settings within cultures with writing practices.[110] However, there is reason to consider the (written) numbering of things as a fundamentally important technique, for it makes a variety of otherwise extremely difficult tasks possible. Consider, for one, the ability to quickly discover which pages of a book discuss a given subject; this is made possible by what we call an *index*, which maps subject terms to a list of page numbers.[111] Such subject indexing is a human art, but it is dependent on page numbers as a prerequisite of its technique.[112] The dependence on some kind of ordered identifier is, of course, commonplace in many organizations which depend on the accounting of *people* as opposed to pages. While smaller bureaucracies might simply use alphabetical ordering and resolve identity conflicts on a case-by-

[110] Braudel, for example, mentions that numbers were often assigned to individual medieval market stalls: see Braudel (1992, 28).

[111] In Peircean terms, an index entry is (broadly) a *dicent sign*: a symbol that is interpreted as a pointing reference (in this case via the iconic matching of page numbers). See Peirce (1931, 2:193).

[112] On 16th-century indexing techniques, see Eisenstein (2005, 70–81).

case basis (i.e. disambiguating two files for "John Smith"[113]), for more complex systems like the national census, or for life insurance agencies, a unique integer identifier is consistently used.[114]

In this section, I will describe how a particular data structure based on the use of unique identifiers—the *B-tree*—implements a similar kind of fast, indexed lookup, and thus provides a certain continuity of technique with the standard practices of the rationalized bureau. Indeed, for database management systems to succeed, 'the files' of a computer-assisted bureaucracy should be at least as accessible as those with rooms full of file cabinets. But before the innovation of random-access disk storage (which inspired the development of databases), what we now think of as computerized "data management" was in a primitive state. The use of magnetic tape for storage encouraged the rather limited use of that medium as a linear sequence of records, one after another, just as punched card records had been processed one after another—perhaps tabulated, perhaps updated, perhaps sorted and output to another tape—but always in serial batches.[115] (See Fig. 4 for an illustration, from a 1959 paper.) By comparison, the "random access" of disk storage—by which any given sector of the disk could be retrieved within a certain bounded window of time (as opposed to the lengthy playback necessary on tape-based media)—was crucial to the conception and implementation of applications capable of the storage and retrieval of multiple types of related records.[116]

---

[113] For example, many smaller U.S. banks did not have unique account numbers for their customers up through the 1950s.

[114] As the computer scientist Donald Knuth puts it, "When a large file must be searched, sequential scanning is almost out of the question, but an ordering relation simplifies the job enormously" (Knuth 1973, 406).

[115] (W. M. McGee 1959) provides an excellent survey of the state of the art at the period, with detailed descriptions of record processing, sorting, and report generation.

[116] For an excellent history of early developments in storage technology and its relation to data management, see Haigh (2009).

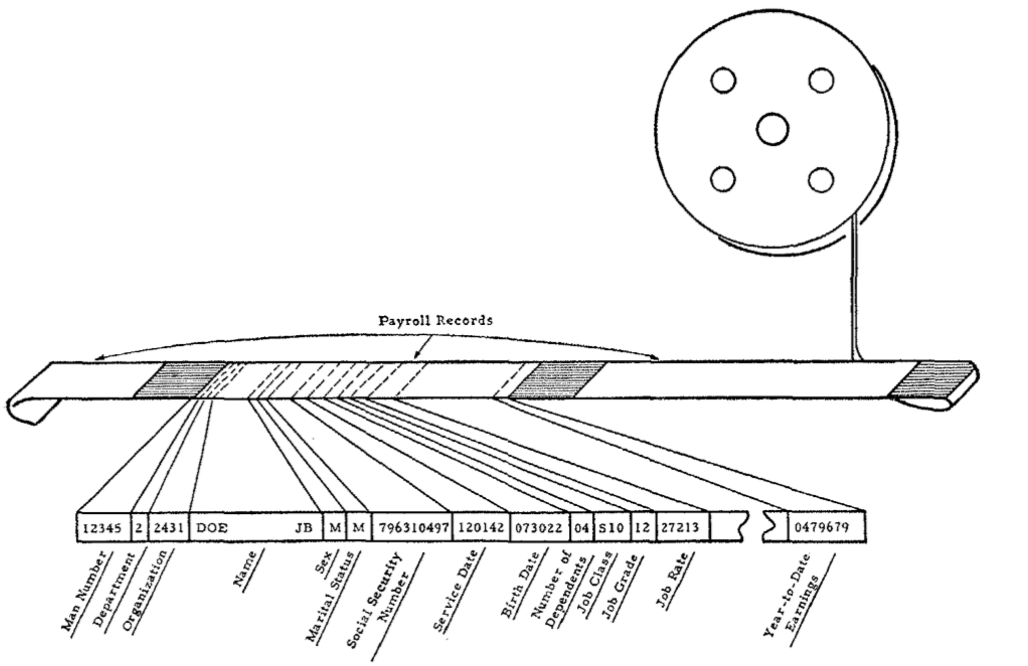

FIG. 3. Typical magnetic tape file

Figure 4. From "Generalization: Key to Successful Electronic Data Processing" (W. M. McGee 1959; © 1959 Association for Computing Machinery, Inc. Reprinted by permission).

But the feature of having random access to some given sector of a disk does not inherently reduce the complexity of inserting, deleting, or even searching for information from a particular record (say, e.g., an employee with identification #12345). Without some mechanism for efficiently translating from the identification number to a disk location, one would (just as on a linear tape drive) start at the beginning of the disk and scroll forward until one found the given record. The eventually widespread solution to this problem involved the progressive realization of a data structure which, in its minor variations, would come to be known as "the ubiquitous B-tree".[117] The B-tree is portrayed in Fig. 5: in order to find a record with a specific unique integer identifier (or, in relational model parlance, a *primary key*), we compare that integer with the

[117] (Comer 1979).

values at each level and traverse down to the appropriate leaf node, which eventually leads to a data page with the actual record stored in it; note that the number of traversals is small as compared with the total number of potential records stored.[118]

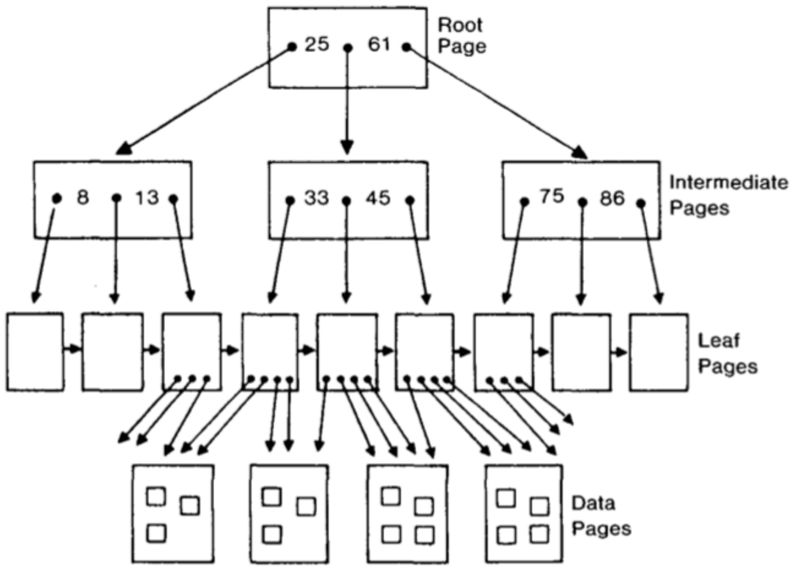

Figure 5. A B-tree index for a relation using an integer primary key, as used in the System R relational database. From "A History and Evaluation of System R" (D. D. Chamberlin et al. 1981; © 1981 Association for Computing Machinery, Inc. Reprinted by permission).

The novelty of the B-tree's design — in comparison to previously existing tree-based data structures— is its guarantee that no matter where records are inserted or removed, the tree remains 'balanced'—i.e., the expected search time is always logarithmically proportional to the

[118] One way to understand the B-tree is to note that its structure is conceptually similar to the organization of books in a large research library system (with, e.g., Library of Congress call numbers like *QA76.9.D3D370* as unique identifiers); physically searching for a book with such a call number is a similarly hierarchical process which can be quite efficient. First, one would go to the library with books in the *Q-QR* range (e.g., a specialized science library); then go to the floor of that library with the *QA* subclass; then go to the aisle with *QA76* in its labeled range; and then find the shelf or shelves with *QA76.9*; and then do a very short linear search on the shelf for the specific book (doing alphanumeric scans for *D3* and *D370* in succession).

number of records.[119] This is performed by an algorithmic "re-balancing" which occurs in the case that a new node needs to be added to an already-full leaf node (this process is shown in Fig. 6). The result is a consistent, relatively fast way to go from a primary key to the corresponding data; and the branching search technique is not all that dissimilar from the methods of the traditional bureau (narrowing down from rows of file cabinets, to a particular cabinet, to a particular file). If—as we described in the previous section—the relational model's distinguishing feature is the transduction of referential relationships to records of symbols, a B-tree is a fast way to go in the other direction. Without this technique, relational models would likely have remained as inefficient as their detractors often predicted.

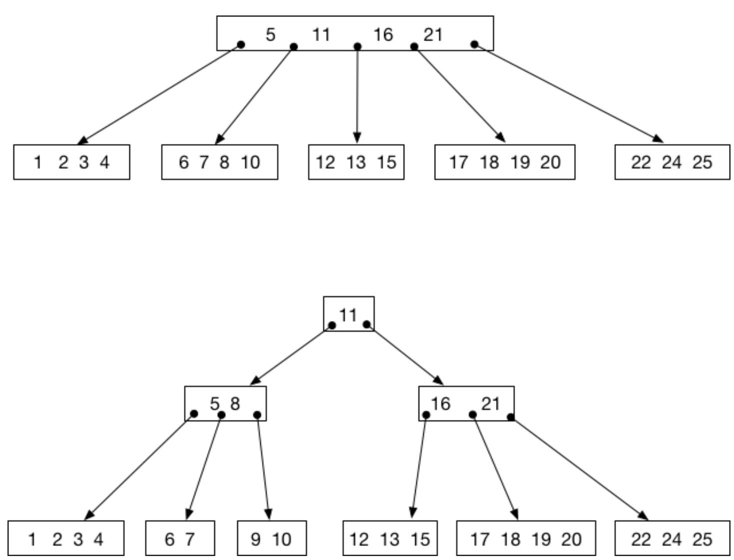

Figure 6. Diagram depicting the dynamic re-balancing of a B-tree upon inserting the value '9' into a full leaf. Both the leaf node and the root node split in succession, creating a new tree with three levels instead of two. Source: author, after Bayer (2002).

[119] (Bayer and McCreight 1972).

When combined with corroborating techniques like these efficient B-tree indexes, Codd's proposal of the relational model, in retrospect, seems ultimately less radical than (pragmatically) conservative. It is as if to say: computers are used most effectively for the processing of standardized records—that is, of structured and serialized sequences of symbols, extracted (i.e., entextualized) from their multivalent surrounding context. Therefore, make all data in the form of a standardized record. Date mentioned this rather explicitly in his 1974 SIGFIDET paper, describing the relational model's "closeness to traditional 'flat files'—i.e., to the enormous number of files which currently exist and are organized sequentially, especially on tape."[120] Codd was even more specific during his interrogation of Bachman at the 1973 SHARE conference:

> "In looking at the progress towards integration of files, we have noticed that the entities, previously processed separately, now have to be more and more heavily interrelated. This has resulted in the very elaborate data structure diagrams that we have seen displayed here... Now it so happens that a flat file and a collection of flat files has all of the power you want to deal with the n-dimensional world. You do not need to introduce any separate concept of relationships: the pointer-style relationships that we see with arrows on [Bachman's] diagrams. It so happens that you can consider the entities like parts, suppliers, and so forth and relationships like 'so-and-so supplies such-and-such a part' in one way, a single way, namely the flat-file way. What are the advantages of doing this? By adopting a uniform flat-file approach you can simplify tremendously the set of operators needed to extract all the information that can possibly be extracted from a data base."[121]

This simplicity of the relational approach—shrouded in appearance, for some, in the formalities of set-theoretic algebra and predicate logic—only became apparent to researchers at varying rates; its attempted implementations quickly converged on the need for indexing techniques like the B-tree which, indeed, possessed an isomorphism to existing bureaucratic

---

[120] (Date and Codd 1974, 95).

[121] (Jardine 1974, 157).

practice for data retrieval. However, there was one further set of intellectual developments—occurring somewhat orthogonally from the central arguments about the relational model—which, by the end of the 1980s, finally allowed relational database systems to completely transcend their role as a promising object of theoretical interest, and instead to take a dominant position in the commercial software landscape, one for which it has largely yet to yield to newcomers. This was the formalization of the *transaction*.

The Transaction Abstraction in the Relational Model

While Codd's 1970 paper title indeed makes reference to "Shared Data Banks", one should not be misled into believing that database management systems were born with the facility for concurrent, networked use. The technological landscape of multiuser (i.e., 'time-sharing') systems over the 1970s was quite diverse, and close reading reveals that many applications intending or claiming to support multiple simultaneous users did so at a coarse-grained scale which would seem unacceptable by present-day standards.[122] For example, banks throughout the period came to rely more and more on computers for high-speed, overnight batch processing; but systems which could perform what became known as *transaction processing* (later *on-line transaction processing*, or *OLTP*) were only being gradually adopted (as in, e.g., with relatively low-throughput ATMs.[123]) The idea of concurrent interactions with computers was not new; the implementation of *multiprogramming*, a goal of computer manufacturers throughout the 1960s, allowed multiple users' jobs to run on the processor in an interleaved

---

[122] For example, a web server can efficiently serve high numbers of overlapping requests; and that web server process may execute concurrently with database server processes (and many others) on the same operating system; and that operating system may in turn be virtualized—i.e., to run concurrently with other operating systems on a single computer. Such is the complex layering of concurrency in the text-artifacts of contemporary computing.

[123] (Bátiz-Lazo 2009).

fashion as they alternately executed or waited for input or output from some resource to complete.[124] But as larger random-access devices (combined with the aforementioned MIS mythology) inspired the development of database management systems, it was easy to envision transactional systems allowing for simultaneous access from more than one terminal. However, as we have seen, such work would need to be accompanied by a strong commercial demand.

As it turns out, the general problem of the limitations of written accounting of economic transactions had raised its head at many firms before the 20th century. The bound volumes of traditional double-entry bookkeeping were, as we would say today, single-user technologies.[125] It is easy to imagine how the limitations on concurrency in traditional record-keeping might have been a bottleneck in the otherwise rapid expansion of the multidivisional organization. One early organized workaround was the *clearinghouse*, whose early manifestations (such as the Bankers' Clearing House in early 19th-century London) centralized end-of-day record processing among multiple institutions, settling accounts far more efficiently than were they to be processed independently.[126] Ledgers were imperfect solutions to reliability and security, and themes of reliability (often under the label of "recovery" from inconsistencies or system error) and security became more prevalent in the database research literature in the late 1970s and early 1980s, as the debates on logical representation became—especially for the apostles of Codd at Berkeley's

[124] Facilities for multiprogramming were included the Burroughs B5000, and in some versions of IBM's OS/360 (for the System/360 mainframe).

[125] A brief discussion of problems regarding concurrent access to manual ledgers is in (Wooton and Wolk 2000).

[126] (Campbell-Kelly 2010).

INGRES[127], IBM's System R group, and other relational database implementers such as the nascent Oracle—largely settled.

By the term 'transaction abstraction' I refer to a set of concepts that coalesced into an increasingly formalized model in the 1970s and 1980s, and which has been differentially implemented on a variety of software systems ranging from the 1960s to the present.[128] Some aspects of the basic concept of transaction, however, extend back to the development of contract law; this fact has been noted by both computer scientists and organizational sociologists[129]. The transaction abstraction came, by the early 1980s, to be encompassed by a set of concepts using the acronym 'ACID'[130]:

atomicity: Individual transactions either occur or—if aborted—have no effect.

consistency: The data appears in a correct, valid state at all times.

isolation: Transactions are isolated from one another; this is equivalent to the appearance of serialized execution.

durability: Successful transactions persist and do not disappear or become corrupted in the case of failures.

---

[127] INGRES (Held, Stonebraker, and Wong 1975), which stood for "Interactive Graphics and Retrieval System" was a long-running relational research database at Berkeley, became commercialized in the 1980s, and is the origin of the currently popular open-source relational database known as PostgreSQL.

[128] The best early overview of the era is an essay by Jim Gray, "The Transaction Concept: Virtues and Limitations" (Jim Gray 1981); the canonical textbooks on transaction processing are Gray and Andreas Reuter's *Transaction Processing: Concepts and Techniques* (Jim Gray and Reuter 1992) and Bernstein et. al.'s *Concurrency Control and Recovery in Database Systems* (P. Bernstein, Hadzilacos, and Goodman 1987).

[129] (Jim Gray and Reuter 1992); (Williamson 1981).

[130] Theo Haerder and Andreas Reuter, "Principles of Transaction-Oriented Database Recovery," *ACM Computing Surveys* 15 (1983): 287–317.

These four concepts are not orthogonal (atomicity and isolation, for example, are closely related), and in fact it is worth noting that *consistency*, by virtue of having particular, sometimes complex, meanings for particular applications (such as making sure that debits and credits cancel out in an account transfer), is not an abstract, formalizable notion, but something whose semantics must be specifically ensured by each application's programmer. By comparison, the guarantees of atomicity, isolation and durability can be reasoned about more formally, under the guise of particular concepts and techniques such as *two-phase commit*, *serialization theory*, and *Undo-Redo protocols*.[131]

It is a remarkable fact that many comparatively ancient computing systems which successfully implemented transactional stability in the past have never completely gone away. From the airline booking system SABRE; to IBM's mainframe-based transaction monitor CICS; to IBM's hierarchical mainframe database IMS (whose logical design has gone thoroughly out of style, but which provides newer facilities for interacting with relational data); all of these systems have their origins in the 1960s and—far from disappearing along with so many other ephemeral software technologies—continue to run today on modern mainframes, allowing large organizations to continue to run hundreds of thousands of lines of code.[132] The notions of enforcing consistency and durability are crucial elements for any bureaucracy's shift away from paper-based bookkeeping, as it ensures that in many conceivable types of system failure, the data will still be successfully written (or successfully aborted). In the case of the relational model, transactions were often cited as a make-or-break feature for enterprise and/or financial usage of

[131] (P. Bernstein, Hadzilacos, and Goodman 1987); (Jim Gray and Reuter 1992).

[132] Both the histories of SABRE (Semi-Automated Business Research Environment) and CICS (Customer Information Control System) will be described at length in Chapter 3. For some discussion of the surprising continued success of SABRE (now known as TPF) and CICS, see Jim Gray and Reuter (1992, 42–43).

relational database products in the 1980s. In an interview, Don Haderle of IBM's System R project describes when System R's successor, DB2, came to be taken seriously:

> "So when version 2, release 1 came out in 1988, there was an article in Computerworld that said "Relational is now ready for primetime." It was simply because we had the features that were there for doing transaction processing. The customers came back and said it's competitive for that environment now. Good enough." (Grad 2007, 12)

Just as the intellectual debates regarding database models rotated around toy examples drawn from the context of large-scale manufacturing (Employee, Manager, Part, Supplier, Warehouse, etc.), the canonical example of the early transaction processing literature is the Debit-Credit transaction, which constituted a conceptual threshold for transactional sufficiency, while also reflecting the role of the banking industry, which held the earliest stake in the benefits of transaction processing. Fig. 7 shows an early diagram of the sort of staging that would characterize the transaction abstraction—here characterized by the ability to invertibly abort or "backout" of the process[133]; and Fig. 8, from (Jim Gray and Reuter 1992), shows how this "sphere of control" can be extended in theory to serial and parallel sequences of actions, using banking and manufacturing processes as examples.

[133] (Davies 1973).

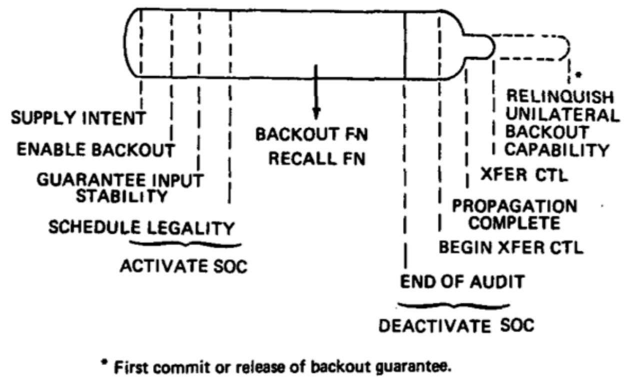

Fig. 1 Sequence of steps bounded by a sphere of control.

Figure 7. Charles T. Davies' early transaction abstraction, which he called a "sphere of control" (Davies 1973, 137; © 1973 Association for Computing Machinery, Inc. Reprinted by permission).

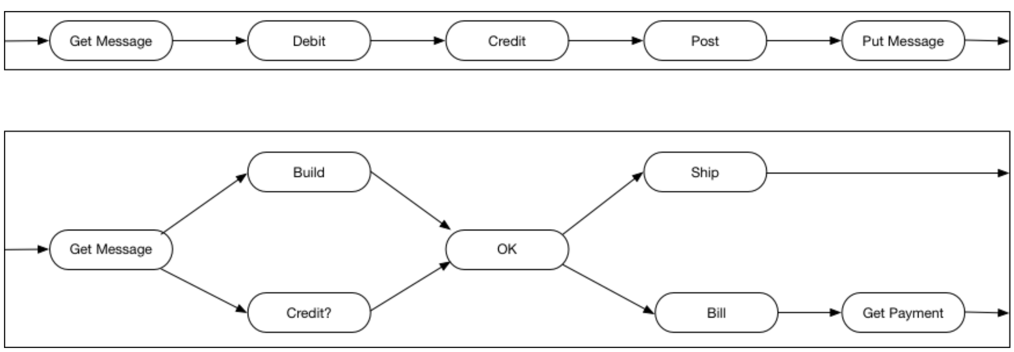

Figure 8. Jim Gray's transactions. Above, a simple debit-credit transaction. Below, a more complex manufacturing/billing transaction with parallelism. Source: author, after Gray and Reuter (1992).

Understanding the development of the transaction abstraction would give us great purchase into Donald MacKenzie's admonition to "open the black boxes" of global finance[134]. But as we have seen with the debate on relational vs. network models, it is not in fact always necessary to open up the black boxes *completely*—to the level of, e.g., the source code—for

[134] (MacKenzie 2005).

various competing individual implementations. One instead can derive great understanding of the transgressiveness of computational innovations (such as the relational model), in the absence of source code, through a historical examination of published (and unpublished) research and debate. However, we are at a serious disadvantage in that the techniques of transaction processing, unlike those of database management systems, have yet to be recognized by historians of computing and other social scientists as a locus of contemporary relevance. By contrast, what *has* been taken as an object of interest by social scientists are certain changes in economic phenomena, such as the vast increase in securities and derivatives exchange known as *financialization* and/or *marketization*[135]; or the expansion of multidivisional enterprises to (seemingly decentralized) transnational operations variously concentrated on (intriguingly centralized) metropolitan financial centers (*globalization*)[136]. Such developments are often illustrated with graphs showing an exponential rise in, e.g., historical trading volume (see Fig. 9), while seeking to describe such rapid change as a political or otherwise human-centric social process. It is thus little recognized that efficient and durable transaction processing—which permits the concurrent, orderly, reliable, and centralized serialization of economic exchange—is in fact a firm technical prerequisite of such financialization, and in general of the proliferation of globally linked continuous-trading markets. It must be understood that the simple increase in speed of single computer CPUs is not sufficient to explain this orders-of-magnitude scaling of computer-centered economic exchange that has occurred since the 1970s. The sociological and historical theorization of the modern formal transaction is necessary to understand this transformation, and is the subject of the next chapter.

[135] (Krippner 2012); (Çalıskan and Callon 2010).

[136] (Sassen 2007).

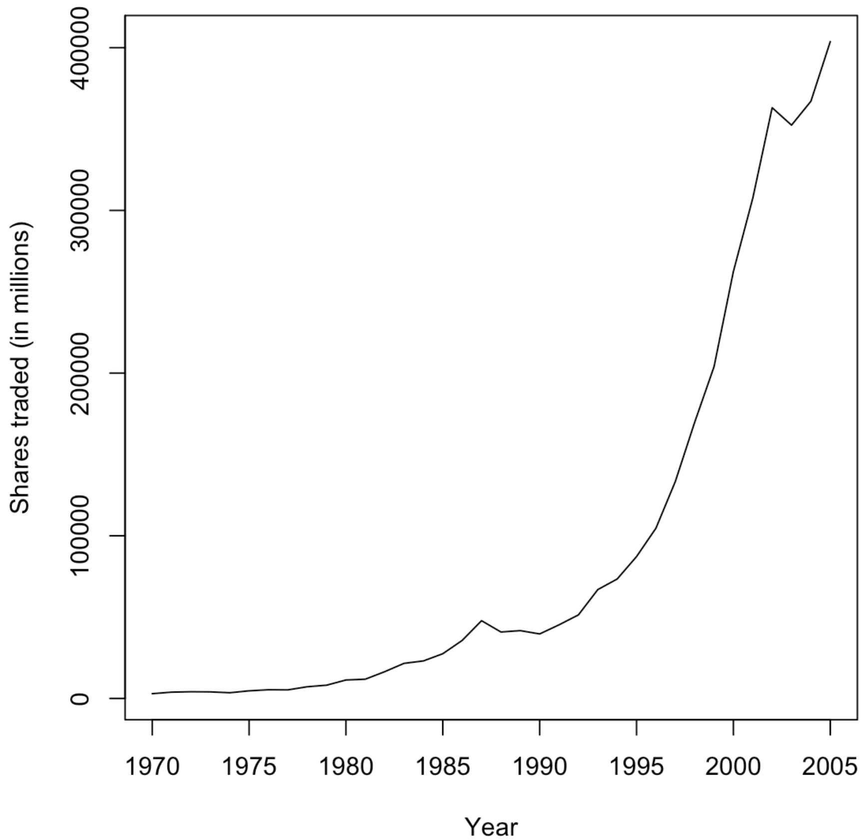

Figure 9: New York Stock Exchange trading volume, 1970-2005.
Figure by author; Data source: NYSE.

CHAPTER 3

THE TRANSACTION ABSTRACTION: FROM THE PAPERWORK CRISIS TO BLACK MONDAY

Introduction: What is a Transaction?

What is a *transaction*? Is it, as contemporary usage suggests, a form of exchange with a bounded beginning and end? Can one truly distinguish it from other forms of dyadic exchange, such as the *gift*, with its endogenous social repercussions? While the term 'transaction' is informally used in many historical and sociological discussions of economic activity, it is very rarely the direct subject of examination, despite its apparently fundamental centrality in economic activity—whether its impersonality is ascribed to the material mediation of money, of capital more generally, or of the rationalized technical activity at the core of the varied large formal organizations through which such transactions must pass. More pragmatically, the term *transaction* is used to denote the unit measure of the ever-expanding (and arguably, Promethean) practices of contemporary finance. Today we recognize the term as a phenomenon as likely to be digital as not, but how did it get there? Who formalized this genre of exchange, and who made its commercialization the basic infrastructure of the global financial industry?

This chapter makes the case for the technological formalization of the transaction—a set of explicit techniques emerging from research on shared databases in the late 1970s and early 1980s—as a fundamental prerequisite and facilitator for the varied economic and social processes of financial expansion, including those of *digitalization* (e.g. the increased use of integrated data systems in financial exchanges (Pardo-Guerra 2011); *financial mechanization* (e.g., the widespread virtualization of securities and derivatives (MacKenzie 2009) and the high-frequency automated trading thereof (MacKenzie 2012)); as well as *financialization*, the heightened centrality of finance in profit-making (Krippner 2012); or, more generally, of *marketization*: the establishment of social arrangements which decouple buyers, sellers, and items of offer ((Caliskan and Callon 2010); (H. White and Eccles 1987)).

With its pragmatic origins in the nascent research fields of operating systems and relational databases, the formalization of the transaction can be characterized by an introverted, ephemeral and highly centralized perspective of mediated exchange, which is in strong opposition to more traditional or holistic conceptions of social systems; it can also be seen as a technical apotheosis of organizational trends of control observed by the earliest sociologists of economic phenomena.[137] This modern representation of transaction, however, began as a practical solution to a very specific technical question: how should a computer system handle simultaneous requests contending to rapidly read and write from the same large data resource? Many organizations in business and government were, in the mid-20th century, becoming increasingly dependent on such data processing systems both offline and online[138], and the goals of maintaining data *consistency* as well as system *reliability* were of the utmost importance; but because early database systems were heterogeneous in interface and design, the facility for supporting high volumes of transactions was either implemented in an ad hoc manner or—in the case of the prototypes of the (eventually dominant) relational database—in a primitive state.

---

[137](Marx 1990), (Weber 1968 [1922]), (Simmel 2004 [1907]).

[138](Cortada 2003).

An Abbreviated History of Transactions

Forms of exchange are universal in human societies past and present; but not all exchange has qualities of the *ephemeral*, whether it be the bounded duration of the exchange itself; the impersonal relations of the traders (perhaps realized via the contractual and formal nature of their future obligations); or the symbolic character of the goods (e.g. a certificate instead of a container of ore). The concept of an exchange with such properties is not one which suits the variety of economies of the *gift* described by Mauss and many other anthropologists[139], and much of the work of economic sociology has been to bring forward aspects of social *embeddedness*, in allusion to the networks of relations which exist beyond the temporary relations of the individualized actors posited by economists' models (Polanyi 1944; Granovetter 1985). By contrast, this chapter—by taking the materiality of modern transactions far more seriously than it has been previously—finds the rapidly increasing flows of transactions of the late 20th century as a *sociotechnical fact* which recursively depends on the constitution and reliability of sociotechnical systems.

While the concept of the purely atemporal and impersonal transaction will remain an ideal type and not a social reality for quite some time, it is this form of standardized, switch-role, and ephemeral exchange which will bound this brief historical and theoretical survey.[140] We consider those arenas which saw the greatest concentrations of exchange activity, in which trade

[139] Mauss' definition of gift is indeed explicitly opposed to an ephemeral and impersonal sense of exchange: "prestations which are in theory voluntary, disinterested and spontaneous, but are in fact obligatory and interested" (Mauss 1954). One useful observation is the lack of accounting that occurs between close friends in contemporary Western life; as Graeber (2001, 218) puts it, "No accounts need be kept because the relation is not treated as if it will ever end."

[140] I will discuss the 'switch-role' category of markets more thoroughly in Chapters 5 and 6, but recall briefly from the introductory chapter that in a switch-role market, a buyer can also be a seller, and vice versa; and a *standard* market is one in which the good or service being exchanged is standardized that the respective status of the buyer and seller is irrelevant (Aspers 2007a).

was *least* likely to involve perduring social obligations outside those of the trading in question. (In the case of debt instruments like a loan which obliges periodic payment, the exchange may carry obligations; but these obligations are formal, quantified, and themselves tradable and convertible (as in the case with bills of exchange).) These sites include the town *marketplaces* of earliest civilizations, continuing through to the present; the larger and more periodic *fairs* which saw a concentration of trading and settlement activity across Western Europe; and finally, the *exchanges*, more permanent urban institutions characterized by distinct, continuous flows of trading activity.

On Centralized Exchange: The Marketplace in History

As our interest is in the development of dramatic and extensive *scaling* in the volume of transactions, primarily in the financial exchange industry of the late 20th century, it is crucial to consider the past sources of constraints on flows of centralized exchange. Before high-speed communication methods such centralization was achieved via the only method possible: geographical aggregation of buyers and sellers, in the form of the temporary social structure, widely perduring in a wide variety of historical and contemporary civilizations, which I will refer to as a *marketplace* (as in, a centralized place composed of *multiple* markets "in" various commodities, each of varying levels of standardization and thus fungibility).[141] These marketplaces of towns and cities, typically held weekly or twice a week, have been carefully depicted by Braudel (1992); in such structured, periodic aggregations, which one continues to find today in nearly all cities, fresh produce and other goods are sold, often still primarily in

[141] As discussed in the introductory chapter, I everywhere strive to distinguish fixed-role markets (such as production markets) from switch-role markets (such as financial markets) wherever possible, and to observe aggregations of such markets in differing goods and services not as a single "market" in their totality, but as a "*marketplace*"—i.e., a (spatial) union of different markets, and not a intersection or coherence.

cash. For Weber, marketplaces were a fundamental feature of all cities ancient and modern.[142] Braudel gives examples of the varieties of 17th and 18th century European markets, all characterized by geographic concentration: from the stalls and canopies and unlicensed peddlers of the town markets (each town had at least one), to the larger cities' covered markets (*halles*), which often contained specialized markets for cloth, seafood, and innumerable other goods.[143]

It must be said that these marketplaces were not devoid of constraints. They were sociotechnical structures which emerge from volitional interests, and what was bought and sold had a moral legitimacy among the participating social groups.[144] But Weber—writing in a fragment, entitled by his editor "The Market (*Die Marktvergemeinschaftung*); Its Impersonality and Ethic"—considers such markets the essentially exclusive province of the discipline of economics (deferring to Werner Sombart), saying that "The reason for the impersonality of the market is its matter-of-factness, its orientation to the commodity and only to that… Such absolute depersonalization is contrary to all the elementary forms of human relationship."[145] It would be some time before markets would be again a serious site of sociological inquiry.

---

[142] Weber does not conceive of cities as independent from the markets which make them possible. He states, "[w]e shall speak of a "city" in the economic sense of the word only if the local population satisfies an economically significant part of its everyday requirements in the local market, and if a significant part of the products bought there were acquired or produced specifically for sale on the market by the local population or that of the immediate hinterland. *A city, then, is always a market center.*" [emphasis mine] (Max Weber 1968, 1213).

[143] For a less European perspective on urban markets, see Geertz (1978) and Fischer and Gabbay (2009) on bazaars.

[144] (Aspers 2009).

[145] (Weber 1968, 636). On the development of Weber's thoughts on markets, see Swedberg (2000).

Clearing and Settlement

Like Weber, Braudel distinguishes his town markets from *fairs*, and from *exchanges*.[146] Fairs were widespread, but different from the (more regular) markets in their combining of small perishable transactions with larger dealer-to-dealer wholesale transactions, and in the settlement of debts (in the form of *bills of exchange*) that often concluded the events. Unlike the more frequent markets, fairs scaled to the size of small towns (sometimes overtaking the host town in question) and combined a wider variety of activities.[147] In the "higher" realms of these fairs—Braudel conceives of their economic activity as a pyramid, with perishable local goods and the bottom and the money market at the top[148]—we see the beginnings of "back-end" processes which resolve and conclude a geographically and temporally dense centralization of economic activity, where trading in different commodities had occurred on previous days. This was made possible due to the use of *bills of exchange*, which facilitated long-distance trade by allowing a buyer of goods to pay the seller with a debt instrument between two other trading parties (a debtor and a creditor), which could then be redeemed at a later date in a location convenient to the seller; this made it possible for the buyer to avoid using cash, and permitted larger variations in immediate liquidity on the part of buyers and sellers. The process of *clearing* includes bilateral clearing (reducing simultaneous debt relationships between two parties to a single debt) to multilateral clearing (doing the same for e.g. three parties in a debt triangle, or for more than three parties.)

---

[146] Weber's comments on fairs as markets for "long-distance trade of travelling merchants" are in his essay "The City (Non-Legitimate Domination)" (Max Weber 1968, 1212–36).

[147] Braudel (1992, 82–92). On the geography of European fairs see also Allix (1922).

[148] We use "money market" in its contemporary sense of a "market for short-term debt instruments" such as bills of exchange, described immediately below.

Exchanges and Clearinghouses in Western Europe

The *exchange*, I argue, was quite another form of social institution. Like those of the town markets, it is suggestive just how close the descriptions across centuries can be; as Braudel describes, "[t]he scene during the short business hours was almost invariably, from the seventeenth century at least, one of noisy close-packed throngs";[149] this distinctive pattern of social proximity and noise is one which indicates either an extreme of the marketplace pattern or a novel phenomenon in its entirety. I would argue that there were three reasons that exchanges became the site for the most intense and most concentrated volume of transactions: (1) *standardization* and fungibility of the goods exchanged, (2) those goods' *symbolic* representation as a relatively perduring text-artifact (such as a stock certificate); and finally, (3) the dissolution of the distinction between buying and selling.[150] While previous forms of markets may have had qualities of being *standard* or *switch-role* in these ways, the exchanges—in focusing and weaving those kinds of markets together in a single centralized site—were an environment distinct from the town markets and regional fairs which, entirely or in part, dealt in physical goods moving along circulatory chains of trade.

Exchanges (or *Bourses*) emerged in European towns such as Bruges and Antwerp in the 15th and 16th centuries. The most innovative was the Amsterdam Bourse—on which London's stock exchange was later based—which traded a variety of contracts including joint-stock company shares and commodity futures (B. Carruthers 1996, 103). Amsterdam differed from other exchanges in "[its] volume, the fluidity of the market and the publicity it received, and the

---

[149] (Braudel 1992, 99).

[150] On the role for liquidity of standardization of goods in markets, see the excellent B. G. Carruthers and Stinchcombe (1999).

speculative freedom of transactions" (Braudel 1992, 101). But it should be emphasized that many other continental exchanges would be characterized as low-liquidity: securities were issued in smaller amounts, which were in turn traded only intermittently among a small group of actors.[151] Even the Amsterdam exchange of the early 18th century did not have a formal structure; it was only in Paris in 1724 that an exchange became formally established.[152]

In the exchange there is a seeming paradox between this dense locus of ephemeral transactions and the properties of the securities transacted: specifically, that while at the same time the exchange of goods becomes depersonalized, ephemeralized, and disembedded from personal and institutional social relations, each traded security itself represents a formal obligation of future repayment which, in essence, puts a value to temporality. It is this "intertemporal transfer of value through time" [153] which is a characteristic of financial practice more generally, and we shall return to issues of temporality in the next chapter.

The London stock exchange in the 18th century transposed many of the properties of the Amsterdam exchange, and its description is recognizably analogous to the large U.S. stock exchanges of the mid-20th century with which we shall be concerned in this chapter:

> [The London exchange] was highly centralized (located in Exchange Alley in central London) and became organized around a group of financial specialists and brokers. A financial press developed that disseminated inexpensive and accurate information on share prices, exchange rates, interest rates, and so on. Standardized financial contracts were used to execute trades, including options and futures contracts. The overall size of the market grew very rapidly, both in the number of different securities traded and the numbers of trades and traders.

---

151 (Michie 1999).

152 (Michie 1999, 3): "This stock exchange both restricted entry to specialist intermediaries—*agents de change*—and had a code of conduct. However, the government limited its membership to only 60 and so a large and active alternative market continued to flourish in the streets outside the stock exchange building, and it was there that the advances in trading technique were being made."

153 (Goetzmann and Rouwenhorst 2005).

Econometric research suggests that it was an efficient capital market, as well. (B. Carruthers 1996, 13).

One important difference in the New York Stock Exchange (informally being established in 1792 and formally in 1817)[154] was its ownership structure; the building itself (constructed in 1863) was owned by the members, as opposed to London where there were non-member owners. The adoption of new technologies, such as the communication facilitated by the telephone and ticker tape, occurred sooner in New York than in London, in part because the exchange better represented the members' interests.[155]

Just as the fairs came to handle the centralized settlement of the increased volume of commercial and money-market exchange, the *clearinghouse* emerges as a (more urban) centralized location for netting and settling more standardized and symbolic volumes of exchange. The Bankers' Clearing House in London is described by none other than Charles Babbage (1832), with thirty clerks settling checks drawn from larger incumbent banks[156]; in securities trading, the London Stock Exchange set up a clearing house for stock (as opposed to sums of money) in 1874 (Norman 2011, 71–94).[157] The clearinghouse can be conceived at its outset as a "provider of calculative facilities for the detachment between counterparties" (Millo et al. 2005a).

The exchange (and especially its facilities for clearing and settlement) were thus the sites of the greatest density of centralization of transactional activity from the 19th century onward, and as securities markets expanded they would also be the site of the automation of that activity.

[154] (Ranald C. Michie 1986).

[155] (Ranald C. Michie 1986).

[156] On Babbage and the Bankers' Clearing House see Campbell-Kelly et al. (2014).

[157] On clearinghouses in finance see also Millo et al. (2005).

However, as much of the "computation" of exchange activity was scattered across a large number of brokerage firms, it would first be necessary for them to reach a crisis point.

The Paperwork Crisis on Wall Street

In the 1960s, the growing success and popularity of mutual funds helped fuel a dramatic rise in trading volume on the New York Stock Exchange.[158] While the individual *execution* of a trade went merely from customer to broker to clerk to floor broker (to verbal exchange with another floor broker), the *clearing* and *settlement* process, which at the time required the ultimate transfer of physical stock certificates, would involve "the branch office, the receiving department, the transfer analyst, the transfer department and the delivery department"[159], totaling at least a dozen distinct paper-generating activities. Increasingly harried and overworked brokerage back offices swelled with new employees—approximately 25,000 (primarily inexperienced) workers were added in 1967 alone[160]—but the brokerage companies, especially the smaller firms—were increasingly unable to keep up. So-called "fails"—i.e., transactions rendered incomplete by delays or other errors—began to reach staggering heights in 1968.[161] It was called the "back-office crisis" or the "paperwork crunch", and it perhaps illustrates the historic apotheosis of physically-mediated financial activity. Contemporaneous accounts of back-office environments describe a paper-bound chaos: "tables, desks, filing-cabinet tops, any

---

[158]Copeland, Mason, and McKenney (1995, 34).

[159](R. B. Smith 1970, 925–26).

[160]Seligman (1982, 456).

[161]To see the scale of "fails to deliver" within 30 days among securities firms in 1968 and 1969, see Morrison and Wilhelm Jr (2008). Benn (2000) explains how the larger firms' accountants, when faced with fails that matched (owing 200 shares of GM to one broker and being owed 200 shares to another), would cancel them, only to produce irreconcilable fails in the corresponding smaller firms.

available horizontal surface were by this time crammed with tall stacks of certificates."[162] In order to allow the back offices to catch up, in September 1967 trading hours were intermittently shortened by 90 minutes; by August 1968, NYSE trading was completely halted every Wednesday.

The flow of a security transaction at the NYSE in the 1960s can be described in better detail. A customer would place an order with their branch-office broker, who would in turn telephone it to the firm's order desk; the firm's order desk would telephone (or send via pneumatic tubes) to the firm's booth on the floor of the exchange, located on the perimeter of the trading floor; and the floor broker (represented by the firm) would take the order as a paper ticket to the location of the specialist, giving it either to the specialist or another trading interest on the floor. To report a transaction back to the customer would involve a similar process, in reverse (SEC 1988 Ch. 7, p. 16).

But this only describes the path an order would take to the specialist and back.[163] Upon execution, the specialist would prepare a machine-readable report of the execution (the "purchase and sales ticket"), forwarded to the purchase and sales (P & S) department of a given firm, which would attempt to keep the books balanced via a "trade blotter" and link with the clearing houses. P & S then forwarded transactions to the Margin Department, to check credit regulations and update the customer's statement; and purchased securities would (eventually) arrive at the Cashier's Department (or "Cage", characterized as an "intricate maze"[164]), the area

---

[162]Welles (1975, 170).

[163]The following details come from the detailed depiction of late-1960s era NYSE transactions in Robbins et al. (1969).

[164](Robbins et al. 1969, 27).

where stock certificates were physically handled, interacting with the "transfer agent" which would deliver certificates from the clearing corporation or brokerage. The volume of backlog which emerged in Cashier's Departments during the paperwork crisis was met with the "raiding" of back-office workers from other firms; with desperate new overtime procedures and a concomitant increase in unskilled workers (which sometimes led to securities theft[165]), and attempts to change over to automated procedures with computer installations; all of which was, for many firms, of little avail.

As one participant later summarized the situation:

> The brokers wound up unable to deliver securities because they couldn't get the physical certificates back from the transfer agents, and they wound up with a fail. Then when you have a fail, keeping track of where the dividends belong and all that sort of thing becomes a monumental accounting problem. Tremendous write-offs took place, and a lot of big brokers went out of business, like Goodbody and F.I. Dupont and others (Stocker 2011).

While some firms had made attempts to better automate their clearing processes (often via outsourcing to "service bureaus" specializing in data processing), analyses of attempted systemic upgrades of back-office processing in the late 1960s reveal how disastrous these early conversions could be. Welles (1975) details McDonnell & Company's implementation of a system called SECURE[166], but as costs and technical errors mounted, they could not avoid liquidation. In general, electronic data processing (EDP) equipment of the era, within an exchange industry lacking experienced programmers and well-defined engineering practices[167], could often not provide the necessary efficiency improvements. During this crisis period—

---

[165]Detailed in SEC (1971).

[166]"System to Eliminate and Correct Recurring Errors" (Welles 1975, 187). Another close look at failed automation of transactions is I. Martin (2012), which details how Barclays bank in England struggled to implement functional "real-time" systems by Burroughs in the late 1960s and early 1970s.

[167](SEC 1971, 176).

including and involving a subsequent economic decline of 1969 and 1970—115 firms in total departed the NYSE via merger, resignation or liquidation; the Wall Street Journal would later describe these firms as having "died from severe cases of too much business."[168] Amidst the crisis, both the NYSE and the American Stock Exchange (AMEX) looked to both RAND Corporation and North American Rockwell, respectively, in 1969, to analyze the paperwork crisis and make recommendations for automation. Both of these organizations had more extensive experience with computing than Wall Street—both had been founding members of the SHARE user group, formed mostly from aerospace and manufacturing clients of IBM in the Los Angeles area in the late 1950s.[169] As such, they had been deeply involved with the most complex and large-scale data processing projects of the 1960s.[170]

The contrasts in methods of RAND and North American Rockwell's studies are suggestive: while RAND's summary (Petruschell et al. 1970) describes their use of mathematical computer simulation of securities trading, which varied initial parameters to provide recommendations, North American Rockwell's study takes a significantly more modular approach. Their recommendations instead involved the streamlining of the processual pipeline of the more problematic modules within the system, such as the transfer agent and dividend clearance. Their goals were to provide "continuous transfer flow" via standardizing (e.g. machine-readable orders and certificates) and streamlining (conceiving of processing pipelines in which some processes can occur in parallel). They also proposed that a centralized depository

---

[168](Rustin 1975).

[169](Akera 2001).

[170]North American Rockwell had worked on the design of IBM's IMS (Information Management System), a hierarchical database system (still in existence today) which began as an attempt to inventory the very large bill of materials for the Saturn V moon rocket and the Apollo space vehicle (Pugh, Johnson, and Palmer 1986b, 589–91).

system and national clearing service be instituted where all stock market transactions could be realized.[171]

It may not be necessary to note that the situation in today's "back offices" has changed dramatically. Muniesa et al. (2011) describes each business division in a present-day investment bank's back office as being "constructed around the specificity of particular computer-based information systems"[172], with multiple major systems, both outsourced and in-house. In the next section I will describe the gradual change in finance technologies which helped the NYSE both avert another paperwork crisis, but which arguably laid the ground for a very different kind of exchange crisis: not one in which the back office is overwhelmed by a doubling in trading volume, but one in which the back office could reliably withstand it.

Changes in NYSE technology in the 1970s and early 1980s

With fewer small brokerage firms remaining and a less dramatic rate of change in transaction volume, the NYSE helped initiate technological changes over the 1970s and early 1980s which could help sustain a high number of transactions without completely eliminating the role of the specialist and the physical stock certificate. These changes can be separated at the outset as a) those involving *pre-trade* aspects of quotation dissemination and ordering and b) those involving *post-trade* activities (clearing and settlement). While some of these changes facilitated early so-called "program trading" (now known as *automated trading*), none truly

[171] SEC (1971, 179).

[172] Muniesa et al. (2011, 1194).

involved what is known as *automated trade execution*, the algorithmic matching and execution of market and/or limit orders.[173]

While the former modifications (those involving the communication of quotations and orders) can be thought of as facilitating an increase in networked interactions with market-makers, the latter transformations can be thought of as improving the organizational and technological bedrock, upon which the increased participation in the market can be made stable. It was obvious to many at the time that the paperwork crisis was in part predicated on the need to physically deliver stock certificates. While some proposed the replacement of certificates with punch cards (including North American Rockwell), and some the complete elimination of the certificate (as in Fischer Black's pioneering vision[174]); but since the 1950s the importance of, at the very least, centralizing the storage of (or *immobilizing*) certificates was well-understood. While a Central Certificate System (CCS) was organized in 1964, in 1969 its organization was at the time considered to be contributing instead of alleviating the backup in settlements.[175]

With respect to the former improvements in handling trading volume, perhaps the best way to understand these processes is to consider the digitization and virtualization of securities exchange in three tiers—each transforming some aspect of the existing human and paper-

---

[173]This is not to say automated trade execution was not implemented elsewhere during this period; Toronto's CATS (1977) performed automated trading of the less actively-traded issues on an open-limit order book system; variants of this system were later implemented in Tokyo in 1982 and Paris in 1986 (K. J. Cohen and Schwartz 1989). Domowitz (1990) provides an excellent early survey of automated execution systems at various exchanges. In the mid-1980s, automated trade execution was considered to only have been successful for small-order markets (e.g. Toronto) or those whose prices were set by a non-computerized primary exchange (such as the Cincinnati Stock Exchange's use of NYSE-set prices) — see (Amihud, Ho, and Schwartz 1984).

[174](Black (1971) and Black (1971b), discussed in (Pardo-Guerra et al. 2010).

[175](Wells 2000, 211); (Benn 2000, 15).

mediated specialist system of ordering and execution.[176] The first tier involves the electronic transmission of bid and ask quotes. At the second tier, there is the electronic transmission (but not execution) of limit and/or market orders. The third tier would provide for automatic matching and execution of said orders. In 1975, Congress amended the Securities Exchange Act to oblige the SEC to encourage the development of a "National Market System"; but discussions of its implementation were not always aware that this third step, which implies the need for a reliable centralized database handling an unknown number of potential requests to read and write, is of a differing order of magnitude in complexity than the mere electronic transmission of quotes and/or orders to screen displays.

Changes in electronic transmission of quotes and orders

In the late 1960s and early 1970s, the use of electronic communication for the dissemination of stock quotes was already quite extensive. Preda (2006) discusses the 19th-century development of the stock ticker, and certain subsequent technological enhancements with respect to quoting—from Teleregister's electronic quotation board to the Quotron I (1960) and Quotron II (1963) systems, as well as the use of NASDAQ by over-the-counter (OTC) brokers— can be thought of, in part, as practical extensions of the ticker concept (in the case of NASDAQ, orders and trading occurred between customers, brokers and market-makers over the phone).[177] The analogous groundwork at the NYSE was facilitated by the 1972 merger of their data-processing operations with those of the American Stock Exchange (AMEX) into a single

---

[176]These tiers follow those presented by Mendelson, Peake and Williams in their 1979 proposal for a national system (Mendelson, Peake, and R.T. Williams 1979, 55).

[177]On Quotron I and Quotron II, see Phister (1989); on the implementation of NASDAQ, see Mills (1972). It should be noted that while systems like NASDAQ were not transaction processing systems in the sense of effecting the ultimate *execution* of financial transactions (which were still handled by the brokers and the back offices), it nevertheless was a centralized data system supporting large numbers of concurrent readers (the retail brokers) and a small number of updates (the market-makers entering new quotes).

organization, the Securities Industry Automation Corporation (SIAC). Among SIAC's projects were the Common Message Switch (CMS) to directly connect member firms to an early centralized data system; this linked with NYSE's Market Data System (MDS) (which began operating in 1965).[178]

At the NYSE, the Designated Order Turnaround (DOT, an electronic order routing system) and Opening Automated Report Service (OARS, a batching system for the 500 most active stocks for the opening call) improved customer-to-specialist order flow. The DOT system, introduced in 1976 by SIAC, forwarded small orders[179], received via CMS directly to the specialist post, where they would be printed on "mark sense cards".[180] In bypassing the floor brokers and providing a unique symbolic identifier for small orders, the DOT system allowed specialists to handle a higher rate of trades while continuing to concentrate on larger orders on the floor. In part, DOT was intended to help the NYSE compete with regional exchanges which had begun to implement automated small order execution systems, although DOT would not automatically execute orders until 1984 (when it thus became known as "SuperDOT").[181] Even so, by 1984 DOT orders participated in about half of all NYSE transactions.[182]

Another important transition in the 1970s, related to the call for a National Market System encoded in the Securities Acts Amendments of 1975, was the increased interconnection

---

[178]On the significance of CMS and MDS, see Keith and Grody (1988). For comments on SIAC see Cortada (2005).

[179]At its outset, DOT would only accept market orders up to 299 shares; and limit orders, up to 500 shares (Seligman 1982, 531).

[180](Keith and Grody 1988, 93).

[181]SuperDOT could automatically execute orders when the NYSE quote was the best quote in the Intermarket Trading System (ITS) *and* the bid-ask spread was no more than 1/8th point. (SEC 1988, Ch. 7, 17-18).

[182](Calvin 1984).

between the NYSE and the regional exchanges. The Consolidated Tape System (CTS) (1974)—alluding to the traditional ticker tape—transferred trade data between the American Stock Exchange, the NYSE, and regional exchanges; and the Consolidated Quote System (1978) and Intermarket Trading System (ITS) (1978) enabled specialists and floor brokers to query quotation prices and transmit orders to market-makers at different exchanges.[183]

As described by Davis (1985), ITS was implemented by the exchanges as a response to Congress and the SEC's increasing emphasis on the securities industry's anticompetitive practices, as exemplified in the 1975 Securities Act. But far from increasing competition, in 1981 it was found that ITS accounted for just 3.1% of total NYSE volume; and that because response time could frequently be over 40 seconds, ITS was avoided during busy trading. Moreover, most of the trading taking place via ITS was for NYSE securities. As former NYSE executive Donald Calvin put it:

> [T]he assumption was, with the consolidated tape, consolidated quote, and ITS, the New York Stock Exchange is in big trouble. Ended up, they got ninety percent of the market. (Durr 2007)

With respect to the implementation of a National Market System, the conceptual tension between the value of competition (recognizing that intermarket competition should reduce transaction costs) and the importance of efficient "price discovery" (which, in the absence of technological barriers, would seem to be best facilitated by a centralized market) is, I argue, of relatively important theoretical interest. Instead of trading moving "off-board" (away from the NYSE floor), the extension of the NYSE quotation and order-placing network (combined with the partial automation of the CMS and DOT systems) allowed its specialists to enhance their

[183] U.S Congress Office of Technology Assessment (1990, 48).

relative network monopoly. The NYSE instead increased their power as the "centers of calculation"[184] for trading (and thus pricing) which formerly would have taken place at regional exchanges, and not the other way around.

Changes in clearing and settlement

While the NYSE's facility to handle increased volumes of trades after the Paperwork Crisis involved the aforementioned improvements in quotation and ordering, it is almost certainly more important to emphasize the increased centralization of systems which might have been previously distributed across each brokerage's back office. Under the guidance of the Banking and Securities Industry Committee (BASIC), formed in 1970, the existing central certificate organization, CCS, was developed into the Depository Trust Corporation (DTC) in 1973. And in an effort to automate the closing of positions, the National Securities Clearing Corporation (NSCC)—which combined the clearing corporations of the NYSE and NASDAQ—was formed in 1977, with SIAC as its facilities manager.[185] The smaller regional exchanges would eventually migrate their clearing and settlement operations to NSCC and DTC, representing an enormous consolidation of large-scale data-processing services that would not have been achievable without the development of scalable databases with support for transactions.

But most significantly, when the trading volume doubled during the "Black Monday" October 1987 crash—totaling 604 million shares in a single day, as compared to the approximately 19 million share volume that crippled the market in 1968—the NYSE's backend

---

[184](Latour 1987).

[185]U.S Congress Office of Technology Assessment (1984, 72). The NSCC and DTC merged in 1999 to form the Depository Trust & Clearing Corporation (DTCC). The comparable clearing entity for options trading is the Options Clearing Corporation (OCC).

systems were, surprisingly, able to cope with the dramatically high load.[186] What made possible not just this order-of-magnitude expansion in transaction volume, but the ability to withstand dramatic surges during the largest securities sell-off in history? As the Wall Street Journal explained two months later, the crash represented “one of the toughest tests undergone by any computer system” for SIAC’s two hundred “TNX and TXP mainframe computers, neatly stacked in about 50 refrigerator-sized cabinets”[187]: these machines were made by a 7,000-employee company called Tandem Computers, founded in 1974, based in Cupertino, CA, and which specialized in two domains which had made their systems invaluable for a wide variety of financial and commercial firms in the late 1980s: *fault tolerance* (resistance to failure in both hardware and software) and high-volume *transaction processing* (the explicit subject of this chapter).[188] To explain the origin of the prerequisites for the massive expansion in scale of financial markets, then, is to explain not just the success of Tandem; but specifically to understand the methods and ways of thinking at their technical core.

Early Transaction Processing in the 1950s and 1960s

Business computing in the 1950s—called at the time *electronic data processing* (EDP)—largely provided similar functionality to the punched card tabulation methods described in Chapter 2.[189] And prior to the mid-1950s, it is important to understand that the *flows* of punched

---

[186](Clemson 2012).

[187](Miller 1987).

[188]SIAC’s installation was, in fact, at the time Tandem’s largest installation (van Kirk 1989, 242).

[189]This point regarding 1950s business computing and tabulation methods is argued forcefully by Haigh (2001) and Cortada (1993). See W. M. McGee (1959) for a description of the data processing of the era.

cards were also limited: that is, inter-computer communication typically used physical paper tape, meaning that there would be of necessity a centralized and finite conversion process from incoming tape data to punched card data.[190]

Such finite, one-record-at-a-time data processing systems were sped up, but not significantly changed in structure, by the explicitly sequential access of magnetic tape. Systems with input and output on magnetic tape instead of punched cards or paper tape would thus also logically be considered 'batch' processing environments[191]. This form of computing should indeed be thought of as analogous to punched-card tabulation: one record at a time; non-interactive; and while each record may take a similar amount of time to process, there are no explicit guarantees about temporality inherent to the form. (Advanced EDP applications at the time might involve calculations based on serial input and output of several reels of tape.[192]) But three developments in the late 1950s and 1960s, each overlapping and intersecting with the others, sought to address the limitations of batch processing, and their combination represents the beginnings of transaction processing. *Multiprogramming* addressed the one-record-at-a-time problem; *interactive systems* addressed the inability to interact with the computer during a computation; and *real-time systems* addressed temporal constraints.[193]

The goal of *multiprogramming*—the art of interleaving data-manipulation processes so as to allow for computation and input/output access to occur simultaneously—would not be

---

[190] (Jarema and Sussenguth 1981, 392). Understanding this is crucial to seeing the pre-1950s limitations of the transduction from what, in Chapter 4, I will call *codata* (that is, data as a potentially unlimited flow) to "regular" data (i.e. data as a static and finite set of records, here embodied by a literal stack of punched cards).

[191](Hansen 2000)..

[192] (DeCarlo 1955).

[193] On multiprogramming, see Steel (1968). On interactive systems, see Schwartz (1968). On real-time systems, see Laplante, Rose, and Gracia-Watson (1995).

possible until one could reliably prevent different processes from accessing a given region of memory at the same time; 1960s computers implementing some form of multiprogramming included the Burroughs 5000/5500 and IBM's STRETCH.[194] Another major innovation in both implementing and reasoning about multiprogramming was Edsger Dijkstra's development of the concept of the *semaphore* in the 1960s—a digital flag, set atomically (i.e., without the possibility of interruption), which permits multiple computing processes to access shared data by organizing regions of mutual exclusion.[195]

The case for *interactive systems* was most forcefully proposed by the computer scientist and psychologist J.C.R. Licklider in 1960 (note the intersection of interactivity and real-time concerns):

> Present-day computers are designed primarily to solve preformulated problems or to process data according to predetermined procedures. The course of the computation may be conditional upon results obtained during the computation, but all the alternatives must be foreseen in advance… However, many problems… would be easier to solve, and they could be solved faster, through an intuitively guided trial-and-error procedure in which the computer cooperated, turning up flaws in the reasoning or revealing unexpected turns in the solution… The other main aim... is to bring computing machines effectively into processes of thinking that must go on in "real time", time that moves too fast to permit using computers in conventional ways. (Licklider 1960, 5).

Licklider, in discussing interactivity, necessarily invokes the notion of *real-time* systems, as interactivity presumes swift, back-and-forth responses. The phrase "real-time" is believed to derive initially from Jay Forrester's Project Whirlwind, which began with research into a flight

---

[194] The IBM 360 had limited support for multiprogramming, with 1967's OS/MFT operating system supporting only a fixed number of tasks; Dijkstra summarizes his critique of the IBM 360's multiprogramming support in Dijkstra (2001). We ignore for the moment *multiprocessing*, or simultaneous execution on more than one CPU. The term "time-sharing" was, before the 1960s, used synonymously with multiprogramming (sometimes hyphenated as "multi-programming"), as in the early papers by Strachey (1959) and Codd (1960), before taking on a more interactional resonance. For the emerging industry of time-sharing computers in the 1970s, see Campbell-Kelly and Garcia-Swartz (2008).

[195] (Dijkstra 1965)Dijkstra (1965).

simulator which would respond realistically to the pilot's actions and thus have "real response times".[196] A different kind of important "real time" application in the 1950s was the development of electronic telephone switching at Bell Labs. It is clear, though, that the notion of what constitutes "real" time is a subjective one, depending on the needs of the situational context. One definition which recognizes this (and which remains reasonable today) was published in J. Martin's popular book, *Programming Real-Time Systems:*

> [A] real-time computer system may be defined as one which controls an environment by receiving data, processing them and returning the results sufficiently quickly to affect the functioning of the environment at that time (J. Martin 1965).

One obvious sphere of application for real-time systems was, as we shall see below, seen to be the management of and response to continuous flows of information in military endeavors (e.g. from radar sensors). But the definition can also be applied to flows of more discretized activity, such as economic transactions, or simply for the interactive use of any monolithic computer. (In the latter case one sometimes came across the term "time-sharing" to connote real-time interactive use; but that term may be more simply associated with a multiprogramming machine with multiple terminals.) What should be clear is that all of these movements away from the batch processing scenario are concerned with various forms of indexicality and temporality. Any reference to the 'interactive' presupposes a direct connection between the data processing system and its social environment; any reference to 'real-time' presupposes a meaningful relationship between the computing process and the passage of time. 'Multiprogramming', then, can be seen as concerning the communicative *pragmatics* of interactive computing—making

---

[196] (Laplante, Rose, and Gracia-Watson 1995); (Redmond and Smith 1980).

possible the facility for more than one act of communication to be handled or processed at a time.[197]

However, this facility for multiprogramming—which, as we shall see, would prove so crucial for the formalization of transaction—was ill-supported by many existing operating systems; as such, the computing environments developed for concurrent transaction processing used *ad hoc* approaches which avoided the limitations of the computer's operating system altogether. These so-called *transaction monitors* instead implemented their own multiprogramming facilities inside a single monolithic process; this had its benefits (including that of functionality in an era of scarce operating-system support for concurrency) and its drawbacks (errors were difficult to diagnose due to the lack of modularity[198]). An exemplar of this type of transaction monitor application was IBM's CICS ("Customer Information Control System", pronounced "kicks"), released in 1969 for the OS/360 operating system and originally intended for public utilities (gas, electric) industries. It could handle 4 transactions per second, accessed by hundreds of networked terminals.[199] CICS was intended by IBM for clients with relatively less complicated data base structures; larger clients, such as those with large manufacturing bills of material (initially, aerospace companies), were expected to use IMS ("Information Management System"), an hierarchical database system.[200] Another, earlier example of a more specialized, domain-specific transaction monitor application was the IBM-

---

[197] Work on overlapping speech and interruptions in linguistic anthropology is not *quite* apropos here, as the simultaneity of programs was ideally conceived to be one of isolation, where running programs would not be "aware of each other". Isolation will become a larger theme in the formalization of transaction, below.

[198](Gray and Reuter 1992, 928).

[199](Campbell-Kelly 2003, 149–52); (Yelavich 1985).

[200](Haigh 2009); (McGee 2009). On the distinction between hierarchical, network, and relational database systems, see Chapter 2.

American Airlines reservation system SABRE.[201] SABRE's original name was *SABER*, standing for Semi-Automatic Business Environment Research; this acronym was inspired by the SAGE system (Semi-Automatic Ground Environment), an extremely large U.S. military project which attempted to centralize a variety of communications (including radar data, radio communication, and locations of aircraft and ships), suggesting that "Saber" would be for business what SAGE was for the military (Head 2002). In the following subsections we will examine more closely the origins and development of early transaction-processing systems like SABRE and CICS, examining their practical contexts of emergence.

Airlines over Airstrikes: SAGE vs. SABRE in the Origin of Transaction Processing

The *SAGE* project was the outcome of Jay Forrester's recasting of the aforementioned Project Whirlwind (which had been funded by the Office of Naval Research) for "command and control" military use, beginning in MIT's air defense lab. Whirlwind had begun as an analog flight simulator project that, as it developed, moved towards a general-purpose digital computer featuring high reliability.[202] In the early 1950s the Air Force's Air Defense Systems Engineering Committee (ADSEC), led by the MIT Professor George E. Valley, saw a connection between the tactics of ground command and Forrester's view of centralized control, and asked MIT to establish a laboratory for air-defense research; this was Project Charles (later named Project Lincoln, and in 1952 became MIT Lincoln Laboratory).[203] Lincoln's "Cape Cod System" focused on the problem of low-altitude surveillance by merging a variety of short-range radars

---

[201]Later versions were known as ACP (Airline Control Program), which was the operating system underneath IBM's PARS (Programmed Airline Reservation System) (Copeland and McKenney 1988); and TPF (Transaction Processing Facility) (Siwiec 1977); (Copeland, Mason, and McKenney 1995); (Campbell-Kelly 2003, 41–45).

[202] (Edwards 1996, 77–78).

[203] (Edwards 1996, 92–93); (Astrahan et al. 1957).

into a composite view in Cartesian coordinates on a CRT screen, on which an operator could associate other information using a light pen.[204] Lincoln selected IBM as the contractor to manufacture field computers based on their experimental implementation; in 1954 this project was named SAGE.

While IBM manufactured the hardware, the software was developed by the Rand Corporation, which spun off its own software division, the Systems Development Corporation (SDC), eventually employing 800 programmers.[205] Each SAGE "direction center" (of which there were over 20 scattered across the United States) was a confluence of many real-time streams of radar, weather, and troop information, receiving "digitally-coded data automatically and continuously… over voice-bandwidth communications circuits" (Everett, Zraket, and Benington 1957). What I want to ask here, then, is not commonly addressed or even, to my knowledge, asked: *in what ways did SAGE differ from SABRE?* And how did those differences—rather significant, as we shall see—lead SABRE, and not SAGE, to become a canonical exemplar for innovation in transaction processing? First, we will look at the development of SABRE, which emerged from a very specific sociotechnical problem.

While the development of SABRE is well documented in the secondary history-of-computing literature (e.g., Copeland, Mason, and McKenney (1995) and Head (2002)), my goal here is to show how practical transaction-processing systems first emerged *not* in reliable, real-time systems like SAGE but in very particular/distinctive contexts of real-world accounting; one need only consider the structure of American Airlines' reservation system in the 1930s to see

[204] (Astrahan et al. 1957).

[205] (Edwards 1996, 103).

why the need for transaction processing began there. Specifically, each airport maintained its own centralized seat inventory for each departing flight;[206] a sales agent thus was obliged to send a request to those in charge of inventory at the relevant departure airport, await a reply, and then communicate the success or failure of the reservation to the customer.

While there was an increase in demand for air travel after the end of the war, American's fleet (the U.S.' largest) in 1945 was still only 85 planes.[207] But these aircraft were, by today's standards, short-ranged, and longer-distance trips were broken up into shorter "legs", each of which was treated as a separate flight.[208] A reservations office design of the era (Fig. 10) involved agents sitting around a circular table, with the flight seating charts stored in a circular 'Lazy Susan' file (the centralization of the reservations process thus being quite explicit).

---

[206] (Copeland, Mason, and McKenney 1995).

[207] (Copeland, Mason, and McKenney 1995).

[208] (Dornian 1994).

Figure 10. ‘Lazy Susan’-centered reservations desk at American Airlines; © SABRE. Reprinted by permission.)

But the reservations offices at departure airports became increasingly congested, as in the below description of American’s Chicago office:

> A large cross-hatched board dominates one wall, its spaces filled with cryptic notes. At rows of desks sit busy men and women who continually glance from thick reference books to the wall display while continuously talking on the telephone and filling out cards. One man sitting in the back of the room is using field glasses to examine a change that has just been made high on the display board. Clerks and messengers carrying cards and sheets of paper hurry from files to automatic machines. The chatter of teletype and sound of card sorting equipment fills the air. (R.F. Meyer (1967); cited in Copeland, Mason, and McKenney (1995)).

An early attempt at improving affairs with electromechanical technology occurred in 1946, with the “Reservisor” system in which an “agent set” (a modified typewriter) sent booking requests to a partially automated, partially manual master control board at the given airport (in this case Boston). But there were still significant issues in both speed and consistency of information, even after various improvements to the Reservisor system in the 1950s:

> Approximately 8 percent of all transactions were in error, which was especially troublesome for an airline that prided itself on its customer service. To process a round-trip reservation between New York and Buffalo required the efforts of 12 people, at least 15

procedural steps, and up to 3 hours of elapsed time. Moreover, productivity was decreasing as passenger itineraries became more complex. The number of passengers boarded annually per reservations employee dropped from 5,100 in 1950 to 3,100 in 1958 (Copeland, Mason, and McKenney 1995).

The threat of the coming introduction of short-haul passenger jets like the Boeing 707 (with a 112-passenger capacity) encouraged American and IBM to propose "a system that would integrate reservations, ticketing, passenger check-in, boarding, air cargo, and management reporting under a centralized teleprocessing design philosophy."[209] Their vision would depend on a variety of technological developments, including the increased reliability of IBM's 7090 computers (which used solid-state logic instead of vacuum tubes) and, especially, random-access disk storage (as opposed to magnetic tape): for the sequence of queries in a reservation system would follow semi-random patterns in terms of which records might be accessed, added, or updated at any moment, for which the linear access of magnetic tape was not particularly amenable.[210]

The initial SABRE system, which went online in 1963, was implemented on dual IBM 7090s, using IBM 1301 disks, located at a site in Briarcliff, NY, north of New York City.[211] In 1965, there were approximately 1,000 agent sets at 60 different airports/locations.[212] Significantly, when TWA and United in 1965 attempted to develop competing (and arguably more ambitious) systems, contracted with Univac and Burroughs respectively (neither of whom

---

[209] (Copeland, Mason, and McKenney 1995, 34).

[210] (Copeland, Mason, and McKenney 1995, 37).

[211] (Siwiec 1977). Each of the sixteen 1301 disks are said to have held 13 million characters each; at 6 bits per character, that would be equal to, approximately, what we would call 9.3 megabytes.

[212] (Parker 1965).

had the teleprocessing experience of IBM’s SAGE work); these systems would not become successfully realized until the late 1970s.[213]

It seems plausible that the importance of SABRE has been exaggerated in the history of computing—certainly its secondary literature exceeds, to a surprising degree, more-widespread software systems like IBM’s CICS (and with respect to the number of transaction-processing systems ultimately deployed in other industries, airline reservations is merely one case)—one which nevertheless remains an important dual corollary to the flow of human airline passengers in an age of deregulation, mergers and globalization.[214]

And so, we can return to my previous question: how was SAGE different from SABRE? Why was the latter, and not the former, not considered an innovation in transaction processing? My answer is that one can understand a SAGE direction center as a *superposition* of continuous but largely independent data flows; for example, radar observations and weather observations do not overtly ‘interfere’ with each other in a centralized system, though they might be conveniently displayed on multiple video screens simultaneously and in real time. Moreover, SAGE was specifically predicated on the human interpretation of such a confluence of indexical information. SABRE, instead, relied on humans for input at the reservation desks, but humans were not necessarily intervening at the center of the system in Briarcliff except to keep it running.

---

[213] (Copeland and McKenney 1988, 357).

[214] However, even the scale of the largest airlines’ reservation systems remains relatively bounded by the relatively slow growth of maximum airport traffic; unlike a stock exchange, there is little potential for the volume of bookings to somehow double overnight.

And in contrast to SAGE, SABRE's problem of airline reservations is one in which many types of communications to the central system are potentially dependent on the next: e.g., one does not want to reserve the same seat on the same flight for two customers—or if a flight is cancelled, one does not want to make subsequent reservations on that flight. More so than the complex real-world contingencies of battle, the world of a given airline firm's reservations is—ideally, if not necessarily precisely in reality—represented as a stable *state of affairs* which only changes when a reservation is made, modified, or cancelled. The core reason for these differences is that, for SABRE, there is a central, standardized commodity at the center of its reservation system: the *airline seat*. Like a financial derivative, airline seat reservations are a form of contract that expires at a particular time. There was no such underlying commodity being bought or sold at the center of a SAGE installation, and thus, despite its "real-time" credentials and reliability in communication, SAGE should *not* be considered a transaction-processing system.

However, there were also similarities. SAGE, like transaction processing systems to come, was equally focused on reliability:

> IBM took many new measures to assure that the extreme reliability and continuous operation requirements for SAGE were met. *To assure continuous operation, any part of the computer system whose failure might bring down the system was duplexed.* Every SAGE direction center was equipped with two complete computers. At all times, one of the computers was active in air defense surveillance while the other was in a standby mode ready to be switched over into the active mode within seconds. The active computer continuously transmitted changes in the air situation data to the stand-by computer… so that the air situation picture would not have to be regenerated when switchover occurred (Crago, Tr. 85970-71; see also Case, Tr. 72251-53; Hurd, Tr. 86375) [emphasis added] (Mancke, Fisher, and McKie 1980, 77).

The use of duplexing components of a system for the purpose of reliability is a consistent theme in the history of computing (and especially in transaction processing), although it is an expensive solution. If SAGE had an overt influence on the future of transaction processing then, it was not in its system's design but in the use of duplexing.

CICS

The long-running *teleprocessing monitor* (or, later, *transaction monitor*) application CICS was originally proposed by the IBM systems programmer Ben Riggins, who in the 1960s was an IBM Systems Engineer working on projects for the Virginia Light and Power Company of Richmond, Virginia. At the time, certain utilities companies—which represented some of IBM's largest customers at the time— had become interested in a way for their customer service representatives, using terminals sending communications over short- or longer-distance cables, make inquiries and updates (such as bill payment or changes of service) to a central file system of some kind. (This origin is reflected in CICS' acronym: "Customer Information Control System"). IBM selected Riggins' proposal to be developed at their Des Plaines, Illinois office, which focused on industry software, along with Jerry Hughes, and Ray Vander Vliet, and Jerry Anderson.[215] An early version, PUCICS (Public Utility Customer Information Control System) was released in 1968, but it became clear that there was a demand for similar online applications outside the public utility sector, and CICS was released on July 8th 1969, licensed at a cost of $600/month.[216] It was one of IBM's first "unbundled" software products (along with the IMS/360 database and GIS, (Generalized Information System), a file management system). This original release could handle simultaneous connections from 50 terminals (communicating via the BTAM (Basic Telecommunications Access Method) protocol), requiring up to 35 KB of storage.

[215] (Rayns et. al. 2011).

[216] (IBM Software Group 2004).

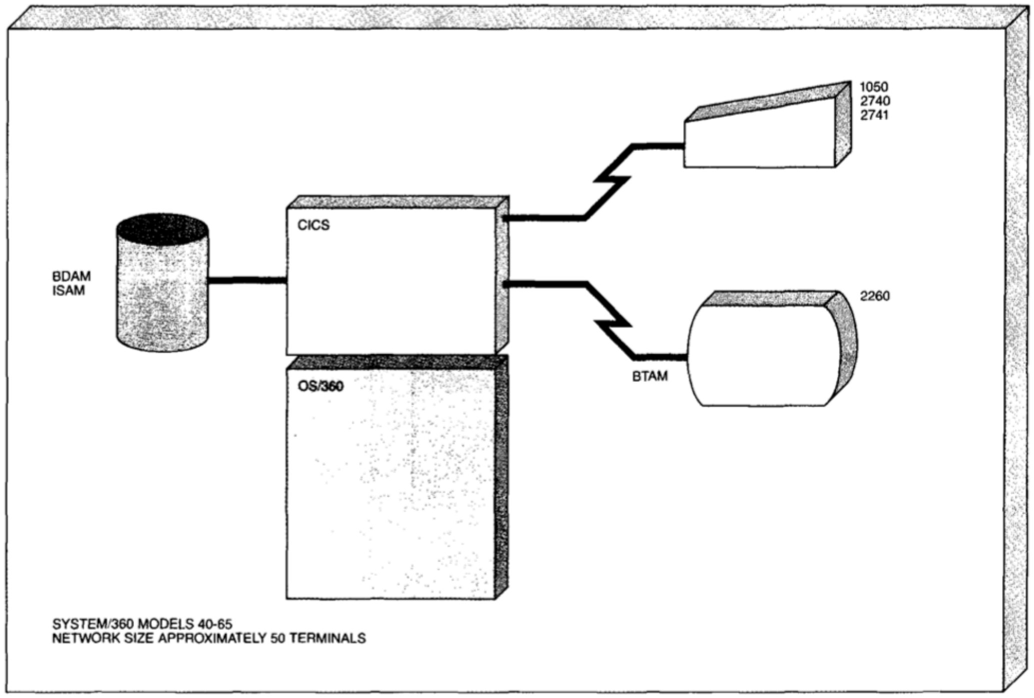

Figure 11. "CICS in 1968" (B. M. Yelavich 1985; © 1985 IBM. Reprinted by permission). This diagram shows CICS as an OS/360 application taking input from a terminal such as a typewriter ('1050') or keyboard/printer ('2740/2741') and using 'BTAM' (Basic Telecommunications Access Method) for data communications to an '2260' video display terminal. Data is stored and retrieved using 'BDAM' (Basic Direct Access Method) or 'ISAM' (Indexed Sequential Access Method).

CICS became successful in a far wider variety of industries than had been originally envisioned, including manufacturing, communications, insurance, and finance; and eventually, encompassing nearly every major commercial industry.[217] In 1974 the development was moved to IBM's Hursley campus in the United Kingdom, with 160 developers. The client base grew from just over 100 in 1971 to nearly 30,000 customers in 1987, when the Hursley team included

[217] (B. Yelavich 2003).

400 developers.[218] Even in the early 2000s, IBM claimed that 470 members of the Fortune 500 used CICS.

It should be stressed that many of the aspects of the transaction concept later described in this chapter were *not* intrinsic to early versions of CICS. For example, the ability to back out of a longer-running update transaction (i.e. to *undo* an operation) was not available until 1974.[219] But even by the early 1980s, transaction-processing support was only part of the story, as CICS had also become a general-purpose *middleware* application which helped bridge or connect the components of disparate IBM systems (and I shall address the history of middleware more generally in Chapter 4); but the long-lasting ubiquity of CICS speaks directly to the importance for many commercial applications of supporting simultaneous updates to a central data store, even if these updates modified an older hierarchical database like IMS, instead of the common relational databases of today.

From the *ad hoc* transaction to the formalized transaction

While the descendants of these early transaction processing applications have proved to be extremely successful, their initial innovations were not the subject of public research, and so they currently represent a sort of materialized lacunae in the history of ideas. Transaction processing was, in a sense, successfully implemented, but at the same time no scientific community had developed a coherent theory of how it might be achieved from first principles or independent of physical data representation.[220] Most importantly, these applications did not emerge fully formed: they added new features gradually as the formal theories of transactions

---

[218] (Campbell-Kelly 2003); Campbell-Kelly cites Mounce (1994).

[219] (B. Yelavich 2003).

[220] For more on the challenges of early transaction processing implementation, see Yelavich (1985).

emerged.[221] Finally, these systems also remained locked to the mainframe platforms for which they were originally developed.[222] CICS and IMS preceded the advent of two important computing technologies: the non-monolithic, modular operating system (e.g. Unix and its descendants) as well as of the *relational* database model, a form of data representation preferred by many large organizations due to its similarities to existing tabular record-keeping methods and its (comparably) convenient, algebraically inspired query language, SQL; this, for example, meant that none of the early transaction monitors supported what would be called today a "high-level" language or interface, or one that bridged IBM and non-IBM systems.

But in general, the two main issues addressed by transaction processing technologies can be described as *concurrency control* (how can multiple, networked clients operate on the same data source?) and *recovery* (how can we avoid losing data in the event of failure at various levels?) I argue below that these two issues, largely addressed separately throughout the 1960s and early 1970s, became intertwined as researchers attempted to implement reliable, multi-user database systems. What I want to show is how it came to be that contemporary commercial databases—which have been relied upon by nearly all medium-to-large formal organizations for decades—support some degree of concurrent transaction processing by default, and thus do not necessitate explicit management of shared resources for casual use; and what implications this has for sociotechnical practices such as securities trading or electronic commerce. Additionally—and perhaps most relevantly with respect to their impact on social structures—the failure rate of all of these hundreds of thousands of transaction-processing databases has become

---

[221]See W. C. McGee (2009, 69) to see how features of isolation, indexing, and recovery were added to IMS in the mid-to-late 1970s, for example.

[222]On the difficulty of porting SABRE's 300,000 lines of assembly, see NATO (1969); Today's surviving CICS and IMS applications run only on (now sometimes virtualized) IBM mainframes like z/OS (W. C. McGee 2009).

exceedingly low, even (and especially) within the largest organizations and at (comparatively) extremely high numbers of transactions per second. The history of the crucial techniques associated with handling concurrency control (e.g. *locking* methods) and those for handling database recovery (e.g. *logging* methods) will be discussed in the next section.

The Development of the Transaction Concept, Part 1: Spheres of Control

To follow the history of the transaction abstraction is to follow, to an extent, the career trajectory of Jim Gray, a prolific researcher at both IBM and Tandem, who won the Turing Award in 1988 for his contributions to transaction processing.[223] As a graduate student at Berkeley during the turbulent late 1960s, Gray became involved with a group led by Butler Lampson building the Cal Time-Sharing System (CAL TSS), an experimental modular operating system. In an early paper based on his CAL TSS experience, Gray proposed that Dijkstra's synchronization primitives (semaphores) were too minimalist, giving too much power to the programmer, who must be extremely careful to set and unset the locks in the appropriate order; he argued that they were ideal only "for cooperating, completely debugged processes".[224] By 1972 Gray had graduated and was working at IBM's Yorktown Heights Research Center.[225]

Two researchers also then at IBM who would prove to be influential to Gray's later work—Charles T. Davies, Jr. and Lawrence A. Bjork—were also developing a theory of what they called *spheres of control* which—as shown in Fig. 12—represented a very early and explicit

[223]Not without humility, Gray later claimed regarding the transaction abstraction: "A lot of other people did the same work. I happened to be first to publish it. I think it's been rediscovered or discovered dozens or hundreds of times and everybody who's discovered it or rediscovered it is proud of their discovery and rediscovery" (Winslet 2003).

[224](Gray 1970, 171).

[225](Frana 2003).

representation of 1) the need to *isolate* processes executing concurrently but accessing the same data; and 2) the need to handle potential failures of these isolated processes—to somehow rewind back to before their action (and/or dependent actions) had occurred.[226] Davies' initial report is significant in that it provides a completely system-independent focus on *recovery* as a theoretical notion, and takes an extremely pragmatic position on the very first page:

> No matter how hard everyone tries, there will always be some remaining errors, whether they be a manufacturer or a user's. It is these residual errors, or lack of quality, which give rise to the recovery problem in the first place (Davies Jr 1972).

Davies' paper also provides some early hints of the mutually related solutions to both recovery and concurrency control:

> ...to extracate [sic] from a deadlock, we need the backout capability described herein. For the purposes of scheduling, the set of functionally related resources must be considered as a single resource, e.g., A+B=C, is an atomic unit with respect to update (Davies 1973, 140)

To "backout" of a sphere of control is, as Davies puts it, "the undoing of the process in all nested spheres of control (iteratively) contained within this sphere of control."[227] This notion of *undo* has, of course, become a highly familiar one for users of today's computers; and it is helpful to think of a transaction as a kind of process which might be invertible, and undone at some point in the case of a problem arising during its execution.[228]

---

[226]Their two initial papers (Davies Jr (1972) and Bjork and Davies Jr (1972)) were internal IBM Research technical reports, which in 1973 were modified and presented at the ACM's annual conference (Davies 1973; Lawrence A. Bjork 1973).

[227]Davies (1973, 137).

[228]According to (Jim Gray and Reuter 1992, 575), the System R team was inspired by an 'undo' feature in the InterLisp programming environment of the early 1970s (Teitelman 1972).

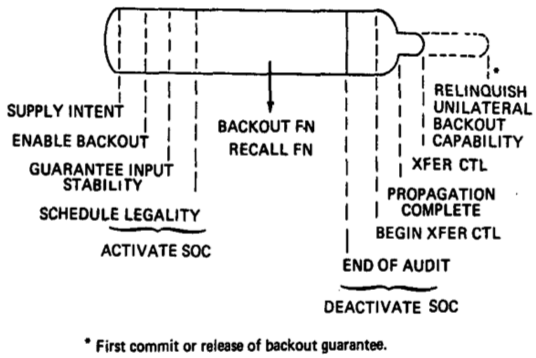

Fig. 1 Sequence of steps bounded by a sphere of control.

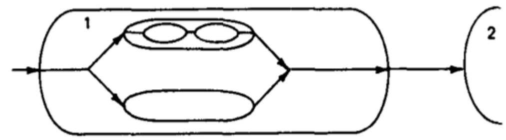

Fig. 2 Control flow among spheres of control.

Figure 12. Diagrams from Davies's "spheres of control" showing a sequence of steps necessary for backout/recovery from error (left) and a hypothetical diagram of nested transactions (right). (Davies 1973; © 1973 Association for Computing Machinery, Inc. Reprinted by permission.)

This conception of the transaction as an "atomic unit" would prove to be fundamental for transaction processing for databases—where every update to the data's "state", no matter the level of concurrent reading and writing, needs to act as an *all-or-nothing* process. Bjork's follow-up paper (see Fig. 13) further developed the idea of recovering from errors taking place within a particular sub-process within these "spheres of control".[229] He develops some conceptual strategies for unwinding and replaying processes with complex transactional dependencies, without going into any technical or algorithmic details. Its stark conclusion is that "recovery/integrity is not additive after system design, let alone after implementation", suggesting that the handling of locking and recovery would need to be a carefully designed internal component of any database system; indeed, this is what Jim Gray and the System R group eventually reported in their implementation of transactions (Jim Gray et al. 1981).

---

[229]Bjork (1973).

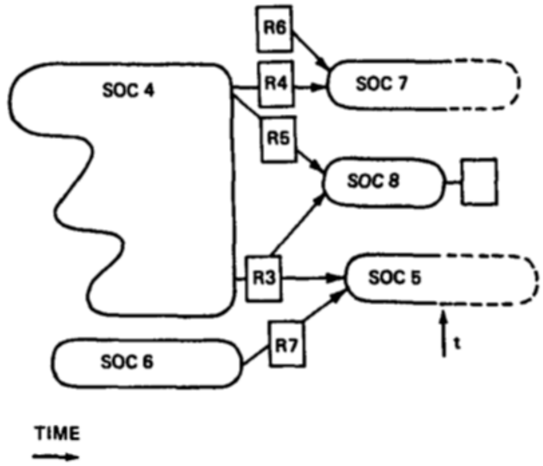

Fig. 5 Error detected in SOC5. Suspect R3 to exclusion of R7.

Figure 13. From Bjork, an early representation of a transaction dependency graph (Lawrence A. Bjork 1973; © 1973 Association for Computing Machinery, Inc. Reprinted by permission).

It is true that few people at the time understood the relevance of Bjork and Davies' articles (Gray and Reuter 1992, 222–23); for it described a computing environment wholly unlike any that existed at the time. (Transaction monitors like CICS and IMS, for example, had no support for isolating arbitrary sequences of transactions, for handling transactions with complex interrelating dependencies, or for nested transactions.) However, the Davies and Bjork papers represent the initial awareness of deep issues which have consumed large-scale computing practice to the present day (in the form of implementing transactions successfully across distributed databases, for example.)

The Development of the Transaction Concept, Part 2: Locking and Scheduling

By 1973, IBM had decided (not without some internal controversy) to assemble a group at San Jose—where Edgar F. "Ted" Codd, the progenitor of the relational database model, was located—which would aim to build a prototype, named "System R", based on Codd's proposals for relational databases.[230] The group included several researchers from IBM Yorktown—including Donald Chamberlin and Raymond Boyce, who would create the "Structured English Query Language" which would ultimately become SQL—who had undergone what they described as a "conversion experience" during a Ted Codd presentation at Yorktown in 1972.[231] Gray, who had since left Yorktown, returned to IBM to join the System R group in San Jose. Their early papers focused on the phenomena of *deadlock*, in which multiple processes (using Dijkstra's semaphores) would "block" on each other, bringing the operating system to a standstill (because, for example, process A was waiting for process B, and vice versa). Chamberlin, Boyce, and Traiger (1974), for example, showed how the literature on avoiding deadlocks was insufficient for a hypothetical shared database system, in which the resources may be described in a system-agnostic way (a command to "lock the employee-records of all red-haired employees" may conflict with a simultaneous request to lock the records of blue-eyed employees, or it may not); in which the requests for locks may be nested; and in which the *granularity* of lock requests may vary (i.e., sometimes a process will want to lock the entire database for modification; sometimes just a set of records; and sometimes just a single record). In these papers, the System R group struggled to come to terms with just how one might actually implement even a simplified version of the kind of world described by Davies and Bjork, one which could support simple *flat* (i.e. not nested) transactions; a simple example is a debit/credit operation which transfers symbolic funds from one record to another. They saw that without some form of transaction *isolation*, "there will be an instant during which one account has been

[230](Wade and Chamberlin 2012). Codd's formulation of the relational database model is in Codd (1970).

[231](McJones 1995). Their phrasing suggests a rich metaphor of the relational model's radical (and by many accounts, cognitively inaccessible) semiotic qualities spreading like an evangelical religion in the Bay Area.

debited and the other not credited" (Eswaran et al. 1976, 624). As such there was a focus on the notion of database *consistency* (also described by Davies Jr. (1972)):

> The data base is said to be consistent if it satisfies its assertions. In some cases, the data base must become temporarily inconsistent in order to transform it to a new consistent state. For example, adding a new employee involves several atomic actions and the updating of several fields. The data base may be inconsistent until all of these updates have been completed.
>
> To cope with these temporary inconsistencies, sequences of atomic actions are grouped to form transactions. *Transactions are the units of consistency.* They are larger atomic actions on the data base which transform it from one consistent state to a new consistent state [italic emphasis added] (Gray et al. 1976, 379).

It is important to consider, at this moment, what sort of environment the transaction abstraction presupposes. If the database is taken to be a model of the world (or more realistically, some relevant subset of it), then that world, at any moment, can be thought of as having a *state*. Transactions, as formalized by the System R group, are the legal transitions from one such state to another such state. Now, there are many aspects of "state transitions" in reality which are not so conveniently discretized (for example, the obligations you may feel if I present you with a gift); and there are others which have been discretized in symbolic form for quite some time—and whose exchange is often accompanied by an inscribed notation on a ledger—such as a monetized transfer of property rights, or the signed completion of a contract.

However, there was still the matter of how more than one of these state-modifying transactions could be executed concurrently. The group eventually rejected the notion of complex "predicate locks" (e.g. handling the above combination of locking sets of red-haired employees and blue-eyed employees); Gray et al. (1976) introduces a simpler granular lock hierarchy (data base, areas, files, and records), but preserves from Eswaran et al. (1976) the concept of a transaction *schedule* (later called a transaction *history*), which is a reordering of the sequences of reads and writes requested by concurrent transactions. The idea is that if two (or more) transactions request to read and write from a given resource over the same period of time, the job of a transaction scheduler would be to order those read and write operations in such a way that their interleaved operation would *appear*, from the perspective of the users who submitted the transaction request, that each transaction occurred one at a time without interruption. (The simplest possible scheduler, then, literally would run each transaction separately, one at a time.) This condition (of concurrent execution being precisely equivalent to some serial execution) is called *serializability*, and the mathematical methods for finding valid and efficient transaction histories became its own theoretical cottage industry (cf. Papadimitriou (1979)).[232]

Writing in the summer of 1977, Gray attempted to synthesize the collective knowledge developed by those then working on System R at IBM San Jose into a typewritten document called "Notes on Data Base Operating Systems", which would become a classic work in database research.[233] After giving credit to his co-workers and humbly admitting that the paper "plagiarizes the work of the large and anonymous army of people working in the field", his introduction sets the stage for the importance of the material to follow:

> *Most large institutions have now heavily invested in a data base system.* In general they have automated such clerical tasks as inventory control, order entry, or billing. These systems often support a worldwide network of hundreds of terminals. Their purpose is to reliably store and retrieve large quantities of data. *The life of many institutions is critically dependent on such systems, when the system is down the corporation has amnesia.*
>
> This puts an enormous burden on the implementers and operators of such systems. The systems must on the one hand be very high performance and on the other hand they must be very reliable [emphasis added] (Jim Gray 1978).

---

[232]As Gray would eventually succinctly put it, the topic of isolation "is variously called *consistency* (the static property), *concurrency control* (the problem), *serializability* (the theory), or *locking* (the technique)" (Gray and Reuter 1992, 375).

[233](Lindsay 2008).

This is to say that, in just a decade after the introduction of disk storage and networked computing, large organizations have, for Gray, taken the role of a hospitalized patient on life support (albeit stabilized), who has “amnesia” in the absence of its technical prostheses. His contrast of the simultaneous needs for high performance and reliability express precisely that intersection of *concurrency control* and *recovery* which would become common in discussions of transaction processing.[234]

Additionally, if one recalls the transformations of the NYSE after the paperwork crisis, one can consider the similarity between the strategic goals of a financial exchange (to handle increased volumes of transactions and avoid or quickly recover from failures) and those of a database system. While the early implementations of transaction processing in finance would occur in retail banking (as in the case of the relatively low-throughput networks of ATMs), they would quickly become of interest to other financial institutions.[235]

---

[234]See, for example, the more mathematically-oriented classic transaction-processing text *Concurrency Control and Recovery in Database Systems* (P. Bernstein, Hadzilacos, and Goodman 1987) as well as contemporary database textbooks like Silberschatz, Korth, and Sudarshan (2010).

[235]On ATMs, see Bátiz-Lazo, Haigh, and Stearns (2013).

Dijkstra (1965)	Semaphores
Davies Jr. (1972), Davies (1973)	Automatic recovery from failure; isolation of transactions ("spheres of control"); nested transactions
Bjork and Davies Jr. (1972), Bjork (1973)	Consistency; Serializability
Eswaran et al. (1976)	Atomicity; Two-phase commit
J. Gray et al. (1976)	Granular locks; degrees of consistency
Lomet (1977)	Atomic transactions; Database research linked to system reliability research
Jim Gray (1978)	Synthesis of knowledge from System R project
Chamberlin et al. (1981)	Overview of System R project
Jim Gray (1981)	The transaction concept: atomicity, consistency, durability
Jim Gray et al. (1981)	The System R recovery manager; Do-Undo-Redo protocol
Haerder and Reuter (1983)	"ACID" (atomicity, consistency, isolation, durability)
Anon. (1985)	Establishment of transaction processing benchmarks

Table 1. List of prominent papers in the history of development of the transaction concept.

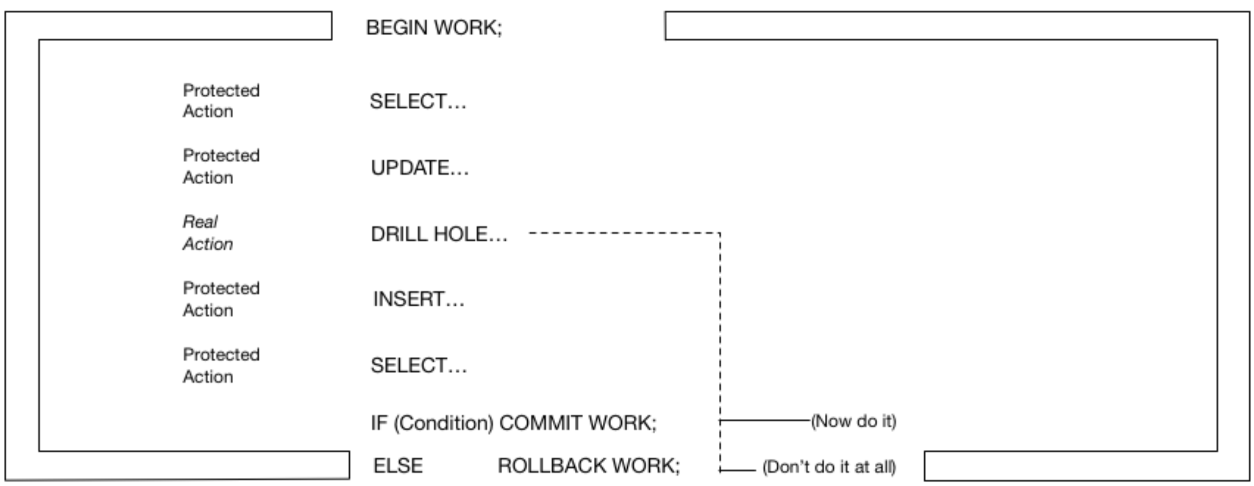

Figure 14. A transaction as a sequence of protected and real actions. Source: author, after Gray/Reuter (1992).

The Development of the Transaction Concept, Part 3: The ACID test

The properties of a database with transactions, as developed over the 1970s, were eventually summarized by the acronym "ACID", coined in a 1983 paper by two researchers from the University of Kaiserslautern, Theo Härder and his student Andreas Reuter (Haerder and Reuter 1983).[236] This acronym persists today as a convenient mnemonic for the properties of any database system that claims to support transactions:

- **A**tomicity: transactions are *all-or-nothing*: they either complete or never occur at all.
- **C**onsistency: transactions are valid transformations of the state of the system. (It should be noted that, as noted by Davies and others, the notion of consistency for a given application—such as a balanced checkbook—is application-dependent and *not* trivially formalizable in any way by the database system. Consistency is, thus, the semiotic odd man out in "ACID".)
- **I**solation: from the transaction's perspective, it is the only transaction running. This means that concurrently executing transactions see the stored data as if they were running serially, or one after another.
- **D**urability: Once a transaction commits, its changes survive system failures.

While we have discussed some of the implementation details for atomicity, consistency, and isolation, we have yet to discuss *durability*, although it is a feature on which the other three clearly depend; for example, if the hardware fails after a committed transaction and cannot recover to that state, we can provide no guarantees on atomicity, consistency, or isolation, regardless of how clever our algorithms were. The primary technique required for the implementation of durability is that of *write-ahead logging*: which ensures that sufficient information about the transaction (such as the facility to undo or redo) are committed to stable /

---

[236]The relational model—in the late 1970s still not widely understood or appreciated outside of the small circle in the Bay Area who had been convinced Ted Codd's proselytization (IBM San Jose's System R group; Michael Stonebraker's Ingres group at Berkeley; and Larry Ellison of Oracle)—had made its way to Germany via Reuter's advisor, Hartmut Wedekind, who had himself directly worked with Codd at IBM San Jose as a visiting scientist in 1972 (Härder and Lehner 2005).

non-volatile storage (e.g. disk) before finally being committed.[237] The technique called *two-phase commit* takes advantage of these log records to implement atomicity; Gray later analogized two-phase commit to the form of a wedding ceremony, which negotiates agreement ("I do") from all participants before the performative enactment of the marriage 'transaction'. Gray introduces this marriage analogy with a suggestive and more general statement that "[t]he transaction concept derives from contract law."[238]

It is worth reflecting further, if only for a moment, on this claim of an analogy between the ACID properties and those of legal contracts. For example, in the opening of a celebrated 1973 essay by the legal scholar Ian Macneil, he quotes the Restatements of the Law of Contracts, which expresses the all-or-nothing-like properties of the contract concept, which indeed has some analogical bearing on our transaction concept: "A contract is a promise or a set of promises for the breach of which the law gives a remedy, or the performance of which the law in some way recognizes as a duty." Macneil contends that "a long and unsuccessful struggle to reconcile this pure and simple concept with what seems to me the real life of contractual behavior has led to this essay", and that:

> The purity and simplicity of the traditional tenet arises from its presupposition that a contract is a discrete transaction. A transaction is *an event sensibly viewable separately from events preceding and following it, indeed from other events accompanying it temporally*— one engaging only small segments of the total personal beings of the participants [emphasis added] (Macneil 1973, 693).

Remarkably, Macneil is here defining a 'transaction'—independently from any research in computing— as one which is, indeed, *atomic* and *isolated*. His argument is that many contracts observed in empirical reality cannot be easily reduced to the simplicity of an ephemeral

---

[237](Gray 1978, 464).

[238](Gray 1981). On two-phase commit, see Gray (1978) and Lampson and Sturgis (1979).

‘transaction’. Macneil’s paper was, as it turns out, the starting point for a broader theory of *relational contracts*, in which a processual, contextual, and social element was brought to a theory of contracts. And so it is important to see that the formalization of transaction embodied in the ACID properties is defining precisely such a discretized, decontextualized, and asocial operation. Understanding this allows us to appreciate why some institutions have more fully embraced the specific computer systems which made extensive use of the formalized transaction, while others did not: specifically, the formalized transaction is most precisely amenable to those “contracts” of the shortest and least-socially-perduring kind—such as the trading of a stock between strangers, as negotiated by their representing brokers—and despite the theoretical possibility of longer-term transactions (as, indeed, originally suggested by Bjork and Davies), we shall see that transaction processing systems were, in practice, made efficient only for particular types of brief, decontextualized operations.

From Tandem Computers to Black Monday

Tandem Computers: The Early Years (1974-1981)

In 1980, Jim Gray moved from IBM to Tandem Computers, a firm founded by Jimmy Treybig and other ex-Hewlett Packard employees in 1974 with the goal of providing reliable, high-throughput transaction processing systems for large organizations such as banks and hotels. An early success story of the venture capital firm Kleiner Perkins (along with the biotechnology firm Genentech), Tandem gradually became a serious competitor to IBM’s entrenched transaction-processing implementations with their so-called “NonStop” systems, hardware and software designed from the ground up to remain in operation. The goal was, to use the terminology of the subfield of fault tolerance, a very high *mean time to failure* (MTTF) and a low *mean time to recovery* (MTTR); The initial NonStop design was aimed at such high-availability applications like ATM machines, Point-of-Sale (POS) systems, and credit-card authorization.[239]

Much of our discussion to this point has focused on techniques for handling high volumes of concurrent transactions, but the issue of the reliability of computers themselves is an older and more basic one—one which inspired the concerns of a simultaneous literature on fault tolerance and reliability throughout the late 20th century. Tandem's business plan meant that they had to develop both reliable software *and* reliable hardware. This is to say: for software which permits continuous/high volumes of transactions can go nowhere without a stable hardware platform, and the computers of the late 1950s, by comparison, were notoriously unreliable, providing a mean time to failure of about 12 hours.[240] While the MTTF of hardware components had improved by the late 1970s, the marketing of Tandem systems in that era was focused far more on reliability of both hardware and software as opposed to transaction throughput.

Tandem's strategy was not to develop the most fault-tolerant hardware possible from the ground up; instead, their designs combined "*fail-fast*" hardware with fault-tolerant *software*. The idea was that any error detected in the (redundant) hardware would immediately cause a failure and be quickly detected and handled by other components and/or the software. The developer of Tandem's proprietary "Guardian" operating system, Bob Bartlett, wrote:

> The system design goal is to provide continuous operation in the presence of a single fault. This requires that all single faults be detectable, diagnosable, and repairable online. In addition, the software must allow reintegration of the repaired module into the system (J. Bartlett 1981).

---

[239](Bartlett and Spainhower 2004).

[240](Gray and Siewiorek 1991). A full exposition on the history of hardware reliability is outside the scope of this chapter; for essays on the varied history of fault-tolerant computing see Avizienis, Kopetz, and Laprie (1987).

Tandem went public as an over-the-counter stock in 1977, and their sales doubled from 1978 through 1980. By then the company's sales were over $100 million with over 1500 employees. In 1981 the firm had 460 clients using over 2,500 Tandem processors, ranging from production operations (semiconductors, steel) to service operations (cargo handling, shipping, food distribution) to banking (retail and investment), insurance, and telecommunications.

Jim Gray and Tandem Computers

When Jim Gray joined Tandem in 1980, their primary product was the "Nonstop I", which ran the Guardian operating system and supported a data management system, data access and query applications, and support for COBOL development as well as a proprietary Tandem Application Language (TAL).[241] The techniques necessary for 'ACID' transactions—which Gray and other developed—were, at the time, being applied by Tandem engineers to the implementation of a "Transaction Management Facility" (TMF), which would implement all the methods of concurrency control and recovery (locking, logging, etc.) in one component module.[242] The formalization and conceptual rigor of Gray's research on transactions at IBM in the late 1970s would thus be used to compete against IBM, who many predicted would stay dominant in the market for transaction-processing systems. Previously, users of Tandem systems had been responsible for developing their program as a "process-pair"—the software equivalent of doubling up on hardware components—and would also be responsible for undoing transactions which perhaps only partially completed in the event of a failure. With TMF, the Tandem user could write COBOL programs with commands like BEGIN-TRANSACTION and

---

[241]The data management system was known as ENCOMPASS; the data access and query applications, ENSCRIBE and ENFORM.

[242]Borr (1981).

END-TRANSACTION, with all relevant commands in between, and the software would be able to automatically back out of a given transaction in the case of failure.[243]

However, it was another matter to convince clients that Tandem's purported combination of fail-safe systems and innovative transaction management features was necessarily worth the expense (systems could cost over $1 million, not including the extensive cost of switching platforms for firms with incumbent systems.). The state of affairs of transaction processing in the largest enterprises in the mid-1980s was described in a fascinating paper (Burman 1985) detailing Bank of America's retail banking backend. Composed of many systems (mostly IBM mainframes) maintained separately, but including "at least on system from many other manufacturers—DEC, Four-Phase, General Automation, NCR, Stratus, Tandem...", this (now-uncommon) view of the interior of financial enterprise demonstrates the practical technological heterogeneity of the era.[244] Significantly, their systems in total only had a 99.5% uptime, with required availability for only 18 hours a day; this still translates to a failure rate of about an hour every two weeks.[245] The author also describes the very real (and then, still-developing) isomorphism between the kind of transactions supported by Tandem's systems and financial transactions:

> In the opinion of the author, banking is primarily an inventory control application. In most inventory control applications and in most enterprises, data merely describes the inventory. Updating a stock record to process an order is quite unrelated to the physical status of the order: the order can be dispatched but the data remain static; the data can be updated, but the stock can stay on the shelf... in banking, by contrast, the data actually is the inventory—the two are synonymous. **In increasingly many cases, the DP [data processing] transaction is the financial transaction** [emphasis added] (Burman 1985, 245).

---

[243](Borr 1981).

[244]Later industry articles reveal that the Tandem installation at Bank of America was a cluster of NonStop TXP minicomputers installed in 1983 and used for dial-up home banking.

[245](J. Bartlett et al. 1998).

It was common for those in Tandem's line of business in the 1980s to refer to the goal of achieving 1000 transactions per second, although the only systems remotely close to achieving that at the time were the specialized airline transaction systems.[246] And because of the heterogeneity of existing transaction processing systems, simple comparisons between the functionalities of Tandem vis-a-vis IMS or other products were not easily available. In the early 1980s, Jim Gray, with the help of twenty-two other professional acquaintances and colleagues, wrote a paper which ported benchmarks previously used at IBM into a new set of benchmarks—a "Debit-credit" test of online transaction-processing power (subsequently *DebitCredit*), as well as *Sort* and *Scan* batch tests—which would allow the comparison of different systems on the basis of not just transactions per second, but cost (in thousands of dollars) per transaction per second.[247] The DebitCredit benchmark, involving the simulation of a bank with 1,000 branches, 10,000 tellers, and 10 million accounts randomly adding and withdrawing funds, was derived directly from an actual case from Bank of America handled by Gray at IBM in the early 1970s.[248]

---

[246](Jim Gray and Levine 2005).

[247](DeWitt and Levine 2008); (Serlin 1993).

[248] (Serlin 1993).

	K-INST	I/O	TPS	$K / TPS	PACKETS
Lean and Mean	20	6	400	40	2
Fast	50	4	100	60	2
Good	100	10	50	80	2
Common	300	20	15	150	2
Funny	1,000	20	1	400	8

Table 2: Transactions per second, after Gray et. al.'s "A Measure of Transaction Processing Power" (Anon., 1985). ("TPS" stands for transactions per second; "$K / TPS" is thousands of dollars per transactions-per-second.)

Gray published his proposal anonymously in the April 1st, 1985 issue of *Datamation*, a long-running computing industry trade periodical, as "A Measure of Transaction Processing Power" (Anon. 1985). It immediately drew attention, in part due to a table of the benchmarks applied to existing systems, renamed with pseudonyms from "Lean and Mean" to "Funny" (a system costing $500,000 per transaction per second); see Table 2. The anonymous paper was distinctive in accounting for input/output (and thus mixing performance characteristics of hardware and software), defining the primary metric "transactions per second" as requiring 95% of transactions to have a less than 1 second response time.[249]

[249] While there has been much written on social-scientific understanding of *standards* as an institutional phenomenon, there has yet been little written on the topic of *benchmarks*. If we conceive of *standards* as "a process of constructing uniformities across time and space, through the generation of agreed-upon rules" (Timmermans and Epstein 2010), then the construction of *benchmarks*—as with the term's etymological origins in 19th-century surveying—is to make possible a metric of *comparison* across heterogeneous sites. That is to say, where standards homogenize, benchmarks provide a total order (in the case of surveying, that of elevation from sea level; in the case of DebitCredit, of transactions per second within a given threshold) and are thus are directly compatible with the interests of customers (and sales agents) in a competitive market.

	1985	1987	1989
CUSTOMERS	1,000	1,300	2,000
OUTAGE CUSTOMERS	176	205	164
SYSTEMS	2,400	6,000	9,000
PROCESSORS	7,000	15,000	25,500
DISKS	16,000	46,000	74,000
REPORTED OUTAGES	285	294	438
SYSTEM MTTF	8 years	20 years	21 years

Table 3. Tandem system outages; data from Jim Gray (1990).

Another aspect of Gray's work at Tandem was the empirical analysis of actual system failures, using data generated by Tandem's clients' installations, to quantify the reliability of the systems over time; see Table 3. While such analyses were not often publicized by computer manufacturers, the mean time to failure (see row labeled "System MTTF") of Tandem's machines, taken in total, vastly exceed what had been previously found or claimed in previous decades (cf. Phister (1976)).

Tandem, the NYSE, and Black Monday

By 1987, 28 stock and futures exchanges were running Tandem hardware in some capacity, as well as 21 of the 25 largest US banks. It turns out that SIAC—the back-end processing organization of the New York Stock Exchange and American Stock Exchange—by that time had become the largest Tandem installation in the world. After new trading records had been set in 1982, SIAC had added and/or replaced some of its mainframes, increasingly adopting Tandem minicomputers as part of their overall configuration in a two-year, $10-million upgrade[250]; a SIAC Vice President was quoted as saying that if their system went down, "[t]here is a high probability that the market would have to cease trading", and that SIAC put "an extremely high premium on recoverability and availability."[251] SIAC's transaction processing included both representatives of the old guard of transaction processing—the Market Data System, MDS-II, was a gradually upgraded IBM mainframe originally installed in 1973—alongside the newer Tandem systems, which were also responsible for broadcasting all of the trades. Between Wednesday, October 14th, 1987, and Tuesday, October 20th, the Dow Jones Industrial Average declined by over 30% before rallying slightly; the volume of trade at the New York Stock Exchange during this one week was comparable to the entire volume of trade for all of 1967. The sequence and potential causes of the events during that week has been narrated extensively by official postmortems (Brady Commission (1988), SEC (1988)) as well as by MacKenzie (2004) and MacKenzie (2006). At specific issue was the increasingly common use of "portfolio insurance"—hedging stock portfolios by short-selling relevant indexes on the futures market—and its potential role in the waves of selling, increases in volatility, and decline in liquidity.

For our purposes, what is significant is that a) at the peak of activity on Monday, October 19th and Tuesday, October 20th, the number of transactions at the NYSE was over twice the typical amount for the period (see Table 4); and that b) the SEC postmortem concluded that "[i]t does not appear that there were many delays in the NYSE's systems used for collecting and routing transaction and quotation information, or in the processing and dissemination of this information by SIAC" (SEC 1988, 7–3); and that "market information systems generally

[250](van Kirk 1989); (Desmond 1984).

[251](Desmond 1984).

performed well during the week of October 19[th]. The Division believes that at least some of the disenchantment with the quality of market information may have been due to the difficulties of trading in fast-moving markets, rather than to the performance of information systems" (SEC 1988, 7–7).

Volume of Shares Traded and Number of Transactions for Typical Week			Volume of Shares Traded and Number of Transactions for Week of Market Break		
	Volume of Shares	# Transactions		Volume of Shares	# Transactions
Monday 10/5	159,707,100	80,792	Monday 10/19	604,330,410	202,084
Tuesday 10/6	175,679,240	92,983	Tuesday 10/20	608,148,710	205,868
Wednesday 10/7	186,348,370	96,529	Wednesday 10/21	449,350,330	190,749
Thursday 10/8	198,701,120	87,920	Thursday 10/22	392,234,180	146,172
Friday 10/9	158,310,910	77,614	Friday 10/23	245,520,770	99,544

Table 4. NYSE Trading Statistics for week of 10/5/1987 and week of 10/19/1987

Conclusion

The kinds of 'transactions' described here are, as stated in the discussion on relational contracts, specifically those which are most ephemeral. However, it is important to recognize that such ephemeral transactions correspond to an enormous proportion of financial activity; and whatever benefits the social interaction on the New York Stock Exchange floor may have had, they were made distinctly less relevant by the many alternative electronic trading venues for securities whose development I will discuss in Chapter 5. The focus of benchmarks on "transaction per second" reflect this interest, as such a parameter has no overt upper bound in certain environments like the NYSE. The industries which originally best represented this limitless interest in sustained concurrent modification of some centralized database are those which make markets for continuous-auction exchange in virtualized securities.

But in 21st-century computing, the most significant technological development is the increasingly widespread use of scalable distributed systems for which the size and scope of data storage and/or potential online interaction for any given organization is technically limited only by one's budget. This chapter describes an important intellectual and pragmatic transition of the previous century—simultaneously of technology and technique—by which large organizations came to be able to fundamentally *rely* on digital systems. I have argued that this development was driven by the interests in and demand for support for high volumes of networked transactions by banking and financial institutions. These on-line transaction processing systems (e.g., client-server databases with "ACID" support) were in turn a firm precondition for subsequent developments which facilitate perduring, digitally mediated interactions at a scale not previously known to human organization. What had begun as an institutional dependence on overnight "batch" processing gave way to the possibility for the continuous operations and high availability necessary for any contemporary global enterprise; and to the interposition of scalable mediation far beyond industrial commerce and finance, and towards the "transactionalization"—and thus, by default, the marketization—of everyday life.

Coda: The Transaction vs. The Message

This chapter has focused primarily on the concept of *transaction,* its formalization, and the impact of its commercialization. But it is my position that *transaction processing* or *transaction management*—the centralization and concurrent handling of large volumes of small database updates—is in some way in a relationship of *duality* with (heretofore less acknowledged) *message switching* and *message delivery*: i.e., the routing and handling of large flows of small communiques along paths in a network. While each can facilitate the other, they can be thought of, philosophically and semiotically, as distinct "ideal types" of operations and techniques. In the next chapter, I will demonstrate this distinction through elaborating the history of message-switching technologies, which are related to— but are never quite fully embodied by—centralized transaction-processing systems. For while the facility for handling concurrent transactions clearly also involves the successful management of a flow of messages containing data about individual transactions, the transaction *concept* is more concerned with the ACID constraints—the atomicity, consistency, isolation, and durability of an executing transaction—and not at all with the distribution of those transactions to decentralized locations. This, in turn, is why the problem of *distributed databases* comes to exist in the first place; transaction processing assumed at the outset a centralized database, without an accompanying theory of how to maintain the ACID constraints when one's database was distributed across space (and therefore also time).

With an extreme centralization of transactions—as is seen on the systems of the major securities exchanges and their member firms—comes the necessity of reporting vast flows of events (in the form of messages), which travel at their own pace to various locations. The essential difference is that each *transaction* represents a computable change of state of a centralized set of relations (e.g. database tables); each *message* is in itself ephemeral (i.e., it does not directly represent a change of state) and "needs" only to successfully travel from a source to a destination along a network (potentially in a particular order). A message may *contain* a transaction, but the message-handing systems are agnostic to the meaning of their content; but a transaction cannot *contain* a message, it can only *trigger* the production of messages. This suggests a certain *duality of transaction and message*: messages *contain* spatially (some quantity of symbolic data) but *flow* temporally (are processed one at a time); while transactions might be said to *flow* spatially (transforming the contents of a database one at a time) and *contain* temporally (executing in effective isolation).

It is worth noting that Bjork and Davies originally described their concept of transaction in the context of a so-called “DB/DC system”, an IBM term logically segregating the “Data Base” (DB) from the “Data Communications” (DC). We have discussed at length the “DB” (Data Base) part of their worldview in our discussions of the database-model “great debate” of Chapter 2 and the formalization of database transaction in this chapter; now we will move on to the “DC” (Data Communication) aspect of reliability. This will take us away from the transactions which represent the unit transformations of finite data, to the messages which represent the unit movement of (potentially infinite) *codata*.

CHAPTER 4

BROKERS, QUEUES, AND FLOWS: TECHNIQUES OF FINANCIALIZATION AND CONSOLIDATION

Introduction: From Transaction to Message

In the aftermath of the so-called Black Monday "market break" of October 1987 on Wall Street—in which major global stock markets lost between 20 to 50% of their value by the end of the month— it was noted that while there were delays in routing and execution of trades on some automated systems and regional exchanges, overall "the market information systems were not subject to any major breakdowns or delays".[252] In a previous chapter, I described how the formalization of the *transaction* in the 1970s and early 1980s—and its subsequent commercialization by firms producing and marketing high-reliability OLTP (On-Line Transaction Processing) systems—paved the way for banks and stock exchanges to rely on 24/7 computer systems focused on high volumes of "transactions per second" (*tps*). This concept of the transaction focused on representing a flux of updates to a database happening *as if* they were 'atomic' (all-or-nothing) and 'isolated' (one-at-a-time), as well as 'durable' (effects of completed transactions can be recovered in the event of failure at some level of the system).

One major vendor of transaction-processing systems, Tandem Computers, had as one of its largest clients the Securities Industry Automation Council (SIAC), which operated the order matching and clearing/settlement systems for the New York Stock Exchange and NASDAQ. While the Tandem installation in the SIAC back offices during the 1987 crash was the largest such Tandem installation in the world, it turns out that it was *not*, in fact, a Tandem system executing the matched transactions; in October 1987, the primary Market Data System was still

---

[252] (SEC 1988, xxi–xxii).

running on an IBM 4341, a system dating in part to the mid-1970s.[253] What, then, were all of those Tandem machines doing, if not centralizing the execution of transactions? The answer is that they running the so-called *Consolidated Trade* (CT)[254] and *Consolidated Quote* (CQ) reporting systems, which were the contemporary analogues to the stock ticker; the first system consolidated, recorded and broadcast reported trades (wherever they occurred) and the second consolidated, recorded and broadcast reported quotes for bids and offers.[255]

While I have previously primarily discussed transaction processing as systems for bringing together volumes of transactions and processing them *as if* they occurred one at a time in isolation, I have heretofore largely ignored the *streams* of communicative *events* in which each transaction is communicated before being centralized, processed, and stored (as if) one at a time. For the reliability of, e.g., Tandem's systems were useful not just in the processing and logging of many simultaneous transactions, but in their facility for reliably handling *messages* (sometimes interpreted as *events*), and their receiving, queuing, handling, and sending. In this major section we will move away from the 'ACID' transaction[256]—which (in its atomicity and isolation) occurs as an abstract, frozen moment in time, a conceptual (if not actual) infinitesimal delta in which a database transforms from one state to another—and to its material (and

---

[253] (Saunders 2004). The entire setup, described in (Desmond 1984), included 120 Tandem computers, an IBM 3083 and 3031 mainframe, along with "two IBM 370/158s and a 370/148; two IBM 3033s; one IBM 4341; four Sperry Corp. mainframes; two 1100/82s and two 1100/60s; and nearly 40 Digital Equipment Corp. PDP-11/05, 11/45 and 11/70 minis."

[254] Often, and confusingly, also called the *Consolidated Tape*, in reference to a ticker-tape-oriented past which I will discuss below.

[255] For subsequent (late-1990s) descriptions of the CTS (Consolidated Tape System), CQS (Consolidated Quote System), and OPRA (Options Price Reporting Authority), see Bach (1998).

[256] The 'ACID' acronym, recall, stands for the properties of atomicity, consistency, isolation, and durability and encapsulates the formal properties for supporting transactions in a database system (Haerder and Reuter 1983).

conceptual) *dual*: flows of messages, passing through time, maintained in some order (where possible), from one place to another.[257]

Another way of focusing on this 'data-in-motion' is to note, succinctly, that *time matters* (Abbott 2001). What does the history of data (and data communication) techniques look like when we pledge not to ignore this fact? In what follows, I will highlight the conceptual lineage of a distinctive technique—known as a "publish/subscribe" (aka *pub/sub*) or *message broker* system—which facilitates certain types of communication which would be initially fundamental in financial services (e.g., brokerages and exchanges) and later for the integration of backend systems for other large organizations, including the facilitation of electronic commerce on the web. Ultimately, this paradigm would become fundamental to many large internet-connected organizations—especially large-scale social media and communication services, which provide a dense mix of traditional isolated database updates and flurries of real-time notifications of events, whether they be user-facing or for internal analytics purposes. In the next section I will

---

[257] To make this claim is to argue for a kind of 'dialectical epistemology' in the history of computing. Databases, semiotically revolutionized through the adoption of Codd's set-theoretic relational model (Codd 1970), needed the concept of the transaction and the reliability it induced to sustain those (otherwise logically atemporal) database systems' claim to correspondence with organizational reality. The messages and events that are the focus of this chapter are the implicitly missing 'antithesis' (or more accurately, 'dual') of this atemporal perspective; to understand empirically-observable computing-in-practice, we must synthesize the views from these two intellectual basins. This observation has analogies elsewhere; the mathematician A.R.D. Mathias, in response to Saunders Mac Lane's attempt to revise the foundations of mathematics through category theory in (Mac Lane 1986), wrote:

> ...I have found myself coming to a view that can certainly be traced back to Plato, namely that there are *two* primitive mathematical intuitions; which might be called the geometrical and the arithmetical; or, alternatively the spatial and the temporal… Let me refer to my contention that there are these two modes, neither reducible to the other, as positing an essential *bimodality* of mathematical thought (A.R.D. Mathias 1992).

Contra Hegelian dialectic, I see little evidence of an imminent unifying and revolutionary synthesis in which the temporality and presentism of data-in-motion can be reconciled and wholly unified with the static view of data-as-archive (although engineers at Google have recently made some claims in this direction (Akidau et al. 2015)). It would seem that this duality is—if not a constant—at least a frequent and recurring source of technical conflict and misrecognition in many fields. Finally, my intuition—Bachelardian in the sense of his *Formation of the Scientific Mind* (Bachelard 2001) but lacking empirical evidence here—is that the roots of this continuing opposition might be found in the conventions of Western primary and secondary schooling in science.

isolate the specific distinctive features that conceptually distinguish this communicative system from others.

A Typology of Messaging

The Western concept of the 'message' as the *spatial* carrying of communication has its origins in Greek gods like Hermes (and his Roman analogue Mercury), who played the role of Zeus' courier—and, in the Bible, angels (ἄγγελοι or *angeloi*)—who were *messengers* between gods, or between gods and men.[258] The image of Hermes/Mercury consistently appeared in the imagery of various newspapers and early Western post offices, from Danish periodicals (e.g. the *Altonaischer Mercurius* in 1698) to the seals (and later, stamps) of the U.S. Post Office from 1782 (with the 'Mercury' being a common title for U.S. newspapers today).[259]

Analogously, the practice of *messaging,* in its various contemporary manifestations, is a consistently distinctive form of communication in its emphasis on the qualities of the discrete, the symbolic, and the *simplex* (or *uniplex* — each message travels in one direction at a time). A canonical example would be that of a *postal* system in which sealed messages can be delivered over long distances and relayed across road networks. This is in contrast to, for example, a face-to-face, in-person conversation, where communication is continuous, immediately indexical (i.e., contiguous in physical/material reality) as well as symbolic (i.e. the utterance of lexical units),

---

[258] On the concept of messengers as an explicit part of a philosophy of communication, see Akidau et al. (2015). Other interesting commentary with respect to angels and messages (and computer systems!) can be found in (Serres 1995).

[259] (DeBlois, Harris, and Pedersen 2012).

and, as telecommunications engineers would put it, "full duplex" (both participants can communicate at the same time, although turn-taking is a universal pattern).[260]

Already, this is a somewhat finer contrast than in some contemporary sociological typologies of communication techniques, which often merely distinguish between "point-to-point" communications and "broadcast" communications (DiMaggio et al. 2001); this particular distinction, in my typology, is only one distinguishing feature among many.[261] Because part of my argument is to more accurately distinguish particular *message-centric* systems from other forms of communication, in Table 5 I have enumerated a variety of communication types and technologies to show in what dimensions they differ from each other. In this chapter, I will focus on three such dimensions, which act as *distinctive features*—in the sense of (Jakobson, Fant, and Halle 1961)—which, when combined, will ultimately isolate the particular modality of modern-day *message broker* systems:

- *Transmission: Synchronous vs. Asynchronous.* These terms generally refer to whether or not the sending and receiving participants of a given communicative act send and receive a message "at the same time" (e.g., as takes place during a telephone conversation) or whether the act of sending and receiving are more

[260] (Sacks, Schegloff, and Jefferson 1974). The 'asymmetrical' nature of face-to-face interaction is discussed in, e.g., (Goffman 1953, 81). I use the term 'indexical' here in the sense of Silverstein, who distinctly summarizes: "Indexes are those signs where the occurrence of a sign vehicle token bears a connection of understood spatio-temporal contiguity to the occurrence of the entity signaled" (Silverstein 1976).

[261] This chapter will be in part focused on a type of communication denoted 'multicast' or 'group communication', in which a sender targets multiple, but not all, receivers in a given network. I would argue that just as Paul Lazarsfeld and the Bureau of Radio Research focused on the social implications of broadcast communication, the current studies of Internet-related social phenomena could benefit from distinguishing between its worlds of unicast communication (most instant messaging, the early web), multicast (most social media communications platforms, as well as the message brokers whose history is described in this chapter), and total broadcast (no longer really present due to the internet's architecture).

temporally segregated (as in “leaving a message” on an answering machine to be heard later).[262]

There are in turn two types of synchrony/asynchrony; at the *contact* level and the *message* level (where the terms *contact* and *message* are used in the sense of (Jakobson 1960)). *Contact asynchrony* occurs when, for example, a communication arrives as a *surprise* or *interruption* (a telephone rings; a telegraph sounder buzzes; an email notification appears); whereas *contact synchrony* is when the sender and receiver are more-or-less permanently locked in a request-reply situation (a model for this might be the ‘bisync’ data communications of 1960s IBM terminals, in which the terminal and mainframe maintain a back-and-forth dialogue on a leased communications line; one is always ‘blocked’ waiting for the other to send or receive).[263] For linguists like Jakobson, the *contact* quality of communication is associated with the *phatic* function of communication—i.e., the significance (or lack thereof) of the presence (or co-presence) of addresser and addressee.

---

[262] There are some subtleties here; *switched* and/or *routed* messaging indeed indexically detaches sender from receiver (so that, i.e., low-priority telegrams, which might be manually decoded and re-encoded multiple times at multiple telegraph offices, could arrive with some delay of hours or more), but sufficiently *fast* switched and/or routed messaging (e.g. the data communications of today’s high-speed Internet) can provide a phenomenological *illusion* of continuous synchrony (i.e. of a computer application uploading data “at the same time” as a corresponding remote downloading application), despite that ‘continuity’ being ontologically composed of many, many switched and/or routed messages (i.e., TCP/IP packets). Moreover, if one accepts that special relativity eliminates the possibility of “true” simultaneity (even/especially for speed-of-light telecommunication), the phrase “at the same time” can itself be problematized. I argue that it is conceptually useful, then, to instead consider ‘synchronous’ communication as corresponding to a state of *waiting* on behalf of the sender/receiver: a situation in which the sender is always waiting for something to be fully received, and the receiver is always waiting for something to be fully sent.

[263] Another, more common form of contact synchrony is *isochrony*, in which communication is consistently “clocked” by pulses separated by an equal interval of time, produced by a clock generator using, e.g., a crystal oscillator.

Message asynchrony corresponds to *relayed* communication—as in a "relay race", but also in the sense of electrical *relays* (devices that, when activated, make or break a connection from one circuit to another)—in which the act of sending or receiving information is detached in time (as in 'store-and-forward' telegraphic communication, discussed below; or an answering machine where a message is 'left' for later retrieval). And *message synchrony*, again, refers to our intuitive sense of a phone conversation, where speech is heard near-instantaneously by the other party. It is this latter opposition which is of somewhat more importance, and so when I say 'synchrony' or 'asynchrony' alone, I will by default mean message synchrony and message asynchrony.[264] (The fact that I have had to explicitly distinguish between contact synchrony/asynchrony and message synchrony/asynchrony does indicate that these oppositions are sometimes conflated in the primary literature.)

- *Unicast/Multicast/Broadcast*. This has to do with the number of *addressees* of the communication in the network; one-to-one communication is unicast, one-to-many is multicast, and one-to-all is broadcast.

- *Finite/Potentially-Infinite Communications* (i.e. *data/codata*).[265] Is the communication *finite*, in the sense that we can expect it to have a maximum

[264] By contrast, Jakobson associates the message with a purely information-theoretic sequence of bits, devoid of contact, context, and code; but for him the message is at the core of what he calls the *poetic* function.

[265] One may ask, especially, why such a distinction between data and 'codata' has scarcely emerged in the voluminous contemporary writing by social scientists and humanists about the consequences of contemporary 'big data' (cf. (Kitchin 2014)). I would answer by appealing to those Polynesian societies of (Lévi-Strauss 1962, 2), who have no names for the plants that they do not use or which are perceived to have no use; the digital humanities and even the newer computational social sciences rarely perform their analyses in an on-line, 'streaming' mode. As I

length and time of delivery, or is it more-or-less ‘potentially infinite’, a Heraclitean ‘stream’ of information? More specifically, does the receiver *not know in advance how much information they are going to get?* I use the term ***codata*** to describe communication which arrives more in the form of an unpredictable flow than not.[266] It is not necessarily important whether this flow is ‘continuous’ or ‘discontinuous’, only that the stream of information is unpredictable in volume and unfurling in time, as opposed to having a *known* size and an *inert* manifestation (in the case of a computer, being held on a disk and/or in memory).

---

will discuss later, most of what the sciences call ‘data analysis’ is performed on finite, static, ‘cleaned’ and thus typically tabularly-organized information; it does not deal with unpredictable volumes of data-in-motion.

[266] The term *codata* comes from research in programming languages such as (Hagino 1987b) and while that community’s definition of ‘codata’ is more formalized, the meaning—that of a kind of data which arrives in unpredictable volumes as opposed to having a known size—is, I argue, comparable. The prefix ‘co-’ derives from category theory (Mac Lane 1986), where it denotes the mathematical dual of a given category (a kind of mathematical abstraction, composed of a collection of static ‘objects’ and transitory ‘arrows’, the latter of which specify the possibility of transduction from one object in the category to another).

Communication Mode	*Communicative Unit*	*Tele-communications Type*	*Communicative Unit: Symbolic vs. Indexical*	*Communicative Network: Direct vs. Relayed/ Switched or Routed*
TELEGRAPHY (DEDICATED LINE)	Telegram	Signal	Symbolic (Morse code)	Direct (permanent cables)
TELEGRAPHY (MESSAGE-SWITCHED)	Telegram	Signal	Symbolic (Morse code)	Relayed along dedicated lines (message-switched)
STOCK TICKER	Listed firm + sale/quote price	Signal	Symbolic	Direct/ "Loop"
TELEPHONY (LINE-SWITCHED)	Voice conversation	Signal	Indexical (human speech)	Direct/Line-switched (cable switching pre 1970s; post, electronic switching (ESS))
RADIO	Radio show or, e.g. radio advertisement	Signal	Indexical (speech and music)	Direct
ETHERNET	Ethernet *frame*	Packet	Symbolic	Direct (contiguous LAN)
INTERNET PROTOCOL (IP)	IP *datagram*	Packet	Symbolic	Packet-routed
TCP (RELIABLE TRANSMISSION OF IP)	TCP *protocol data unit*	Packet	Symbolic	Packet-routed
RPC	Remote Procedure Call request/reply	Message	Symbolic	TCP
MESSAGE QUEUE	Message	Message	Symbolic	UDP or TCP
MESSAGE BROKER	Message	Message	Symbolic	UDP or TCP
SMTP (EMAIL)	Message	Message	Symbolic	TCP

Table 5. Typology of telecommunications modes across various distinctive features.

Communication Mode	*Contact synchrony/ asynchrony*	*Message synchrony/ asynchrony*	*Unicast/ Multicast/ Broadcast*	*Finite /Potentially infinite communications (data vs. codata)*
TELEGRAPHY (DEDICATED LINE)	Asynchronous	Synchronous	Broadcast	Finite
TELEGRAPHY (MESSAGE-SWITCHED)	Asynchronous	**Asynchronous**	Unicast (via Broadcast relays)	Finite
STOCK TICKER	Asynchronous	Synchronous[a]	**Multicast**	**Potentially infinite**
TELEPHONY (LINE-SWITCHED)	Asynchronous	Synchronous	Unicast (one-to-one); Multicast ("party line" shared service)	Finite (channel) (each phone call has its setup and teardown)
RADIO	Synchronous (always transmitting)	Synchronous	Broadcast	Finite (each radio show is expected to end at some point)
ETHERNET	Asynchronous	Synchronous[b]	Broadcast	Finite (Fixed-sized frame)
INTERNET PROTOCOL (IP)	Asynchronous	**Asynchronous**	Unicast	Finite (fixed-size datagram)
TCP (RELIABLE TRANSMISSION OF IP)	Asynchronous	**Asynchronous**	Unicast	Finite (though each TCP protocol unit can be of varying size)
RPC	Synchronous	Synchronous	Unicast	Finite
MESSAGE QUEUE	Asynchronous	**Asynchronous**	Unicast	**Potentially infinite**
MESSAGE BROKER	Asynchronous	**Asynchronous**	**Multicast**	**Potentially infinite**
SMTP (EMAIL)	Asynchronous	**Asynchronous**	**Multicast** (one sender can have multiple receivers via CC)	Finite (email has bounded size)

[a] As discussed above, this notion of simultaneity is a simplification from the special-relativistic perspective; a better way to think about it is whether the sender and receiving actors *actively wait* for a message to be received/sent.
[b] It is not uncommon for information technology practitioners to refer to Ethernet as 'frame-asynchronous and bit-synchronous'; i.e. contact-asynchronous and message-synchronous.

Table 5 (con't). Typology of telecommunications modes across various distinctive features.

The Ticker, the Philosophy of Time, and Social Theory

The *stock ticker* was a device introduced in the early 1860s by Edward A. Calahan, an employee at the American Telegraph Company, and installed on the floor at the NYSE and a handful of brokerage offices in 1867. Calahan combined his novel ‘printing telegraph’—the name given to all receiving devices which could automatically convert telegraphic communications to materialized symbolic text—with the logic of an earlier system set up by Samuel Spahr Laws of the nearby Gold Exchange, in which an operator on the trading floor could transmit gold quotations via a specialized keyboard to real-time “gold indicators” (instruments with dials) via a wire circuit.[267] The resultant system of Calahan, dubbed “the ticker”, was “both network and machine”.[268] Electrically linked to a keyboard near the trading floor, each ticker device would print out on a continuous roll of thin tape the name of a security and its corresponding price quote.[269] In recent years, the ticker’s materiality has been isolated as of theoretical interest for economic sociology in a short series of articles by Alex Preda and Karin Knorr Cetina, and in this section, I will attempt to show how the ticker’s distinctive *multicast, synchronous*, and *potentially-infinite* qualities can help explain their specific epistemological interest in the device. First, I will distinguish the stock ticker’s ontology/phenomenology from contemporaneous telecommunication circuits/devices like the telegraph; then I will summarize Preda and Knorr Cetina’s work in this new context; and then I will discuss how other philosophers, including Bergson, Bachelard, and Mead, can provide a

[267] (Hochfelder 2012).

[268] (Moore 2016).

[269] (Preda 2006). By the 1870s, stock tickers could also print the traded volume of the security (i.e. the number of shares traded for a given transaction).

deeper understanding of how some of the ticker's qualities phenomenologically influence the message-broker systems which we are working towards.

The Ticker in Relation to The Telegraph

In the decades preceding the introduction of the stock ticker on Wall Street, the use of *telegraphy* had expanded rapidly, with Europe and North America being connected by a transatlantic cable just a year before in 1866 (transatlantic attempts began in 1858 but failed shortly thereafter or were not reliable). The innovative communication scheme devised by Morse was nothing less (or more) than a mapping of the English alphabet to a variable-length on-off serial code which was effected by the operation of a *key* (a manually operated electrical switch), which by opening and closing a lengthy copper wire circuit, would trigger a *sounder* at one or more distant stations. This type of *modulation* (or technique for varying an indexical connection to communicate information[270]) of the first electromagnetic telegraphs is known as *on-off keying*.[271]

The close connection between stock exchange trading of the 19th century and the early expansion of telegraphy (and other expedient relays, including pigeon post[272]) is well-

---

[270] The term *modulation* was commonly used for centuries in the context of music and, later, acoustics, and became more commonly used in this sense in the radio era, e.g. *amplitude modulation* (AM) and *frequency modulation* (FM).

[271] Morse code in practice was, in fact, more than just a serial encoding of the English alphabet; it also was in part a serial-symbolic transduction of *pragmatic* communication, regarding other aspects of the communicative situation besides the pure message. Examples include 'Invitation to transmit' (i.e. , "go ahead") ('– - –'), notification of an error ('- - - - - - - - -'), end-of-message ('- – – – -') (Dodge 1921, 9). (That is to say, these example patterns are intended to be interpreted as indexical, instead of being interpreted as symbol characters to be transcribed. In addition, the *call sign* of a telegraph office, when transmitted, is simultaneously an indexical sign (the sounder being electrically affected by the distant opening of the key) and a symbolic sign, interpreted indexically to refer to the addressee of the message.

[272] The first Baron Rothschild is said to have used an elaborate homing-pigeon relay from Paris to London, including the intermediary points of Dover and Calais, which are still the preserve of advanced high-frequency trading (William Bernhard Tegetmeier 1871).

documented: one important early telegraph line (which initially competed with an existing optical semaphore relay) was between the Sandy Hook lighthouse in New Jersey and the Merchants Exchange on Wall Street (a progenitor of the New York Stock Exchange); and the first link on the east coast of the United States, in 1846, connected Philadelphia's Merchants' Exchange (predecessor of the Philadelphia Stock Exchange) to traders on Wall Street. And almost immediately, newspapers began criticizing 'speculators' who used the technology for 'inside' trading (Du Boff 1980). Telegraphy was also an important factor (along with railroads and the storage facilities of Cronon (1992)) in the development of commodity exchanges, by introducing so-called "to arrive" contracts which specified the date of delivery (A. D. Chandler 1977, 211).[273] Telegraphy was also widely used for longer-distance coordination by expanding businesses; private individual communication, by comparison, represented a minority of telegraph use.[274]

As the networks of telegraph offices expanded, telegrams could be *relayed* from one of a 'spoke' of local telegraph offices to each other or to a central telegraph office, and in turn to another locale's office, at which point the message would be transcribed on paper and taken on foot by a messenger to the recipient.[275] By contrast, the information on an early stock ticker was be transmitted in a relatively circumscribed geographical region from a manually operated keyboard on the trading floor linked to a so-called 'loop' circuit with multiple 'listening' ticker devices. Reporters in the "Board-Room" would transmit quotes by telegraph to two ticker firms:

---

[273] As pointed out by (A. D. Chandler 1977, 195), the railroad and telegraph were in a symbiotic relationship, with the railroad providing the right-of-way for the telegraph and the telegraph providing an efficient way to coordinate railroad traffic.

[274] (DuBoff 1982); (Tarr, Finholt, and Goodman 1987).

[275] (Standage 1998).

one, the New York Quotation Company, would relay quotes by ticker to over 1,100 member brokerages; the other, the Gold and Stock, would distribute them to clients outside the exchange—including the "bucket shops", locations where unregulated side betting would occur (Hochfelder 2006).

The method of relaying telegrams between offices in a larger network became consistently known as a *store-and-forward* system; it thus produced what, in my typology, I called *message asynchrony* — the indexical detachment of the act of sending and receiving of information. The ticker was, with respect to message synchrony vs. message asynchrony, more like a telephone call, which indexically links sender and receiver in a single continuous circuit. While I will not discuss early telephone systems here for reasons of space, it should be noted that the kind of *circuit switching* (or *line switching*) used to indexically connect telephone users (as in the use of telephone switchboards and their operators) produces a message synchrony which can be overtly contrasted with the asynchronous *message switching* of telegraph offices. Significantly, the density of technical development and innovation in telephone line switching in the early 20th century was, in part, due to the fact that the number of possible (indexical) one-to-one connections increases approximately as the square of the total number of subscribers in a network (Mueller 1989); so while the store-and-forward telegrams of Western Union's network could be given various levels of priority and delivery time in high-traffic situations, the live telephone calls of the Bell System necessitated a comparatively lengthy monopolization of wires from the sender to the receiver.

The Economic Sociology and Philosophy of the Ticker

Some of the distinctive properties of the ticker device, observed from an economic-sociological and philosophical perspective, were described by the sociologists Alex Preda and

Karin Knorr Cetina in the mid-2000s. This work began with field research by Knorr Cetina and Urs Bruegger on the trading floors of three investment banks in Zurich, where they focused on currency traders, their "face-to-screen" orientations (Cetina and Bruegger 2002b), and the "postsocial" qualities of the traders' relationships to markets as presented on trading screens (Cetina and Bruegger 2002c).

Knorr Cetina and Bruegger's approach to these markets, focusing on a "reflexive, temporal form of coordination" (Cetina and Bruegger 2002a, 932) was explicitly conceived as complementary to studies of *social networks* which they deemed insufficient for explaining the relationships of market participants in an elaborate communicative screen-space. The next year, Knorr Cetina published another essay which emphasized currency traders' world as a *flow architecture*, drawing on Heraclitean metaphor to describe market reality (Cetina 2003b); but this to was opposed not to an material ideology of stasis but to an increasingly glamorous sociological theory of networks. Knorr Cetina's opposition between network and flow is interesting for the purposes of this chapter because the story of message brokers (and the history of distributed computing systems more generally), as we shall see, is one which must quite overtly *bring together* network and flow. It is less that Knorr Cetina is wrong to oppose them, and more that the concept of 'network' to which she addresses her critique is a static one devised by social scientists. It is not that the reflexivity and performativity which she finds in her global markets do not exist in real-world social networks; it is that reflexivity and performativity were not part of social scientists' network models (at the time, and to some extent still today, consisting merely of 'nodes' and 'edges', and eschewing indexicality and processual change).

Preda, instead of examining the high-tech world of modern-day trading, took to the 19^{th} century and found himself studying something oddly relevant (both historically and ontologically) to Knorr Cetina's 'flow architecture'. In his examination of the history of the stock ticker (via archives in New York, Philadelphia, and London), Preda proposed to see the ticker technology as what he called a '*standardizer'* (which makes traces of trading activity in a standardized textual, printed format) and also as a '*generator'* of an unpredictable flow of values that moves faster or slower along with the trading activity of the moment. Preda also argued that "technology is social action" and thus (citing 20th-century Viennese sociologist/philosopher Alfred Schutz) "generates time structures". He drew two relevant distinctions in this short passage:

> For instance, a technology that produces data sporadically and at irregular intervals differs from one producing data continuously and at regular intervals. Data perceived as representing past transactions differ from data representing current transactions (Preda 2006, 757).

This first distinction (in the first sentence), I argue, closely corresponds to my concept of (contact) *asynchrony* vs. (contact) *synchrony*; and the second distinction (in the second sentence), to *data* vs. *codata*—i.e., between transactions-as-stored-record vs. transactions-in-the-moment. Preda is here drawing from Schutz' distinction between *performed action*—"action as performed act, as the thing done" and *working action*—"action as an ongoing process" (Schutz 1962, 214)—which can be used to distinguish between "arrangements such as a table or list that refers to the past" (i.e. data) vs. the supposedly volitional aspect of "data presented as a continuous flow" (i.e. codata). (Preda 2006, 757) The stock ticker is thus an intriguing device in its mix of contact asynchrony, message synchrony, and codata: it is an unpredictable flow of discrete interruptions, which nevertheless is closely temporally linked to the site of transaction. It was also only possible to the extent that the circuit could be spatially extended; in Paris, by contrast, brokerage houses were scattered all over the city, and therefore we should not be surprised that the stock ticker did not catch on with remotely the same virulence as in America.[276]

Preda shows that in the early decades of the ticker, brokerage firms still used postal letters to communicate to clients, and convincingly argues that the ticker was initially *"not* wanted for efficient, accurate and broad diffusion of price data" but instead was desired "because it helped reinforce social status and a monopoly over authoritative price data" (Preda 2006, 765). This was in part due to the separation between the higher-status Regular Board, which traded in periodic *call auctions,* and the Open Board, which traded *continuously*. But in November 1870, the Regular Board merged with the Open Board, trading in the same room; and the difficulty of call-auction markets to coexist with competing continuous markets would recur as a theme in the history of financial exchanges in the 21st century.[277]

Bachelard, Mead, and Multicast Codata

What social theories, then, are necessary to bridge the 19th-century synchronous world of Preda's ticker with the 21st-century infrastructure of Knorr Cetina's flow architectures, in which traders differentially subscribed to a massive variety of streams of market and news events? In this subsection I will explain (1) how the philosopher Gaston Bachelard problematized Henri Bergson's critique of special relativity by proposing a duality of duration and event, and thus provided a philosophical basis to the oppositional qualities of later distributed computing

---

[276] "[The Americans] are amazed to think how it can be possible that immense speculations are carried on in Paris without a 'ticker', though such is the case. Some years ago an attempt was made to introduce the [ticker] system [in Paris], but the electricians in charge were inefficient, and the service was so bad that it was finally abandoned. The offices of the *Agents de Change* and *Coulissiers* are scattered throughout the city, and messengers and telephones are the media through which fluctuations are made known" (Gibson 1889, 84).

[277] Specifically, the Arizona Stock Exchange of Steven Wunsch (Muniesa 2011) would run into difficulties with regulators in its role as both an electronic exchange and its use of a call auction instead of continuous trading; and in the 21st century, the IEX exchange would use fast-paced call auctions to compete against continuous exchanges (Lewis 2015).

paradigms; and (2) how the philosophical work of the sociologist George H. Mead can help us understand our other categories — those of *multicast codata* (i.e. digital communications with multiple recipients, unpredictably flowing in the moment) — as a specifically and intrinsically *social* phenomena in a way that conceptually unicast and/or static data communications/formations are not.

Technoscience in North America and Europe in the late 19th century was immersed in the problem of *clock synchronization*, driven in part by the expansion of railway networks (P. Galison 2003, 40). Various American and European inventors from the 1830s onwards responded to the challenge of synchronizing clocks by devising systems (with varying degrees of success), typically of a 'master-slave' orientation, which would coordinate a *primary* clock with *secondary* clocks via electromagnetic signals. The intensity and diversity of these projects, from office buildings to military communications, certainly indicates a demand for temporal consensus in many aspects of bureaucratic conquest; synchronized clocks, from this perspective, represent a most literal version of Latour's 'immutable mobiles', those various forms of inscription apparatus (including clocks, but also maps and records) which facilitate administration at a distance.[278] To use my terminology, such systems intended not necessarily to send arbitrary *messages* between a master clock and slave clocks but to merely preserve a reliably periodic *contact synchrony* (or *isochrony*), so that each could tick in relative unison.[279]

---

[278] "Commercial interests, capitalist spirit, imperialism, thirst for knowledge, are empty terms as long as one does not take into account Mercator's projection, marine clocks and their markers, copper engraving of maps, rutters, the keeping of "log books", and the many printed editions of Cook's voyages that La Pérouse carries with him" (Latour 1986).

[279] *Isochrony* can be understood as a more periodic form of contact synchrony, in which a back-and-forth communication structure occurs with messages of similar length.

The problem of clock synchronization eventually led directly to Einstein's proposal of special relativity (Einstein 1905), which in a few short years after its publication was widely seen as an exemplar of modern intellectual thought, one which should be accounted for not just in physics but in philosophy and elsewhere (P. Galison 2003, 24–25). Einstein's 1905 proposal—that, as a consequence of the upper limit to the speed of light *c*, that time was only meaningful with respect to a reference frame—was prefigured by Poincaré, who in his 1898 essay "The Measure of Time", noted the seeming arbitrariness of measuring lengths of time and/or determining simultaneity for distant events; this was in opposition to the then-popular philosophy of Henri Bergson, for whom the true conception of duration and simultaneity was something *intuitive* as opposed to something that could be formalized geometrically.

The work of Poincaré and Einstein would lead to the concept of *space-time* as promulgated by Minkowski, who stated that, in the wake of Einstein, "henceforth space by itself and time by itself, are doomed to fade away into mere shadows, and only a kind of union of the two will preserve independence" (Minkowski 2012). In Minkowski's famous diagrams (P. L. Galison 1979), the paths of objects moving at different speeds could be plotted on the same two-dimensional graph—with distance on one axis and time on the other—with a (linear) skewing transformation depending on each object's relative speed (and thus demonstrating how 'simultaneity' is in the eye of the beholder).[280] This conceptual approach was extremely problematic for Bergson, who had long resisted the 'spatialization' (and thus quantification) of time in all of its forms. As Bergson describes in his first work (1899's *Essai sur les données immédiates de la conscience*, retitled in English as *Time and Free Will*), consciousness should be

---

[280] Minkowski's spacetime was 4-dimensional (3 spatial dimensions and one temporal dimension); in the two-dimensional diagrams, objects could only move along a single spatial axis.

characterized by a temporal "qualitative multiplicity" which he calls *duration* (*durée*), and this is to be contrasted with our perception of space, which is quantitative and measurable, and whose metaphors dominated then-popular ways of thinking—including (especially) those of time and consciousness, which Bergson aimed to revise.[281] Bergson wanted to rediscover a fundamental distinction between space and time, which he blamed Kant for subtly conflating. Later, in *Creative Evolution* (1907), Bergson used the metaphor of the cinematographic film (which at that time would have recorded around 15 frames per second) to show how time, in the view of the physicists, was discretized and turned into uniform points on a single spatialized line. For Bergson "the contrivance of the cinematograph" is identical to that of popular knowledge about time (Bergson 1944 [1907]).

How would Bergson, then, understand the stock ticker? For it is certainly experienced as a kind of flow, but one overtly made up of distinct, unpredictable symbolic events (namely, the reported execution of trades on the trading floor, as announced by the trading floor keyboard). The brokers who came under its spell indeed seemed to be immersed in a spiritual hypnosis; but it is one formed from nondeterministic interruption and not duration. The ticker, which does not print characters in a continuously periodic motion, does not spatialize time in the manner critiques by Bergson; but is also quite literally spatializes time by converting the market's activities into a linear tape. Moreover, as we will argue below with the help of Mead, the *broadcast* quality of the ticker extends beyond any individual's subjective experience, and is thus instead profoundly *social*). Beyond its synchronic presentism, the ticker manages to throw a wrench into Bergsonian thought in every other one of its aspects.

---

[281] (Bergson 1910).

In contrast to Bergson—who was raised in London and Paris and entered the École Normale Supérieure at the age of 19—Gaston Bachelard worked for the *Postes, Télégraphes et Téléphones* (PTT — the administrative unit which later become France Telecom in the 1980s) for some years after his secondary education. Transferred to Paris, Bachelard took night classes and acquired a *license* in mathematics in his twenties, and he had applied to be a professional telegraph engineer before World War I broke out (Chaplin 2007). Entering the Sorbonne in his late 30s after multiple years in the trenches, his advisors Abel Rey and Léon Brunschvicg were opposed to Bergson's perspective (Chimisso 2001).

One can get an initial sense of Bachelard's intervention towards Bergson through the titles of his early books; with *The Intuition of the Instant* he is problematizing Bergson's claim that it is only *duration*, and not instantaneity, that one can experience and understand through an introspective 'intuition; and with *The Dialectic of Duration* he wants to put into dialogue with Bergson's 'qualitative multiplicity' that which Bergson was resolutely opposed: the *quantitative* multiplicity of discontinuities, "lacunae", and events.[282] Bachelard's critique of Bergsonian duration was born from the observation that the subjective experience of temporal phenomena can vary, and that the conception of a single *dureé* (duration) was insufficient. He argued, instead for a duality of duration and event (which I would characterize as related to our dichotomy of contact synchrony and contact asynchrony):

---

[282] It should be noted that Bachelard's dialectic is not a Hegelian dialectic (in part because Hegel's *Phenomenology of Spirit* remained untranslated in France until 1939). Bachelard's view is that science moves forward by overcoming "epistemological obstacles" (Gaukroger 1976); Chimisso describes Bachelard's distinctive view of the dialectical development of the sciences as follows:

> "For Bachelard, science advances in a ceaseless overcoming of its negations, that is, of epistemological obstacles. Obstacles are produced by human imagination, and as such are negations of rational knowledge. However, they are necessary to the process of knowledge, for knowledge can advance only by negating those negations." (Chimisso 2001, 85–92)

> Botanists who limited their science to saying that all flowers fade would just be doing the same thing as some philosophers who underpin their theories by repeating that all things pass away and that time flies. We very soon saw that between this passing of things and the abstract passing of time there is no synchronism, and that temporal phenomena must each be studied according to its appropriate rhythm and from a particular point of view. When we examined this phenomenology in its contexture… we saw that it always comprises *a duality of events and intervals.* In short, when we looked at it in the detail of its flow, we always saw a precise, concrete duration that teemed with lacunae [emphasis added] (Bachelard 2000, xii).

And Bachelard denies that Bergson's homogeneous thought is possible without the possibility of discontinuous redirection or interruption:

> "Bergson takes psychological intuition to be a priori a continuous thread, imposing an essential unity on experience as though experience could never be contradictory or dramatic… Even in the most homogenous order of thought, you cannot go from one essence to another by continuous thought" (Bachelard 2000, 42).

With Bachelard, we can understand that the sense of duration can arise from a quantitative multiplicity of instants and discontinuities; and as I will show later, it is this claim on which much of today's digital, networked world—which would likely have been so largely unexpected for the telecommunications engineers of the early 20th century—relies.

The reason for analyzing and quoting Bachelard's opposition to Bergson is that from my perspective, Bergson represents a philosophy of temporal experience which is so aggressively qualitative and continuous (and, thereby, indexical and synchronous) that it unfortunately resists application to the technics of messaging: i.e., of the asynchronous arrival of symbolic (and switched/routed) data, which nevertheless can be differentially experienced or reasoned about as a kind of continuous flow. Message-based communication networks, as I develop them here, thus create for their users a different *kind* of Bergsonian duration, which may (in a context of high message volume and high reliability) nevertheless still be phenomenologically experienced as Bergsonian duration: consider the subjective sensation of even a synchronous (call-and-response)-like activity of "surfing the Web"—which is in fact composed of the disorderly arrival of discrete messages (in the form of HTML documents and images).

As Abbott (2001, p. 23) points out, Bergson's theory is wholly *asocial*, and Bachelard—despite his influence from Pierre Janet, a French psychologist who shared Maurice Halbwachs' perspective that memory is ontologically social—does not significantly improve on this state of affairs. (Where Bachelard is ultimately a social philosopher, it is in his recognition of the pedagogical relation of teacher and student in his theories of scientific knowledge.) To move to a theory of telecommunications which can be ontologically social, we must instead consider George H. Mead, who in his posthumous work *Philosophy of the Present* declared at the outset that "[t]he world is a world of events" (G.H. Mead 1932, 1).

Mead's present-centric perspective has its origins in Bergson's *Time and Free Will*—but Mead (correctly, in my view) rejects the Bergsonian notion that because all is continuous (if heterogeneous) flux (duration/*durée*) we must necessarily privilege an introspective/psychological perspective.[283] Mead goes even further than even Bergson, in his claim that the present is all there really is. Unlike the transaction abstraction, in which completed operations can potentially be undone, for Mead the past "...is expressed in irrevocability". This is a clue that the world he envisions is not like the world of the relational database, and instead is about that passage in time which the transaction abstraction seeks to erase. In Mead's presentism, the occurrence of emergent events is how we know time (Abbott 2001, 227); there is no past or future in themselves, only past and future as they relate to the passage of emergent events in the present. (Adam 1994, 39). He writes:

> "The social character of the universe… we find in the situation in which the novel event is in both the old order and the new which its advent heralds. *Sociality is the capacity of being several things at once*" [emphasis added] (G.H. Mead 1932, 49).

---

[283] (Emirbayer and Mische 1998); (Joas 1997). For Mead on Bergson, see George H. Mead and Moore (1936).

This is a radical definition of sociality, and allows us to make claims about the *relative* sociality of communication technologies based on their synchronic/asynchronic qualities, their unicast/multicast qualities, and their data/codata qualities. One can give Mead's definition of sociality a *Peircean* reading, in which the interpretation processes core to Peirce's semiotics—which can occur in a mix of iconic, indexical, or symbolic modes—are precisely those which demarcate a past from a future, by projecting and/or refracting signs from the present into the future.[284] In this hybrid Mead/Peirce view, sociality is a function of the interpretants (i.e., interpretation-processes), as signs flow through the present; and the more possible interpretants there are for a given sign-object, *the more social the sign*. (Pragmatically, the variety of possible meanings would be constrained by social norms, expressed in other modalities, such as the stylistic 'objectivity' of a newspaper.)

Peirce was never strict about the mental character of his semiotic interpretants; his theory of responses to signs could, without serious modification, be migrated to a sociotechnical attitude which takes humans as ontologically technical beings. Moreover, interpretants are themselves sign-generating processes, of which further interpretants can take as their sign. One might argue, then, that the *stock ticker* is a sociotechnical device/network which, in its transformation of the sender's intepretations of trading floor activity (transducing from indexical electrical modulation to typed symbols), itself facilitates further pragmatic intepretations for a *multiplicity* of human tape readers on the receiving ends—thus creating a potentially more uniform world of signs for brokerage houses and bucket shops (and today, for cable news viewers and Yahoo! Finance users); this constitutes the market 'lifeworld' for Knorr Cetina. However, via Mead, we can also

---

[284] For this analysis I largely restrict myself to Peirce's writing on the 'Division of Signs' and on 'Icon, Index, and Symbol' (Peirce, 1931, sec. 2.227–2.308).

see the ticker as intensely *social* in that its spatially-replicated utterances can simultaneously have a wide variety of meanings to its different addressees: the updated price of a stock can mean riches for one broker and ruin for another, and relative indifference in so many more. The ticker was thus both a device and network for making the markets "more social" in Mead's sense.[285] From this perspective we can also observe a distinction between the 'antisocial' valence of the archive (i.e., static data) in comparison to these multicast flows of data-in-motion.[286]

It is true that continuous trading in financial markets today is dependent on these flows of data in the present, or as close to the present as possible; the brokerage office's world (and those of the bucket shops) has for a long time been a Meadean "world of events". But at the same time, the emphasis on the present specifically in financial technology (and in the not-unrelated practice of gambling) is matched by its emphasis on the *future*; to use an example of a longer-term contract, a mortgage is valuable in the *present* to the borrower, while it is valuable in the *future* to the lender. While it is common to think of *trading* in general as an exchange of property and money, it is arguably better thought of as this sort of exchange of temporal value, between value in the present and value in the future. As the management professor and historian William N. Goetzmann has most concisely put it, "financial technology is a time machine we have built

---

[285] This perspective may induce an interesting critique to common complaints about *mass* or *broadcast* media, in which their existence facilitates not some kind of homogeneity but a kind of sociality. One might see the argument of *Imagined Communities* (Anderson 1983) was that the multicast *sociality* (and thus possibility for multiple interpretations) engendered by the distribution of the "publish-subscribe" newspaper was actually multicast nationalism.

[286] In particular, the pedantically archival aspects of particular social media applications like Facebook (whose backend systems save most of their users' actions for eternity) should be seen as an artifact of their technological environment (which, in the early 2000s, privileged the promise of database systems to archive entire organizations for later analysis), and not a universally social technique, as if the world had always been made of note-takers and scrapbookers. This is in contrast with applications with appear to privilege the ephemeral (e.g. Snapchat).

ourselves".[287] Emirbayer & Mische, in their discussion of agency, take the position that agency is simultaneously (but differentially) oriented towards the past, the present, and the future[288]; and we can see financial sociotechnics as a style which is perhaps significantly more oriented towards the present and the future than towards the past.

In the next section, I will argue that the intellectual resistance by computing practitioners to implementing techniques of multicast and potentially-infinite streaming data (as well as the asynchronous styles of communication which will ultimately engender a more intensive and diverse phenomenological synchrony) are precisely related to (conscious or unconscious) strategies to control and eliminate the complex interpretative processuality and indeterminacy of social life—for example, by representing an entire organization's state as a centralized and static hierarchy, as in early database models.

Early Distributed Systems: Towards a Bus of Messages

The field of *distributed systems* is focused on computing techniques and technologies using more than a single, centralized computer, in which machines and/or sensors nevertheless might appear to the system's users as a single coherent system.[289] The term 'distributed' in this context was initially popularized by Paul Baran's 1964 RAND report "On Distributed Communications", which considered what would be required for "a future all-digital-data

---

[287] For this simple and yet highly fruitful insight on the temporality of finance I owe these observations of William Goetzmann: see Goetzmann and Rouwenhorst (2005) and Goetzmann (2016).

[288] (Emirbayer and Mische 1998).

[289] This concept of a *distributed system* as conceptually coherent is to be distinguished from the more general topic of *computer networks* in which this goal of conceptual coherence from the user's perspective is not explicitly required. It can be noted that early researchers' attempts to define the field converged on the necessity of a specific issue: that of an unpredictable temporality between the production of a communicative event and its arrival at its destination (LeLann 1981).

distributed network which provides common user service for a wide range of users having different requirements" (Baran 1964). Research on this topic went under the name of "distributed computing" in the 1970s, and as the relevance of work in the field of data(base) management systems increased through the late 1970s, it became increasingly common to also see "distributed computer systems" or "distributed systems".

What were the early higher-level mechanisms for communicating from a program on one computer to a program on another (as opposed to from a terminal to a mainframe—what IBM would call "host computing")?[290] From Table 4 we can see some examples of the transmission of discretized information; and for the purposes of this chapter we would like to focus on the distinction between those forms of discretized communication which are denoted as message-*synchronous* (in which the sending process' writing and the receiving process' reading occur simultaneously) or message-*asynchronous*, in which the sending and receiving processes are *not* in some direct, indexical contact during respective writing and reading (also referred to as being *decoupled*);[291] some examples of the latter includes sending and receiving letters in the mail, subscribing to a newspaper, Telex (in which teletype messages are routed and resent in an exchange-like process), and email (routed in a similar manner, but relying on the routing of the Internet Protocol (IP) instead of a teletype exchange).

---

[290] By "high-level" I mean to indicate an eventual focus on the phenomenological experience of *software* and its users (or, in this chapter, *middleware* which resides between software and the operating system and network); this will correspond to, what, in the OSI 7-layered network model, would be anything above the *transport layer* (e.g. TCP, the Transmission Control Protocol, itself dependent on the well-known packet-delivering Internet Protocol (IP) at the *network layer* one level down).

[291] The term 'synchronous' (which occurs frequently in the early 20th century when 'asynchronous' emphatically does not) originally referred to temporally synchronized processes; so that, e.g., a *synchronous motor* is one where the shaft's rotation corresponds to the provided alternating current; *synchronous communications* require that the sender and receiver be synchronized to a common clock.

Most forms of networked data communication between terminals and central computers before the 1980s were a form of *synchronous* communication—in which the sender and receiver process must temporally synchronize during the sending/receiving.[292] This was true for the earliest "batch" transmissions of data, which merely piggybacked on existing telegraphic communication: IBM developed machines to convert from punched cards to the paper tape understood by teletype systems and vice versa.[293]

IBM first introduced terminals with "two-way response" in the early 1960s; the IBM 1062 teller terminal, for example, was designed specifically for financial services after a study with First National Bank in Chicago.[294] IBM's systems in the early 1970s had an overwhelming variety of communications techniques—"35 different teleprocessing access methods and 15 different data link controls"[295]—but in the mid-1970s both IBM and the International Telegraph and Telephone Consultative Committee (CCITT) developed more comprehensive competing international standards for data communications. IBM's standard, introduced in 1974 for networks with a single host mainframe, was called Systems Network Architecture (SNA), and CCITT's was called X.25; these data communications standards proceeded in implicit opposition to the more 'open' proceedings of the ARPANET work.[296]

---

[292] As described by MIT's Barbara Liskov in her 1979 paper "Primitives for Distributed Computing" (Liskov 1979), there are three possible variants of communication primitive: (1) *no-wait send* (what I call here 'asynchronous'); (2) *synchronization send* (what I call'synchronous'); and (3) *remote invocation send*, a synchronous send which explicitly waits for a response.

[293] (Jarema and Sussenguth 1981).

[294] (Jarema and Sussenguth 1981).

[295] (Spragins, Hammond, and Pawlikowski 1991, 138).

[296] (Russell 2014); (Abbate 2000). On IBM's SNA as predicated on "the advent of distributed processing" see Sussenguth (1978).

While our framework of distinctive features in telecommunication (sync/async, unicast/multicast, and data/codata) is intriguingly applicable to the pragmatic and technical debates which led to the eventual dominance of the nascent Internet's packet-switching protocols (at the internetworking level) and of data protocols like Ethernet (at the level of the local area network), I will skip ahead to the debates for inter-process communication within distributed systems (whether it be on the Internet or on a local area or wide area network). For the story of TCP/IP (a form of discretized, contact-asynchronous, relayed communication across multiple networks), see (Abbate 2000); and for the story of Ethernet (a broadcast, contact-asynchronous, message-synchronous local area network protocol), see (Burg 2002). In the rest of this section I will describe the development of asynchronous and multicast data communications paradigms which specifically emerged in the distributed systems research inspired by the nascent Internet; these paradigms would come to profitably cross paths with the heterogeneous, increasingly digital environment of the financial industry in the late 1980s and early 1990s.

The Remote Procedure Paradigm

The development of the *remote procedure call* (RPC) paradigm—in which communication between processes in a distributed system happens in analogy to calling/invoking a procedure in a programming language like COBOL or Pascal—appears to be a fascinating case of a misunderstanding between the concept of *duality* and the concept of *equivalence*. In August 1978, a paper entitled "On the Duality of Operating System Structures", by Hugh Lauer (of Xerox PARC) and Roger Needham (of University of Cambridge), was presented at the Second International Symposium on Operating Systems at IRIA in France. The authors argued that one could conceptually divide the operating systems of that era into two types: one which was *procedure*-centric and one which was *message*-centric. Difficult to read today—as a contemporary reader will scarcely be familiar with many of the now-dead operating systems it uses as examples of both the procedure-centric and message-centric types—at the time it was interpreted as an argument that when faced with a choice to implement inter-process communication in the form of a procedure primitive or send/receive message primitives, each were *equivalent*.

While Lauer and Needham's discussions were restricted to the internal implementation of a single computer's operating system, and did not directly address communication between more than one computer, its origins at Xerox PARC meant that it did not take long for their dichotomy to reach those interested in those interested in distributed systems. The University of Texas computer scientist James Peterson documented in his notes for an October 1978 workshop on distributed computing at Harvard (incidentally attended by Jim Gray) that Howard Sturgis of Xerox PARC had brought up Lauer and Needham's recent paper, an act which "provoked a heated argument about the extension of this duality to distributed systems" .[297] In Peterson's retelling, a suggestion was made that "communication should be by a *remote procedure call* rather than sending a message and waiting for a return" [emphasis mine]; but for some present, the obligatory request/reply paradigm was problematic: "[i]t was argued that there exist systems which are structured as a pipeline, where all information flows in one direction only and this request/reply model is inappropriate."[298]

[297] Howard Sturgis helped introduce the term "client" as in "client/server computing" in a 1978 Xerox PARC research report on the Juniper distributed file system (Israel, Mitchell, and Sturgis 1978); see also Sturgis, Mitchell, & Israel (1980). Sturgis had also worked with Jim Gray on the CAL TSS experimental time-sharing operating system at Berkeley (Jim Gray et al. 1972).

[298] (Peterson 1979).

This term "remote procedure call" used by Peterson implicitly referred back to debates between researchers on the 1970s ARPANET at the Stanford Research Institute (SRI) and at Bolt, Beranek, and Newman (BBN) in Cambridge, Massachusetts. SRI in 1974 developed an approach "for inter-process and/or interhost communication and control" which they called a "procedure call protocol" (PCP), mentioned in a "requests for comments" memo (RFC 674).[299] SRI's proposed protocol was to "[create] a distributed programming and process control environment… In effect it makes procedures and data structures of remote software systems as accessible to the programmer as those within his own system."

At BBN, Richard Schantz replied to SRI's proposal with a critique later that year— "Commentary on Procedure Calling as a Network Protocol" (RFC 684, R. Schantz, 1975)— he "[voices] an objection to the 'PCP philosophy', in the hope of preventing this type of protocol from becoming the de-facto network standard for distributed computation" and "[takes] exception to PCP's underlying premise: that the procedure calling discipline is the starting point for building multi-computer systems"[300] His essential argument was that inter-machine communication is likely to be more analogous to *interprocess communication* than to invoking a procedure in a program; at the time, interprocess communication (also known as *IPC*) would occur either via (a) sharing random-access memory (which was not tenable for distantly located

---

[299] (J. E. White and Postel 1974). The "RFC" acronym for these memos is due to Stephen Crocker (Crocker 2009).

[300] (R. Schantz 1975). BBN, as the implementer of some the ARPANET's early gateway computers (known as the Interface Message Processor (IMP) systems), was also a site for early experiments in packet switching. (Kahn et al. 1970) BBN in the early 1980s would also work on Cronus [an early "distributed object computing environment", which focused on system heterogeneity. See R. E. Schantz (2006) for a detailed history of BBN's early work in distributed computing.

computers) or (b) a pair of send/receive commands which specify some abstract numerical "ports" for each process on which to send and receive data.[301]

In 1980, a workshop on distributed systems outside of San Diego found the nascent field in a philosophical flux. One proposed system, by David Cheriton, mixed the semantics of the procedure-calling metaphor of SRI with a more asynchronous queue of messages on the receiving side.[302] But the overall eventual influence of Lauer and Needham's argument on the distributed systems community was unmistakable. As prototypes of RPC were developed at the Xerox PARC research lab in Palo Alto (see Fig. 15), it led to conclusions like the following (from a PARC research paper by Bruce Jay Nelson, then a CMU Ph.D. student), based on the notion that the two paradigms were "equivalent":

> The decision to provide message-based or procedure-based control primitives should be founded on considerations of machine architecture and programming environment rather than on intrinsic properties of messages or procedures themselves (Nelson 1981).

This was, it would seem, tantamount to saying that there is no difference between the direct indexicality of line/circuit switching and the store-and-forward quality of message switching. As I hope I have established, these differences (between the synchronous and the asynchronous; and in terms of the utilization of the communication medium) are quite extensive.[303]

---

[301] (R. Schantz 1975). To understand the distinction between a procedure call discipline and a messaging discipline, consider that in the procedural metaphor, a sender must wait for the receiver to reply, and a receiver must wait to be invoked; there is in this system no way for the sender to communicate *indirectly* (by "leaving a message" to be responded to later). The former is analogous to our 'synchrony' category (sender and receiver must attend to the message's transmission simultaneously); the latter is analogous to our 'asynchrony' category.

[302] ("Report on the Workshop on Fundamental Issues in Distributed Computing" 1980).

[303] One sees here the hints of a longer-term pattern, in which a younger generation on the U.S. west coast, seemingly sufficiently ignorant of a previous generation's debates (and those before that), focuses on and promulgates a specific semiotic technique—whether it be the relational model (recall, in opposition to the rest of east-coast IBM), or synchronous RPC (in opposition to east-coast BBN and their deep familiarity with message-switching paradigms,

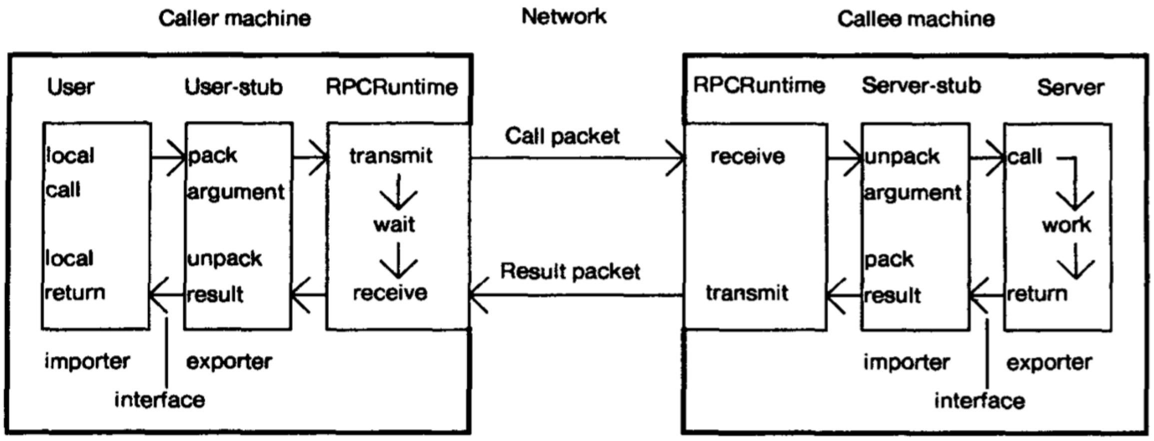

Figure 15. A diagram of a remote procedure call (RPC) from one machine to another (Birrell and Nelson 1984; © 1984 Association for Computing Machinery, Inc. Reprinted by permission).

While Nelson (and his collaborator Andrew Birrell) implemented RPC on Xerox Dorado workstations at PARC (Birrell and Nelson 1984), the most influential implementation was likely by Bob Lyon at Sun Microsystems; known as "Sun RPC", it had a relatively unrestrictive copyright, and the source code was posted to the Usenet newsgroup `mod.sources` in its entirety in April 1985, helping RPC to spread to systems at MIT and elsewhere.[304]

RPC and its Discontents

But by 1988, the Dutch operating systems researchers Andrew Tanenbaum and Robert van Renesse proclaimed that within their community "remote procedure call has achieved sacred

in the form of packet switching). Another example would be the return of hierarchical data formats in the case of XML in the early 2000s and non-relational document databases in the early 2010s.

304 Sun RPC later became known as ONC (Open Network Computing) RPC [RFC 1831]. According to (Callaghan 2000, 27), Bob Lyon referred to his implementation as "Sun of Courier", to imply that he was directly inspired by Birrel and Nelson's Xerox Courier RPC. For a survey of RPC systems see Tay and Ananda (1990) and Ananda, Tay, and Koh (1992).

cow status… It is almost universally assumed to be the appropriate paradigm for building a distributed operating system." They brought up several conceptual problems, including:

- "Who is the server and who is the client?": the assumption that remote procedure calls occur between a client and a server does not permit more ambiguous relationships of pushing and pulling data.
- Situations where a user wants to abort a command by sending an interrupting message (a situation where an asynchronous message-based system would be preferable).
- The RPC paradigm is "inherently a two-party interaction" and thus cannot *multicast* or have 1-to-many (or many-to-many) interactions.
- There is a lack of 'parallelism' (in this case meaning that the server is always waiting for the client or vice versa), and a lack of data streaming facilities.

Tanenbaum and van Renesse contrast synchronous RPC with a paradigm of "nonblocking SEND and RECEIVE primitives"—that is to say, asynchronous messages.[305] They also use an example of a Unix pipeline — the terse terminal command-line functionality to invoke a program/process and stream its output as the input of another process[306] — and show how if someone tried to create a distributed Unix operating system which used RPC as its

---

[305] Strictly speaking, the dichotomous category of *blocking*/*non-blocking* communication is not the same as the *synchronous*/*asynchronous* dichotomy (while 'typical' RPC is blocking and 'typical' asynchronous messaging is non-blocking, variant implementations do exist). The former refers to whether a sending or receiving action returns immediately to the invoking procedure or process, regardless of whether synchronous or asynchronous / store-and-forward communication is happening "under the hood". See Cypher & Leu (1994) and also (Houston 1998) for clarifications on this distinction. The terms *blocking* and *non-blocking* can be traced to mid-century telecommunications switching technology, e.g. (Clos 1953); see also (Joel 1982).

[306] We will discuss Unix pipes in more detail below.

primary communication primitive, it would not always obvious which process is the “client” (i.e. the invoker of the procedure) and which is the “server” (the invokee).

Indeed, alternative systems to RPC, by that point, had already begun to be developed. Along with the procedural/message-based distinction—which is analogous to our ‘contact synchrony’ and ‘contact asynchrony’—there are two distinctive features of interest: (message-)synchronous vs. (message-)asynchronous and unicast vs. multicast.

	Procedure vs. message (similar to contact synchrony vs. contact asynchrony)	*Message synchrony vs. message asynchrony*	*Unicast vs. multicast*
RPC (Xerox PARC)	procedure (i.e. request-reply)	synchronous	unicast
V KERNEL (Stanford)	**message**	synchronous (send); **asynchronous** (receive)	**multicast** ("group communication")
ISIS (Cornell)	**message**	**asynchronous**	**multicast**
TEKNEKRON INFORMATION BUS (TIB) (Teknekron Software Systems)	**message**	**asynchronous**	**multicast**

Table 6. Distinctive features of four prominent communication paradigms and/or systems for distributed computing.

I will introduce three of these multicast or "group communication" systems: the V kernel of David Cheriton at Stanford, the ISIS system of Ken Birman at Cornell, and the Teknekron Information Bus, produced by an originally Boston-based startup called Teknekron Software Systems, which eventually became the Silicon Valley firm Tibco. While the distinctions between these systems may initially seem pedantic, I argue that their distinctions, and in the debates they engendered, represent a kind of *pragmatic* and *commercial* philosophical materialism towards the nature of events and causality.

Publishing in the Free Marketplace: The V Kernel (Stanford, 1983)

In the early 1980s, Stanford professor David Cheriton (along with Willy Zwaenepoel and Tim Mann) implemented an experimental operating system called the *V kernel* for small networks of Sun workstations communicating over *Ethernet*, an early high-speed digital networking system, using coaxial cable, developed at nearby Xerox PARC in the mid-1970s.[307] Ethernet was distinctive in its strategy of *broadcasting* packets of data to all [computers] in the network, each of which is "listening" for packets specifically addressed to them. Cheriton, in his work on the V kernel, took this analogy to the level of communication between computing processes, conceiving of a multicast mechanism for inter-computer (and inter-process) communication. Interestingly, the metaphor he used for his distributed system was a "*free marketplace*… services are offered by *servers*, while *clients* communicate with servers to negotiate and receive services." "In contrast to this free market model", he wrote, "a single machine operating system acts as a *centrally planned economy* [in which] hardware resources are controlled and allocated by a benign dictator that provides services to applications" (D. R. Cheriton 1984).

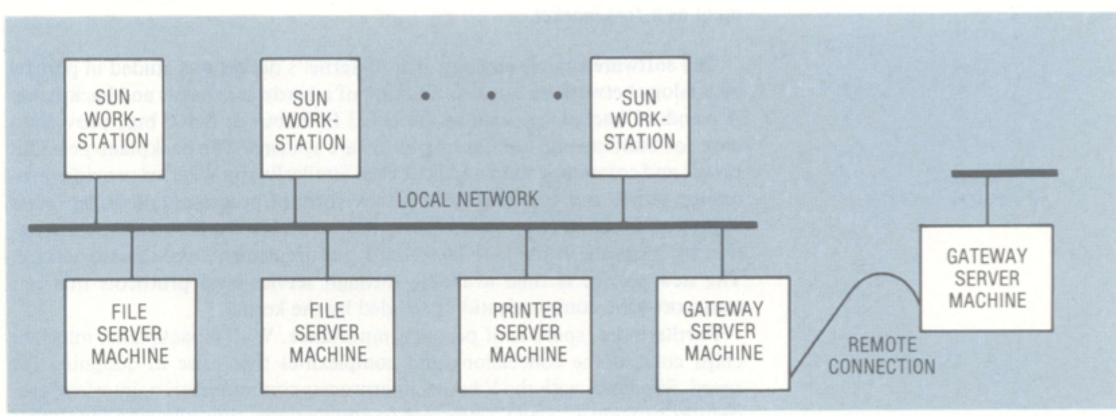

Figure 1. A V domain of local network-connected machines.

Figure 16. Illustration of the V kernel; from (D. R. Cheriton 1984; © 1984 IEEE. Reprinted by permission).

---

[307] (R. M. Metcalfe and Boggs 1976). Cheriton's "V kernel" should not be confused with the contemporaneous Unix 'System V' (5) operating system.

The V kernel, as per Table 6 above, used messages to communicate between workstations, file servers, and other machines on the network (see Fig. 16).[308] However, in Cheriton's design, the sending process acted as if it was calling a procedure, as in RPC; while the receiving process put incoming messages on a queue to be dealt with later (i.e., asynchronously), replying at its leisure while the calling process is blocked. Cheriton admitted that this was a curiously 'asymmetric' design in an earlier presentation in the 1980 workshop.[309] While the use of these semi-asynchronous messages as opposed to fully synchronous procedures (as in RPC) was not in and of itself a significant improvement, Cheriton's use of the intrinsically broadcast medium of Ethernet made it possible for the V kernel to implement what was then a novel style of *group communication*, such that one computer could communicate with multiple others using a single send message (see Fig. 17); because all the computers in an Ethernet network receive all messages, it was also theoretically plausible for a *subset* of computers to *listen for* a *subset* of messages, in what PARC members termed *multicast* (in opposition to unicast and/or broadcast).[310] This, as Cheriton indicates, "can be dramatically more efficient than group

---

[308] The V kernel's use of messages (independently of its multicast properties) was influenced by a handful of previous experimental operating systems emphasizing message-based communication, such as Per Brinch Hansen's RC 4000 (used for a real-time control system in a Polish ammonium nitrate plant) (Hansen 1967), the DEMOS system for a CRAY-1 at Los Alamos (Baskett, Howard, and Montague 1977), and the Accent system at CMU (Rashid and Robertson 1981).

[309] "The reasons for this view are that it is the simple, familiar semantics, it is commonly needed, and it is efficient to implement… The "asymmetric" semantics is justified mainly by experience. It fits well the server/client model, and also seems to handle pipes, or real-time systems. It seems like the natural way of looking at it." ("Report on the Workshop on Fundamental Issues in Distributed Computing" 1980).

[310] (Shoch, Dalal, and Redell 1982, 20). The terms *unicast* and *multicast* (regarding communication at the level of a process) should be distinguished from 'point-to-point' and 'multipoint', which refer to physical connectivity (how many hosts are joined in a given network). David Boggs' 1982 dissertation, *Internet Broadcasting* carefully distinguishes the former ("logical connectivity") from the latter ("physical connectivity") (Boggs 1982).

communication using repeated one-to-one transmissions"—as would have been necessary if communication in the V kernel were restricted to Xerox's RPC.[311]

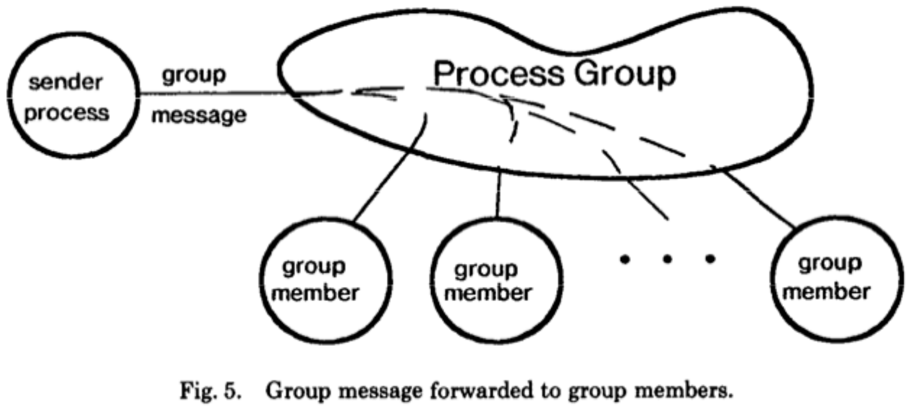

Figure 17. Illustration of *group communication*. From (David R. Cheriton and Zwaenepoel 1985; © 1985 Association for Computing Machinery, Inc. Reprinted by permission).

Where the V Kernel had obvious limitations beyond the use of synchronous send and receive, it was in the realm of reliability for its group communication; specifically, the researchers' definition for 'reliable' meant that only one member of the group of receivers needed to reply to the message; this criterion would not meet any industrial standards of reliability.[312] However, their system provided a fruitful metaphor which would become quite important:

> Putting the onus on the receiver for reliable delivery leads to what we call *publishing*. It is so named because it mimics real world publishing. That is, information to be sent to a group,

---

[311] It is worth mentioning that some prominent summaries of telecommunication styles in the 1970s did not isolate group communication or multicast as a distinctive method (despite, e.g., the existence (for some) of multiple-recipient Telex); instead, as in (National Academy of Engineering 1973), the categories included "Personal" (i.e. 1-to-1), "Mass" (1-to-N), from Dow Jones to radio and periodicals, but not an explicitly multicast (1-to-M or M-to-N) paradigm.

[312] It should also be noted, however, that Sun's RPC did eventually support a (sending) broadcast feature, which may have had similar functionality to the V kernel's group communication. (Tay and Ananda 1990).

> the *subscribers*, is filtered through the *publisher*, which collates and numbers the information before issuing it to the subscribers. A subscriber noticing a missing issue by a gap in the issue numbers or a new issue not being received in the expected time interval requests the *back issue* from the publisher. Thus, instead of automatic retransmission until the receiver acknowledges the message, the receiver must request retransmission if it is required (David R. Cheriton and Zwaenepoel 1985).

This *publish/subscribe* metaphor for group communication, as we shall see, ultimately became a common way of conceiving of the technique's possibilities and problematics. It is especially appropriate because of the analogy between the asynchronous and discretized properties of, e.g., distributing copies of the same newspaper to a number of geographically disparate subscribers. Cheriton, however, was cognizant of the difficulties that the request-reply procedural paradigm posed for multicast communication:

> "It appears reasonable to extend other message systems to support group operations in a fashion similar to V. *It is less clear whether remote procedure call mechanisms are amenable to group communication without seriously straining normal procedure call semantics*" [emphasis added] (David R. Cheriton and Zwaenepoel 1985, 105).

While Cheriton's "free market" terminology would not be developed further in later V Kernel papers—and in neither the V Kernel or his previous systems projects did he overtly indicate an interest in commercializing his work—it was an apt metaphor for a system design which would soon prove valuable in the nation's most high-volume marketplaces. Just as the packet switching of IP "lifted" a store-and-forward communication paradigm to the level of the network, the V Kernel "lifted" the broadcast communication paradigm of Ethernet to the level of the distributed systems application. The picture that emerges is that of communicative invention coming not from new patterns formed of whole cloth, but of the creative transposition of technical styles into different layers in a communicative hierarchy of similar features.[313] Later

[313] Other telecommunications paradigms contemporaneous with the nascent Internet transposed different paradigms. For example, the store-and-forward technique of teletype exchange was adopted for the purposes of distributing messages and files by periodic modem transfer instead of relying on Internet protocols; examples include UUCP and Fidonet, popular among bulletin board systems (BBS) administrators. Without an overt transport layer, distribution

‘innovations’ (i.e., transpositions of paradigms) would in turn make multicast communication a familiar experience at the application level for, e.g., users of social media platforms.

It can be seen that while “multicast” publish/subscribe techniques clearly existed (in, e.g., subscriptions for newspapers in the U.S. in the mid-1800s which were dependent on the existence of mail-routing procedures), the concept is re-invented (and seen as sufficiently novel) in a differing context of delivery (of message-switched data communications). I would argue that this sociotechnical variation of the history of knowledge is not so much a theory of grand ‘cycles’ but more that one should expect the redeployment of existing sociotechnical techniques (in this case, communication from one to many) on novel material-technological scaffoldings. So what the V kernel did for a local network (provide the potential for one-to-many data communication), social networks like Twitter do for the users of the vast agglomeration of local networks that is the Internet (although the multicast data might be limited, e.g., to 140 characters).

Birman’s ISIS System (Cornell)

Meanwhile, in upstate New York, the computer scientists Ken Birman, Thomas Joseph and others had been working on a project called *ISIS*, with the initial goal of providing “high-level support for fault-tolerant distributed computing”; this can be thought of as a very early attempt to conceive of what we now experience as 21st-century “cloud”-style computing, in which multiple servers provide the illusion of a single centralized system which can recover from the failure of multiple nodes; one of their early reports uses as an example a reliable, distributed

---

could be problematic; Fidonet required each node to store a list of all the other nodes and corresponding phone numbers, for example.

electronic mail system which could survive numbers of server failures.[314] The ISIS project's emphasis on reliability meant that they focused on certain potential problems with multicast communication, such as messages being delivered out of order, or in a different order to different recipients. This, suggestively enough, can be thought of as a similar situation to the simultaneity problems that challenged Poincaré and Einstein—except that the goal is for each node in a distributed system to handle messages (whenever they are delivered) in the same order as others.

Lamport and the Formalization of Event Ordering

Birman's solution to this problem was inspired by an ahead-of-its-time 1978 paper "Time, Clocks, and the Ordering of Events in a Distributed System" by one Leslie Lamport, which applied insights from special relativity—specifically, that events observed as 'simultaneous' differ for observers traveling at differing velocities—to the problem of computers in a network sending and receiving messages with unpredictable time delays, which could suffer from the same problems (so, i.e., different processes might observe different orderings of events). Lamport, a Brandeis Ph.D. who had noticed the relationship between special relativity and an RFC from two BBN employees about the difficulties in maintaining copies of databases on the early Internet (Johnson and Thomas 1975). However, Lamport's diagrams, modeled on the Minkowski-inspired diagrams portraying time on a vertical axis[315], provide an intriguing *pragmatic* twist on the "light cone" diagrams of spacetime in special relativity:

---

[314] (Kenneth Birman et al. 1985). This chapter only focuses on a small debate involving group communication, and cannot be considered a history of replicated data in general. Ken Birman has said that "[a] proper history [of replicated data] would also talk about the contributions of Amir, Babaoglu, Chockler, Dolev, Guerraoui, Kaashoek, Keidar, Meliar-Smith, Moser, Moses, Schiper, Stephenson, Van Renesse and many others" (Birman 2012, 306).

[315] Lamport cites the illustrated text by Schwartz (1962) and (Taylor and Wheeler 1966). It should be noted that Lamport (and his followers) are not in the least engaged with scholarship regarding a causal theory of time, held to be initially propounded by Leibniz and Kant and later becoming a debate in analytic philosophy and the philosophy

In relativity, the ordering of events is defined in terms of messages that *could* be sent. However, we have taken the more pragmatic approach of only considering messages that actually *are* sent. We should be able to determine if a system performed correctly by knowing only those events which *did* occur, without knowing which events *could* have occurred [emphasis in original] (Lamport 1978).

Fig. 1.

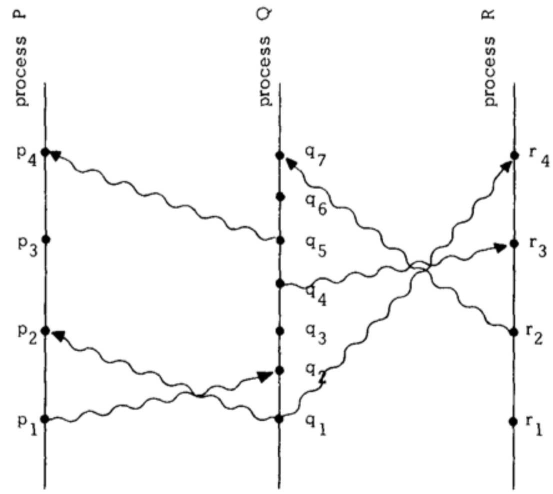

Figure 18. Lamport's adoption of two-dimensional Minkowski diagrams from physicists illustrating special-relativistic communication, named for (Minkowski, 2012 [1905]). (Lamport 1978; © 1978 Association for Computing Machinery, Inc. Reprinted by permission).

Lamport's diagrams, showing 3 different processes sending and receiving messages at various times, defines what he calls the "happens before" relation, denoted by the arrow → :

- *If a and b are events in the same process, and a comes before b, then a → b.*
- *If a is the sending of a message by one process and b is the receipt of the same message by another process, then a → b.*
- *If a → b and b → c then a → c. Two distinct events a and b are said to be* ***concurrent*** *if a ↛ b and b ↛ a*. (Lamport 1978, 559)

---

of science (Abbott 2001). See also Sklar (1977) who rejects the idea that temporal relations between events are the same as causal relations between events.

Lamport's "happens-before" relation is, to speak in the language of set theory (and he does), a *partial order* (it is *transitive*, *antisymmetric*, and *irreflexive*). As Lamport describes, the happens-before relation provides a sense of which events in a distributed system may be causally related; and (according to Lamport) "two events are concurrent if neither can causally affect the other". He goes on to show that if each process increments its own local "logical clock" (today known by students of distributed systems as "Lamport clocks") and sends that local incremented value along with each message, the processes can maintain a (somewhat arbitrary, due to concurrency) *total order*[316], which is in turn useful for forcing each process to simulate the same "State Machine". This total order is precisely the feature which Birman was looking for in his ISIS project.

Lamport's State Machine technique is thus an attempt to make something like a single process out of many, by finding a method to help ensure that each process deals with events in the same order. While before we have spoken of set theory as being largely *atemporal* (as in the case of Codd's relational model), I will argue that the use of the concepts of partial and total order in the context of distributed systems have an analogous decontextualizing effect. Just as the atomic transaction (as enforced by 2-phase commit and durable logging) is an attempt to mitigate the potential for failure across the passing of time (in the form of a database update that subjectively takes place in isolation), Lamport's synchronization is an attempt to mitigate the potential for failure in the "passing in space" (in the form of messages sent back and forth with

[316] A *total order* is transitive, antisymmetric, and *total*, meaning that (using Lamport's notation) either a < b or b < a; i.e., there are no unclear cases (as in Lamport's 'concurrency') where neither a → b, nor b → a.

arbitrary delays).[317] In both cases, debates will eventually emerge as to the necessity (and sufficiency) of these formalisms given the contextual embeddedness of real systems.

A Note on Bachelard: Lamport Clocks and the Dialectic of Duration

How can one place Lamport's technique, which seems so effortlessly to transpose Einstein's special relativity into a pragmatic and material reality, in the context of Bergson's critique of special relativity? As I mentioned previously, Bergson's opposition to special relativity was primarily towards the kind of abstraction which he called the *spatialization* of time, the reduction of all-things-temporal to a single (conceptually real-valued) dimension which one manipulates algebraically, which undergoes linear transformations depending on the inertial frame of the observer under consideration. Lamport's intervention instead generalizes this unknowability of simultaneity to a context in which communications can travel at any speed, not just at the speed of light. It does this also by spatializing time, by considering each event, each sending or receiving, as occurring at some ordinal moment from the relative perspective of sender or receiver; and so Lamport's abstractions would be objectionable to Bergson as well.

By contrast, Lamport clocks are wholly in line with Bachelard's understanding of causality in *The Dialectic of Duration*. Bachelard points out that, in contrast to Bergson's call for introspection as a method for inferring causality, that the very notion of causality presupposes a discretization of reality:

---

[317] While the phrase "passing of space" sounds fanciful, it is important to realize that when phrases seem to only be applicable to time and not space, they may be repressing their duality. Similarly, Stiegler, in thinking of the notion of "breaking the time barrier" (in this case analogous to the "sound barrier" which is surpassed with sufficient spatiotemporal velocity), says:

> This reflection can only acquire meaning when certain effects of technical development are carefully examined: namely, those that in computing one calls "real time" and in the media "live"—effects that distort profoundly, if not radically, what could be called "event-ization" *[événementialisation]* as such, *that is to say, the taking place of time as much as the taking place of space* [emphasis added] (Stiegler 1998, 16).

> …[A]ll causality is displayed in the discontinuity of states. We show one phenomenon to be a cause and another to be an effect when we draw a line round each of them which defines and isolates them, giving each the unity of a name, and revealing the essential organic character of each… We do not have to take account of the duration in the cause or of the duration in the effect in order to link them temporally… *There is nothing really objective in time other than the order of succession* [emphasis added] (Chimisso 2000, 53–54).

It thus does appear as if Lamport's rationalization of distributed clocks is in line with Bachelard's defense of the limited, but pragmatic, rationalization of causality.

'Virtual Synchrony' as the Dissolution of Space

Birman called his Lamport-inspired form of reliable ordering 'virtual synchrony'. As he later put it, "[t]he execution "looks" synchronous, much as a transactional execution "looks" serial."[318] We can see here an analogy between the *illusion of ordered synchrony* created by Birman's system—assuming that there are no unrecoverable failures—and the *illusion of atomicity and isolation* in the world of transaction processing. In the latter, the illusion is of making *time* meaningless; it is as if every transaction occurred in a single instant, as opposed to tediously modifying table after table. In the former, the illusion is of making *space* (i.e., physical distance) meaningless; it is as if all of the distributed proceedings were occurring in one place, instead of being distant (and therefore with an unreliable order of operations). There is thus a fascinating 'space vs. time' duality in the reliability engendered by the team at Tandem focused on centralized reliability, and the reliability engendered by the work of Birman and others in distributed systems.

As Birman's ISIS became adopted by actual users, the team found that while they had always conceived of ISIS as a distributed system from the ground up, that users actually preferred to connect it with existing programs, using existing network protocols. They observed

[318] (Ken Birman et al. 2006).

that users liked to use ISIS as a 'supervisor' for systems integration projects (where, e.g., a batch-oriented 'legacy system' was modified to run in a continuous and networked environment); and financial firms like brokerages liked ISIS for multicast in heterogeneous Unix environments; (Kenneth Birman and Cooper 1990). These were situations in which the possibility for transforming the entire organization to use a single conceptually-centralized system was pragmatically untenable; what these researchers would later discover is that their techniques were better used as a kind of 'glue' than as a general architecture.

And therefore, the infrastructures on which contemporary distributed technosociality is predicated would *not* be wholly predicated on the synchronous and antiphonic form of communication represented by the Remote Procedure Call. Just as human sociality had long been a mix of "full-duplex" and "broadcast" interaction (e.g. listening with accompanying facial expressions, talking over one another, giving speeches to multiple listeners) and asynchronous communication (sending and leaving messages for one another), our distributed computing systems would come to be accompanied by similarly full-duplex, multicast, asynchronous technologies. The cultural demand for those technologies would not, however, come from the general public. It would come from the financial industry.[319]

---

[319] In a 2008 retrospective, one prolific distributed systems engineer looked back at the fundamental problems with RPC, highlighting the disconnect between the illusion of a single system and the pragmatic reality of failure-at-a-distance:

> [T]he fundamental problem is that RPC tries to make a distributed invocation look like a local one. This can't work because the failure modes in distributed systems are quite different from those in local systems, so you find yourself having to introduce more and more infrastructure that tries to hide all the hard details and problems that lurk beneath. Network partitions are real, timeouts are real, remote host and service crashes are real… no amount of hiding or abstraction can make these problems disappear (Vinoski 2008).

The Message Bus in Finance: Teknekron Software Systems

Trading rooms in the mid-1980s were characterized by miles of cables, proprietary trading systems, and a variety of market data sources with heterogeneous displays cluttered on desks. The cables connecting these dozens of technologically incompatible information sources were braided and wound through desks and under carpets, and large electric fans attempted to cool down the masses of electronics.[320] There were a handful of companies producing incompatible market data terminals—the ubiquitous Quotron, as well as Reuters[321], ADP, Telerate, and Knight-Ridder, with hundreds of thousands of terminals in use worldwide.[322] Beyond these displays, Wall Street brokerages—here we are concerned with the "front end" of trading desks as opposed to the "back offices"—used crude video switches provided by companies like Micrognosis Inc. and Rich Inc. (founded in Chicago, and acquired by Reuters in 1985) to limit the number of necessary monitors, with a cost per trading desk of about $30,000.[323] While these firms gradually moved into digital switching technology—with Reuters having made more progress in London than in New York (Blackford 1988)—customers complained that the digital information provided was not easily integrable with their own computer environments.

In 1985, Vivek Ranadivé, an MIT student from Bombay, raised $250,000 from the pioneering Berkeley-area startup incubator Teknekron—which had been founded in the late

---

[320] (Thornton 2000); (Ranadive 1999). In the fixed-income market, it was common for dealers' desks to have five different monitors from different brokers (Blackford 1988).

[321] Reuters, the London news wire service, had entered the financial data industry in the 1960s by partnering with the U.S.-based Ultronic Systems to distribute 'Stockmaster' quotation/display terminals (Ransom 2014).

[322] (U.S Congress Office of Technology Assessment 1990, 133).

[323] (Roman 1987).

1960s by a mix of academics and entrepreneurs—to start his own company to address these issues in the financial industry; the incubator retained majority ownership and the resulting company was called Teknekron Software Systems.[324] Ranadive was not the only one developing something like this in the late 1980s; Sam Somech, at the time a consultant for Goldman Sachs, developed a product (then known as "Distributed Message Queuing" or DMQ) for the Bank of New York to receive clearing and settlement instructions on DEC VAXs from brokerage firms which kept track of positions on IBM mainframes.[325] Another company in New York (for which Somech would later be director of research and development), Systems Strategies Inc., offered similar products.[326]

Ranadivé's system integration tools were first selected by Merrill Lynch for their equity trading support unit in November 1988. The department's VP told *Wall Street Computer Review* that "Teknekron is giving us the glue to put it all together… The value we can add is the merging of externally available data along with internally available data."[327] Their goal at the time was to provide "customized presentation of multiple market information services as well as applications (deal entry, position keeping etc.) and analytics that require real-time market data" (Kondo and Chithelen 1988).

---

[324] Teknekron's first contract was "to model radio waves in an urban environment", and thus in some sense was involved with data-in-motion at its earliest stage (Baldonado 2015). I do not explore the matter here, but it is highly suggestive that Teknekron co-founder Harvey Wagner studied as an undergraduate under the philosopher of space and time (and founder of the University of Pittsburgh's Philosophy of Science department) Adolf Grünbaum, and called Grünbaum the "principal intellectual influence" on his life and credited him with giving him a "deep understanding of science and an appreciation of its role in modern technology". (It should be noted, however, that Grünbaum's writings express few issues with Einstein's spatialization of time and strictly relegate Bergsonian *durée* to an artifact of human consciousness.)

[325] ("IBM, Stratus, DEC Talk To Each Other With DMQ" 1988).

[326] "Products and Services : Packages Link VAX to IBM", p. 24, Network World , Mar 2, 1987.

[327] (Schmerken 1989).

In 1992, Dale Skeen, a Teknekron employee and former IBM researcher who had influenced the ISIS system, presented at USENIX—then a prominent conference where industry practitioners in computing could present papers alongside academic researchers—some of the details of what Teknekron Software Systems had been building. His paper, entitled "An Information Bus Architecture for Large-Scale, Decision-Support Environments" (Skeen 1992) explained his use of Cheriton's publish/subscribe paradigm and the "software bus" architecture (see Fig. 19) which used the analogy of a hardware *bus*—the name for a common communication framework for various components of hardware in a computer—to describe the system to which messages where published and disseminated.[328] Skeen describes the asynchronous, multicast communication model between publishers and subscribers as "subject-based addressing"; the idea being that a workstation could, e.g., notify the Information Bus of its subscription to "Eq.ibm.trade" in order to be asynchronously notified of all trades of IBM stocks, or to "Com.gold.news.reuters" to be asynchronously notified of all news items from the Reuters news feed about gold commodities futures.

[328] The basic design of a hardware bus goes back to the earliest computers—in the ENIAC this type of control was called a 'digit trunk' (Rojas and Hashagen 2002).

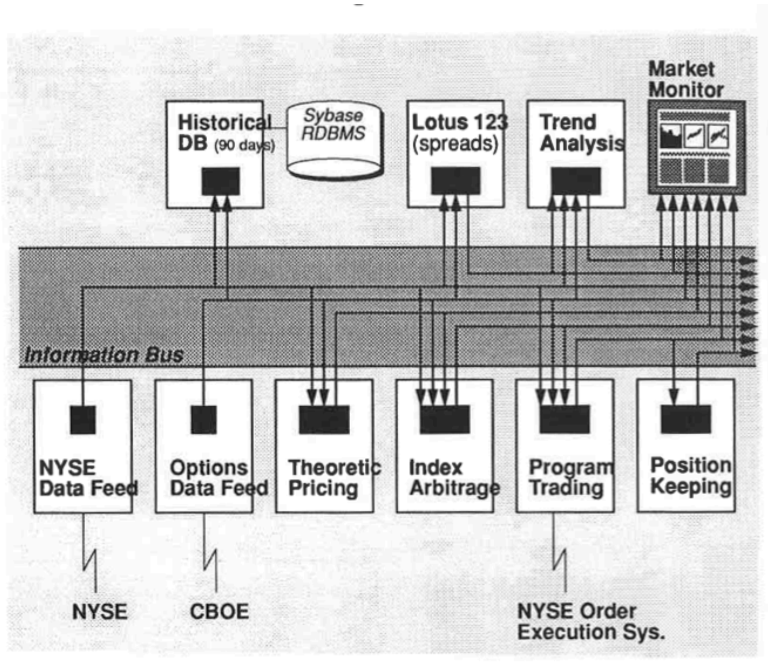

Figure 19. Illustration of the "information bus" architecture
(Skeen 1992; © 1992 Dale Skeen. Reprinted by permission).

Skeen's patent applications for the reliable publish/subscribe multicast architecture of the 'TIB' (short for 'The Information Bus') reveal the set of contemporaneous systems which provided distributed message-broker-like features outside of the financial industry. These include Usenet newsgroups; the Zephyr messaging system; and research on windowed development environments (the FIELD system). In each case, reading and writing messages was organized in the form of a (sometimes-replicated) publish/subscribe architecture, in an effort to decouple publishers who may not be wholly aware of all of their subscribers.

As mentioned, one of the other early 'systems integrator' firms developed a queue-like system to communicate between DEC and IBM systems; but DEC had itself developed its own application, known as PAMS ('Process Activation and Message Support') whose goal was to provide a common application programming interface (API) for heterogeneous systems. PAMS' origins were in a set of applications developed for automation at Inland Steel, and this lineage gives us a chance to reflect on the way that the automation of manufacturing processes likely involved the independent invention and reinvention of techniques which would be useful for all sorts of heterogeneous processes unfolding in time—namely, those of asynchronous and/or multicast queues of transduced events.

1994	Teknekron Software Systems Inc. acquired by Reuters Group PLC for $125 million[a]
1995	Goldman Sachs helps Teknekron develop Enterprise Transaction Express (ETX) middleware[b]
" "	JP Morgan replaces Micrognosis with TIB for 2000 traders[c]
" "	Salomon Smith Barney uses proprietary system based on TIB[d]
" "	Teknekron announces Rendezvous Software Bus to reach a consumer base outside of financial/manufacturing[e]
1996	Teknekron renamed Tibco, Inc.
1997	Ranadive spins off Tibco Software as a separate company selling to non-financial market. 65% owned by Reuters, with minority stakes by Cisco Systems and Mayfield Fund[f]
2000	1000 employees; revenue in 1999 $96.4M.

[a] (Wingfield 1999); [b] ("TST 10th Anniversary, Leaders Of The Pack" 1997); [c] ("TST 10th Anniversary, Leaders Of The Pack" 1997); [d] ("TST 10th Anniversary, Leaders Of The Pack" 1997); [e] ("Teknekron Debuts New TIB; Reuters Opts To Use Old One" 1995); [f] (Wingfield 1999)

Table 7. Abbreviated timeline for Teknekron Software Systems / Tibco after the acquisition by Reuters in 1994.

The Information Bus of Teknekron Software Systems had comparable functionality to that of Birman's ISIS project, in its goal of reliable multicast distribution of asynchronous messages. But as is shown here, this work was pitched towards the financial industry from the beginning, and best serves to illustrate the techniques which would come to be an infrastructural element of every trading room in the decades to come. In the following subsection, I will show how the message brokers and queues developed for the financial industry extended into a larger industry of "message-oriented middleware", and describe how the message- and event-centric models of computing remained in conflict with their dual—the synchronic request-reply model of RPC and its "object-oriented" descendants like CORBA (Common Object Request Broker Architecture)—throughout the late 1990s and early 2000s, as commercial organizations moved to access customers on the Internet.

The Middleware Concept

The term 'middleware' was introduced by Alex d'Agapeyeff—founder of the early UK software company CAP Ltd.—in the landmark 1968 NATO Science Committee conference in Garmisch, Germany on "Software Engineering" (a then-provocative title as the development of computer programs was, at the time, not considered to be a technical skill on par with the complexities of electrical and civil engineering). d'Agapeyeff provided a diagram called an 'inverted pyramid' (see Fig. 20) in which 'middleware' sits in between the applications and the system's service routines, and which might in part protect those programs from their dependency on 'lower levels'. However, the term did not catch on, and lay largely dormant for two decades, as software developers continued to write programs directly 'above' the operating system, with the exceptions being software for transaction monitors like CICS and Tandem's, which indeed provided a kind of application-level 'middle' ground which obscured some of the communicative complexity of on-line transaction processing.

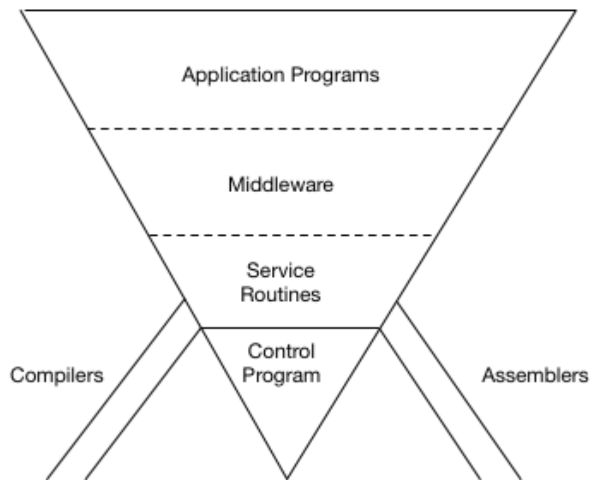

Figure 20. The 'inverted pyramid' of d'Agapeyeff.
Source: author, after Naur and Randell (1969).

But in the early 1990s, as the term re-appeared, precisely in information technology industry journals discussing 'systems integrators' along the lines of Teknekron Software and the aforementioned Systems Strategies. The term's revival was accompanied by the emergence of an 'open systems' approach to networking and software development, first positioned against the hegemony of proprietary IBM hardware and increasingly inspired by the proliferation of Unix operating systems, and later by the late-1980s formation of the Open Software Foundation (OSF) by Digital Equipment Corporation (DEC) and other companies to organize against the potential dominance in the Unix space by AT&T and, at the time, Sun Microsystems.[329] Some commentators of the early 1990s saw middleware as a way to literally bridge the emerging world of 'client-server' computing, characterized by personal computers (PCs) connected to Unix or

[329] The longer, and more interesting, connections between the systems theory of (von Bertalanffy 1968) and the open systems movement is discussed by (Russell 2014).

VAX VMS minicomputers, and the still-present world of mainframe computing.[330] The 'hardware bus' metaphor (of a central place where data from heterogeneous peripherals pass) was thus transposed, not just to the level of streams of stock quotes, but to the motley and heterogeneous collections of systems which had come to characterize the data processing and IT departments of large firms.

Initially, the revived term 'middleware' conflated all types of communication in distributed computing systems; so file transfer protocols were middleware, but so was any kind of software for distributed databases; an RPC library was middleware, but so were the more 'message-oriented' middleware systems like Teknekron and its competitors like DEC's MessageQ and System Strategies' ezBridge Transact. It became gradually apparent that message-oriented middleware (given the acronym of 'MOM') was distinctive in its asynchronous qualities, and could be seen as *complementary* but not identical to RPC-style communication (Dolgicer 1993).

IBM, in particular, realized in the late 1980s that it had no comparable product for what was needed in the financial industry—a way to communicate between their existing transaction processing systems like CICS (often running on IBM's MVS operating system) and an increasing variety of non-IBM systems.[331] Two employees at IBM Hursley in the UK, Rob Drew and Dick Dievendorff, however, had previously worked on a prototype 'queue manager' for MVS; but in order to provide compatibility with Unix and Tandem systems (for which Hursley employees were wholly unfamiliar), IBM decided to partner with New York's Systems

---

[330] (Millikin 1992).

[331] (Flaherty 2011). MVS (for Multiple Virtual Storage) was a common operating system for the System/370 mainframe and its descendants. It should not be confused with *VMS* (for Virtual Memory System), an operating system produced by DEC for its VAX minicomputers.

Strategies—such an external partnership, for IBM, being extremely uncommon—to provide the under-the-hood support for IBM's newly proposed message queueing API. The resultant product was known as MQSeries, and in early press releases in March 1993—announcing the partnership with Systems Strategies—IBM was keen to emphasize the asynchronous aspect of MQSeries: "a concept a bit like programs leaving phone-mail messages for each other".[332]

Unix, the Stream Metaphor, and Codata

Queueing software like IBM's MQSeries can be understood as the *unicast* version of a message broker like TIB (with each queue only corresponding to a single sender and receiver). But the asynchronous aspect of messaging middleware (whether unicast or multicast), for 1990s commentators, usually took precedence over the 'codata' aspect (outside of casual references to 'streams of information'); the industry analyst Roy Schulte, who wrote research notes for Gartner on middleware for over a decade, for example, usually focused on the asynchronous and/or 'connectionless' (i.e., non-indexical) aspect.[333] But I would argue that the concept of *messages-as-flow* is equally relevant (both technically or philosophically), and that this quality is likely related to these middleware systems' origins in both academic Unix cultures and the late-1980s 'open systems' community, both oriented around a genre of operating system whose basic primitives—such as the method of reading and writing files, or to perform basic interprocess communication—came intrinsically in the form of codata: a source of information which might never terminate.[334]

---

[332] (Newsbytes News Network 1993).

[333] (Schulte 1996a); (Schulte 1996b); (Schulte 1997); (Schulte 1998).

[334] This aspect of Unix derived from the decision by the designers of Unix not to distinguish between "random access" and "sequential" reading and writing of files, and not to impose any particular fixed record size for data. (D. M. Ritchie and Thompson 1974, 367).

This phenomenologically presentist aspect of Unix, which is in conceptual opposition to 'batch'-like approaches which iterate over a finite and known quantity of records (which Unix can also support, if the programmer insists), is both known and sometimes invisible to practitioners. Its most well-known manifestation is the feature known as the *pipe* (signified at the Unix command line by the character '|'), which permits one process to communicate with another in the form of a flow of reads and writes; this was implemented at the behest of Doug McIlroy, who in 1964 wrote a Bell Labs memo which included a short manifesto of "what's important", whose first item was that "[w]e should have some ways of *connecting programs like garden hose*—screw in another segment when it becomes when it becomes necessary to massage data in another way" [emphasis added].[335] As implemented by Dennis Richie and Ken Thompson at Bell Labs, Unix provided these 'pipes' from the outset (D. M. Ritchie and Thompson 1974); the pipe metaphor would later be extended in the framework for terminal and interprocess communication known as 'STREAMS' for Unix System V in the mid-1980s, which had a more general message-passing interface (Dennis M. Ritchie 1984). It seems plausible that the early systems integrators, who strove to implement message-passing from mainframes to Unix and other systems, found themselves necessarily adopting the same infrastructural metaphor as a matter of course.

Financial Middleware at the end of the 1990s

Finally, we can return to SIAC, the backend organization of the New York Stock Exchange. In 1998, Eliot Solomon, a VP at SIAC, detailed the growing importance of middleware at the exchange over his 12 years of tenure, involving the in-house development of

---

[335] (Dennis M. Ritchie 1999).

middleware libraries (called Common Software) and describes the then-present-day setup in 1998:

> You will notice that we sandwich message switching systems between our functional systems (or our functional systems between message switching systems). *We use extremely reliable asynchronous messaging middleware… which is a fundamental part of our basic approach to, and delivery of, ultra-reliable systems*. Indeed, to achieve this, we had to build our own middleware.
>
> *Wherever you look into SIAC's trading systems, you will find asynchronous messaging infrastructures.* In some ways we regard ourselves as being primarily a message switching company. Some- times, though, we look at it a little differently. From this angle we see ourselves as a factory automation or process control company.
>
> Both are valid views. Both demand middleware to enable disparate systems to work together" [emphasis added] (Solomon 1998).

SIAC had to develop its own middleware, linked to the Tandem machines, because the scale and scope of multicasting quotes and trades at a centralized stock exchange exceeded anything commercially available at the time. But one can see that the features introduced at the beginning of this chapter—the asynchronic, multicast, and flowing qualities of the message—were all found to be as crucial to the operating of a digitally enhanced stock exchange as the formalization of a transaction. In the next chapter we will see these technologies from a wholly different vantage point, that of competition and regulation, as the digital securities marketplace—composed of this dual blend of centralized transaction processing and middleware messaging—permanently disrupted the definition of the exchange, and provided the template for innumerable 'disruptive' marketplaces of the future.

CHAPTER 5

WHERE DO ELECTRONIC MARKETS COME FROM?

Introduction

The previous two chapters focused on the history of both the concepts and the commercialization processes of on-line transaction processing (OLTP, in Chapter 3) and message-oriented middleware (MOM, in Chapter 4), arguing that these techniques and technologies were fundamental to the digitization of the trading and back-end systems of the New York Stock Exchange; however, the role of *politics*, especially in an industry as heavily regulated as the securities industry, has been rather underaddressed. In this chapter, I will consider the exchange more generally as a *firm* and as part of a broader theory of markets and marketplaces, in order to show how the process of automation of the exchange was as much a political outcome as a technological one.

From a sociology of markets to a sociology of the exchange

The sociological study of markets is often characterized as a project intending to problematize the assumptions of neoclassical economic theory, with its efficient equilibria of rational actors (Fourcade, 2007; Fligstein & Dauter, 2007). This has, perhaps unintentionally, led down a path which emphasizes the analysis of *financial* markets—those paradigmatic sites which (at least in theory) realize particular notions of competition and information. But financial markets do not emerge spontaneously: they instead most often develop as a trade facilitation service, provided by particular institutions—namely, *exchanges*.

As described in Chapter 3, in the 19th and most of the 20th century, exchanges tended to be member-owned cooperatives. But the last two decades of the 20th century saw a significant transformation as these institutions became threatened by firms that provided automated platforms matching buyers and sellers. In this introductory section I highlight the importance of understanding and theorizing the transformation of exchanges for the sociology of financial markets. I detail the development and regulation of technologically-centralized and electronically-interlinked trading venues in the U.S. securities exchange industry, which were made possible through the development of on-line transaction processing and messaging middleware of Chapters 3 and 4; and I show how the role of the traditional stock exchange became blurred by a form of market "disruption" leading to the demutualization of exchanges, the fragmentation of financial market venues, and the potential for pathological high-frequency trading (HFT) practices.[336]

This chapter's story is about the ontological and discursive transformation of the exchange—what it is; what its legal definition is; and the historical relation between the two. The case at hand will demonstrate that the political transformation of markets on behalf of state regulators—while sometimes considerably removed from technological developments (in terms of direct action)—is inextricably informed and interwoven with technological processes.

Exchanges as fixed-role markets that produce switch-role markets

In 2005, Patrik Aspers—as part of a critique of Callon (1998)'s theory of performativity—made the claim that economic sociology "misses a crucial distinction between two kinds of markets: exchange role markets, such as financial markets, and fixed role markets,

---

[336] I choose to study the U.S. case because it is an early instance of the regulated interlinking and routing of orders for securities exchanges. For a comparison of U.S. regulations and those of the Market in Financial Instruments Directive (MiFID), see Boskovic, Cerruti, and Noel (2010); and for the relationship of algorithmic trading and MiFID, see Lenglet (2011).

such as producer markets for commodities" (Aspers 2005, 33).[337] His typological distinction was developed further in later works (e.g. Aspers (2007) and Aspers (2011)), changing what he called "exchange role markets" to "switch-role markets", to indicate more directly that actors on either side may *switch roles*: that is to say, it is possible (or common) for buyers to switch to becoming sellers, and vice versa. (See Fig. 21 for an illustration.[338]) The other primary ideal-type distinction introduced by Aspers was that of *standard* markets, where the good or service being exchanged is standardized and represented via some measure or contract; versus *status* markets, where the buyers and sellers are distinctive and can be ordered in relation to one another. The apotheosis of the *switch-role* and *standard* market, then, is a modern securities market, where a buyer can rapidly "flip" a stock within microseconds (i.e. switch from buyer to seller), and the goods being traded are perfectly standardized and fungible (i.e. the buyer or seller is solely concerned with that stock's price than the relational identity of the seller).

[337] By "producer markets" Aspers referred to what Harrison White isolated as *production markets* in his influential papers which called for a sociological understanding of interfirm competition (White (1981a), White (1981b)). Aspers' distinctions may be also seen as in the tradition of the "multiple market" critique of the economic conception of markets described by Zelizer (1988).

[338] The differing size of the diamonds in the fixed-role market diagram can represent the differing *status* of the sellers in typical production markets, in contrast with the standardization of buyers and sellers found in switch-role financial markets; see Aspers (2011).

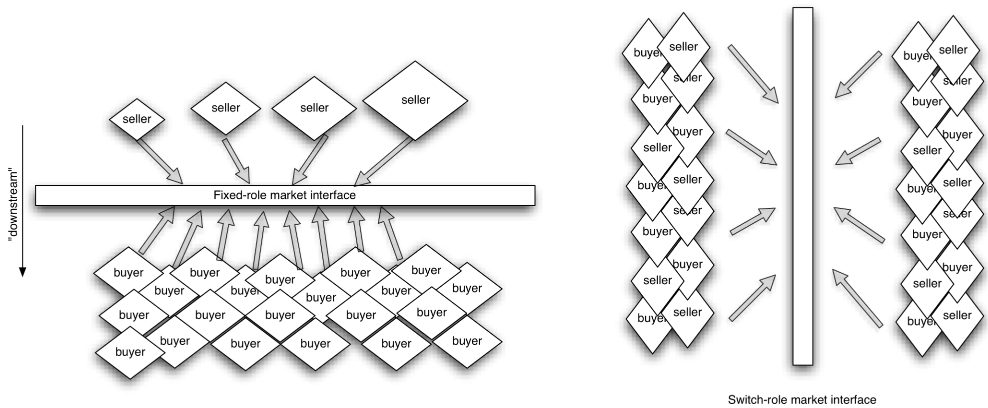

Figure 21. a) A fixed-role market. b) A switch-role market. Source: author.

While it was clear to Aspers that financial markets were obvious examples of the switch-role and standard market, neither Aspers nor many other economic sociologists were, until recently, particularly concerned with the *stock exchange* itself in its role as a *firm*, a structured institution without which those financial markets would not exist.[339] If one considers the stock exchange as an organization which can be in competition with other organizations—as in the case earlier in the 20th century, between the New York Stock Exchange (NYSE) and regional exchanges like Philadelphia's at which one could trade NYSE-listed securities—one can see exchanges as sellers in a *fixed-role* market for trading services (concisely, a "market for liquidity"[340]), where the "buyers" of those trading services are various individual and institutional traders, buying and selling stock on the platforms produced by the exchanges (and mediated by the exchange's authorized brokerage firms and/or dealers); see Fig. 22. Exchanges,

---

[339] A newer article (Ahrne, Aspers, and Brunsson (2015)) does point out that exchanges "usually take the form of associations or firms" and contrasts this with contemporary economists' assumption that markets can appear spontaneously. Works focusing on the Paris Bourse as a firm and/or institution include Hautcoeur and Riva (2012) and Lagneau-Ymonet and Riva (2015), but the history of inter-exchange competition there is less extensive than in the U.S. cases.

[340] Friess and Greenaway (2006, 162).

then, are themselves in fact producers; and what they produce are market platforms to match buyers and sellers of various securities. In brief, *an exchange industry is a fixed-role market that produces switch-role markets*.[341] And just as Aspers (2007, 379) insisted that "no existing theory can be used to explain both [fixed-role and switch-role markets]", one can often find in non-specialist discussions of stock exchanges certain basic terms (such as "market" and "competition") being interchangeably applied to both the fixed-role market competition (for trading services, between exchanges) and switch-role market competition (between buyers and sellers of a given stock to transact at a favorable price).

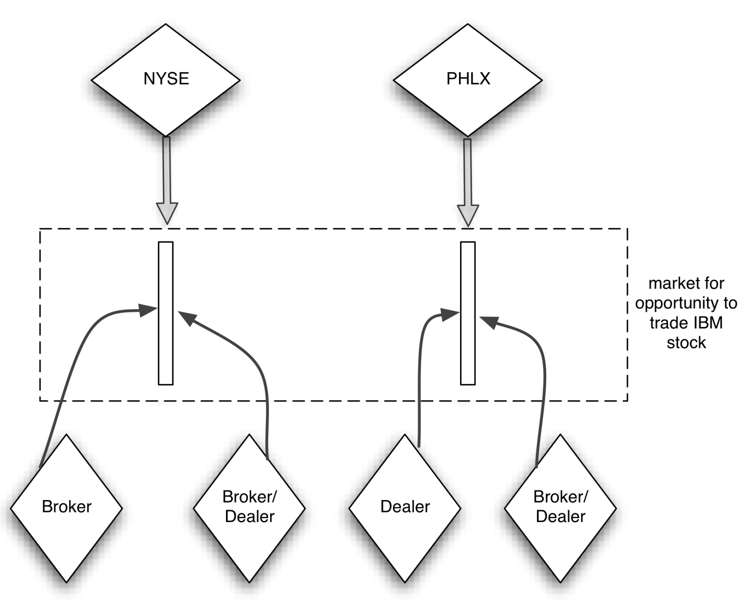

Figure 22: Producers of trading services in the 1980s for IBM stock include the NYSE and the regional Philadelphia Stock Exchange (PHLX). Brokers and dealers are "in the market" for the exchanges' services, which consist of switch-role markets in which they can alternately buy and sell IBM stock. Source: author.

[341] In this formulation, the products of an exchange are services—specifically, "trading services", a term not infrequently used in more specialized literature to describe what exchanges produce; e.g., Schwartz & Francioni (2004, pp. 133–135).

The trading facilitation services that these exchanges produce, in turn, take the form of multiple (switch-role) financial markets for individual securities. Until now, the social studies of finance (SSF) literature has focused on largely these latter switch-role markets—as in the ethnographies of trading floors (Baker 1984), trading screens (Cetina and Bruegger 2002b) and investment-bank trading desks (Beunza and Stark 2012b)—but paid little attention to the institutional conditions that create and maintain them.[342]

Missing from these accounts is the (fixed) role of exchanges as institutions which can compete to attract these trading agents. This focus on switch-role markets in SSF is in contrast to Harrison White's emphatic focus on fixed-role production markets in his economic-sociological theory. White's view, put succinctly, is: "A producer's market organizes producers into an array of parallel roles whose primary focus is each other" (H. White and Eccles 1987); this asymmetric logic is quite different from the structural similarity of buyers and sellers in a financial market.[343]

In part, this lack of recognition might be attributed to an assumption that White's theory of production markets should only be applied to goods, and not services. The classical distinction between goods and services, which goes back to Adam Smith, is worthy of continued consideration in economic sociology. Callon, Méadel, & Rabeharisoa (2002), for example, forcefully suggest that we should see in discussions of the rise of the service economy a

---

[342]The subdomain within economics focusing on fixed-role markets is that of *industrial organization* (IO) (Schmalensee and Willig 1989). Some of the notions from contemporary industrial organization, such as *multi-sided markets* (Rochet and Tirole 2006) are quite suggestive and can permit a good degree of theoretical complexity (despite their canonical examples including somewhat imaginary entities, like now-nonexistent "singles bars").

[343]White has outlined and elaborated on this idea in many articles, beginning with White (1981a) and White (1981b) and culminating with the monograph *Markets From Networks: Socioeconomic Models of Production* (H. White 2002). Intermediary presentations on similar material include White & Leifer, (1988), White (1988), and White (1992). White's explicit influences from economics are manifestly not neoclassical theorists like Walras, but instead include Chamberlin on monopolistic competition (E. Chamberlin 1933) and the signaling theory of Michael Spence; On Chamberlin, see Swedberg (2003, pp. 113–114).

"profound transformation of the rules by which markets function".[344] In our case, in one "market" (a financial market for a given security) we have the furious turnover of symbolic property rights and a form of competition which is (theoretically) solely a function of price; in the other "market" (the exchange industry) we have, among competing exchanges, the much less cleanly demarcated competition for the provision of trading services—a "product" which is hardly uniform. Because examples of production markets in economics lean toward straightforward examples using standardized goods, it may be less obvious that exchanges also form a production market, albeit a semiotically and interactionally complex one: namely, their product is the facilitation of the continuous exchange of goods which—in the case of securities—are so standardized as to be represented by certificates in a centralized clearinghouse, or indeed nothing more than symbols in a computerized database.[345]

In the next two sections, I shall examine the fixed-role and switch-role aspects of exchanges in turn, emphasizing the sociotechnical and technopolitical aspects of each. By *sociotechnical* I aim to highlight a greater sensitivity to issues of technologies and techniques with respect to phenomena largely understood with technological factors *in absentia* (such as the notion of *embeddedness*).[346] By *technopolitical*, I want to fuse the sense of technology as *volition*

---

[344]Callon, Méadel, & Rabeharisoa (2002, p. 196). Gadrey (2000) describes theoretical progress in the goods/services dichotomy, including those of Peter Hill, who points out the traditional weaknesses of neoclassical economics in the study of services: "Because services cannot be transferred from one economic unit to another, models of pure exchange economics of a Walrasian type in which existing goods are traded between economic units are quite inapplicable and irrelevant to services" (Hill 1977, 318).

[345](Callon & Muniesa 2005).

[346]The term 'sociotechnical' is analogous to the sense of *sociomateriality* Orlikowski/Scott (2008), but we intends to not privilege any of the senses of technology-as-tool, technology-as-technique, technology-as-social, and technology-as-volition, as in Mitcham (1994).

(Mitcham 1994) with a definition of politics as *intentional institutional change* (Glaeser 2010) to suggest that there can be no politics absent of sociotechnics, and vice versa.[347]

Fixed-role markets in exchanges: the provision of trading services

In the consideration of the exchange as part of a production market I take as our unit of analysis the exchange as firm, as in the tradition deriving from Coase (1937), and see producer firms as intrinsically involved in multiple markets—the *upstream* markets of which they are buyers, and the *downstream* markets of which they are sellers. Since in our case the exchange is a producer of trading services, the immediate "downstream" consumers of these trading services are brokerage firms, who in turn can be seen as providing those trading services further downstream to institutional and retail investors.[348] The crucial role of the state in affecting the arrangements of firms in a production market goes unmentioned by White (1981b), but is asserted forcefully by Fligstein (1996). Indeed, for Fligstein, stable production markets are something that occurs not despite, but because of, explicit intervention on behalf of the state.[349] In our case, however, we can consider neither production markets nor the state regulation thereof as occurring independently of their sociotechnical arrangements. Competition in the provision of trading services, as we shall see, is dependent on the technological relations between exchanges; and regulatory change can be enacted both in response to these technical relations, and to intentionally induce these technical relations.

---

[347]Gabrielle Hecht and Paul Edwards use the term 'technopolitics' to refer precisely to such a "hybrid form of power" with "cultural, institutional, and technological dimensions" (Hecht and Edwards 2010).

[348] Some studies in finance research that see the exchange industry in this way, taking an industrial-organization perspective, include Macey & Kanda (1989), Domowitz and Steil (1999), and Cantillon and Yin (2011).

[349]Dobbin (1994) and Fourcade (2009, 36–37) argue that the regulatory tradition in the U.S. (going back to the 1891 Sherman Antitrust Act) normalized oligopolies as inherently "competitive" within legal discourse. It should also be noted that Aspers criticizes Fligstein for only considering the role of the state in production as opposed to financial markets (Aspers 2009).

While there are sociologists (Muniesa (2000), Pardo-Guerra (2011)) who have focused on the history of particular exchanges (the Paris Bourse and the London Stock Exchange, respectively) as technological institutions, I argue that there is a great degree of opportunity for the field of economic sociology to attempt to address topics which, for economists—and its subsequent market microstructure literature—are considered "puzzles" in the context of financial exchanges. Economic depictions of the history of stock exchanges, for example, often provoke the phrase "liquidity attracts liquidity"[350], which is to say that whichever exchange at any given time has attracted the most customers for a given security may remain incumbent due to a "network externality".[351] This fact highlights why notions of "embeddedness" (Granovetter 1985)) were so eagerly applied to finance. For example, take the "network externality puzzle" discussed in the survey of Madhavan (2000), which asks why financial markets remain "fragmented" (as in, multiple exchanges compete to provide markets in the same security). How, indeed, do financial markets become fragmented in this way?

Switch-role markets in exchanges: the trading of securities

In contrast to the fixed-role markets in which exchanges compete with each other, exchanges themselves *produce* switch-role markets: an investor interacts (directly or indirectly) with an exchange in order to gain access to arenas of buying and selling of securities. These markets use the "continuous double auction" system of financial markets, which today dominates securities and derivatives exchanges worldwide: in between the "call auctions" which open and

---

[350] "Markets consolidate because traders attract traders. Trading is easiest and cheapest where most traders of an instrument or similar instruments trade. Liquidity attracts liquidity" (Harris 2003, 539).

[351] "As the value to one trader of transacting on a given trading system increases when another trader chooses to transact there as well, such a system is said to exhibit network effects or network externalities" (Domowitz and Steil 1999).

close an exchange, orders to buy and sell may be posted at any time in a continuous fashion.[352] Because each market for a given stock is switch-role, a buyer of a stock can become the seller of that stock immediately afterwards (and vice-versa, in the case of short-selling). (Today, this temporal window within which a trader—or trading *agencement*, as per Çalıskan and Callon (2010)—may buy and sell a quantity of stock has today been reduced to a matter of microseconds.) Because the goods being bought and sold in a market for, e.g., a given stock are homogenous and strictly delimited, they have historically posed as an exemplary representative system for the general equilibrium theory of the 19th century French economist Léon Walras, which modeled buyers and sellers' continuous interests (mediated by an auctioneer in a so-called *tâtonnement* process with zero transaction costs) to uncover an (presumed) underlying price.[353] And because the assumptions of the general equilibrium theory happen to be isomorphic to an idealized version of financial markets devoid of (in the economists' nomenclature) 'network effects', 'imperfect information', and 'trader heterogeneity', these situations are obvious ground for empirical disputation of microeconomic assumptions.

By contrast with the example of fixed-role markets, the sociotechnical dimension of financial markets has been carefully examined by a number of researchers, including Knorr Cetina & Bruegger (2002) and Zwick & Dholakia (2006). However, only recently has the social studies of finance field directly considered the technological and computational implementation of financial markets, as in the discussion of the Island exchange platform in MacKenzie &

---

[352]See Friedman (1993). The term *continuous auction* can thus be contrasted to the *call auction*, in which orders are aggregated and then later matched at periodic, pre-arranged times. For a classification of exchange trading systems based on empirical observation in the mid-1980s, see Cohen et al. (1986, 16–37).

[353]It is sometimes stated that Walras' original model was designed on the actual call auction process of the late-19th century Paris Bourse (Walras states: "let us go into the stock exchange of a large investment centre like Paris or London" (Walras 1954). That the Paris Bourse ever functioned in a manner similar to Walrasian *tâtonnement* is disputed by Walker (2001).

Pardo-Guerra (2014), and in my Chapters 3 and 4 here. Electronic exchanges, at their core, (1) automatically perform the matching of orders and execution of trades to exchange some symbolic entity—for which some kind of on-line transaction processing (OLTP) system, as described in Chapter 3, is necessary; and (2) report and/or broadcast data about currently available quotes and executed trades—for which some kind of multicast message-oriented middleware (MOM) system, as described in Chapter 4, is necessary. With sufficient hardware (disk space, networking and communications, memory) backing up such a functioning order-matching and data communication system, they can be extended to perform simultaneous matching in multiple contracts; and electronic derivatives exchanges, with (for example) a variety of expiration dates and strike prices, benefit strongly from this digitized facility for increased scope. In the late 1990s, observers noted the broad significance of these affordances:

> Automated systems can now be tailored quickly and inexpensively to accommodate trading in a growing number of securitized products, such as equities, bonds, currencies, financial derivatives, pooled mortgages, agricultural commodities, electricity, pollution emission permits, and hospital bed allocations (Domowitz and Steil 1999, 46).

And while the role of the state has not gone underaddressed in discussions of financialization processes (as in, e.g., Krippner (2012) and Pacewicz (2013), the specifically technological aspects of the politics of financial markets are a currently developing field (Pardo-Guerra and MacKenzie 2014).

	FIXED-ROLE/ PRODUCTION MARKETS	SWITCH-ROLE/ FINANCIAL MARKETS
Microeconomics	Coase (1937), Chamberlin (1933), Schumpeter (1942), Schmalensee and Willig (1989)	Walras (1954 [1892]), Demsetz (1968), Madhavan (2000), Hasbrouck (2007)
Embeddedness	White (1981), Granovetter (1985), Uzzi (1997)	Baker (1984)
Politics	Fligstein (1996)	Carruthers (1996)
Sociotechnics/ Technopolitics	—	Knorr Cetina & Bruegger (2002), Beunza and Stark (2012); Pardo-Guerra and MacKenzie (2014)

Table 8: Comparison of studies on fixed-role markets and switch-role financial markets.

Setting the stage: the last days of the club

Let us consider the New York Stock Exchange (NYSE) in the early 1960s, then a member-owned, non-profit cooperative,[354] As correctly noted in the Securities and Exchange Commission's *Report of Special Studies of the Securities Markets*, the term 'securities markets'—both at that time, and today—"encompasses both the markets for distribution of securities into public hands and the markets for continuous trading in outstanding securities" (SEC 1963a, 9); the former refers to the issuing (via an underwriting investment bank) of an initial public offering (IPO) of stock for a newly public firm; and the latter refers to the financial markets of which this chapter is explicitly concerned.[355]

"The NYSE" was thus in actuality a surfeit of separate switch-role markets, one for each listed security, with a variety of intermediating actors (in this case, the primary intermediates

[354]The period of transition before the end of fixed commissions in 1975 is well-documented in Welles (1975).

[355] These are referred to as "primary" and "secondary" markets in securities, respectively (Harris 2003, 209–10).

were the 600+ brokerage firms which were then members of the NYSE).[356] The custom at the time between the NYSE and the next largest exchange, the American Stock Exchange (Amex), would be for the latter to list smaller companies; once they were "battle-tested", they could de-list from Amex and list on the NYSE.[357] Meanwhile, because NYSE-listed stocks were not traded on any other major (non-regional) exchange, what we might now consider "competition" in these securities markets was less present; each exchange thus had an effective monopoly in providing trading services for a given stock.[358]

The 1963 *Special Study* was also significant in its early discussion of the possibility for automation; while the discussion of the automation of order matching and trade execution was highly speculative, but there was more interest in integrating various reports (including the exchange tape) to provide "a continuing, comprehensive market picture."[359] As part of the study, the SEC also commissioned a study by the Univac computer-manufacturing division of Sperry Rand, which concluded that "one centrally located computer would have sufficient capacity, speed, and capability to accommodate the reporting of the listed markets as well as the over-the-counter market."[360]

---

[356]These intermediating firms are called the *sell-side*; one can think of them as intermediating between traders and/or their representatives (a.k.a. the *buy-side*) and the exchange itself. This is to say, it is the *trading services* that the buy-side is buying and the sell-side is selling, *not* the securities themselves. Also note that this perspective of the exchange's products as a set of independent markets is a simplification; various factors (including prohibitions and fees) may encourage investor diversification within an exchange's markets as opposed to across them.

[357]Seligman (1985, p. 7) describes the Amex as a "minor league" to the "major league" NYSE.

[358]Coffee (2002, pp. 1769–1770). There was also an array of independent dealer markets for trading securities; these "over-the-counter" (OTC) markets were also known (in aggregate) as the "third market". Additionally, Rule 394 (later Rule 390) prevented NYSE members from effecting trades in the over-the-counter market (the dealer markets regulated by NASD) (Seligman 1995, 387–88).

[359]SEC (1963, pp. 354–355).

[360]"Listed markets" refers to financial markets hosted by the exchange (e.g. NYSE, Amex) on which a stock first made its IPO. "Over-the-counter" refers to the trading of these and other stocks in settings not hosted by a formal

The transition away from floor-based trading was also sown by the Paperwork Crisis of the late 1960s, described in Chapter 3, in which a steady rise in trading volume—led by increased trading on the part of institutional investors for mutual funds and pension funds—crippled the clearing and settlement "back offices" of NYSE member brokerages, leading to waves of mergers and departures of over a hundred firms from the exchange.[361] A subsequent investigation by the SEC (SEC 1971a) led to a deliberate centralization of securities and the formation of the Depository Trust Company (DTC) in 1973, and the centralization of clearing and settlement services in the form of the Securities Industry Automation Corporation (SIAC).[362]

Centralized quotations and automated execution: NASDAQ and Instinet

By 1971, NASDAQ—the automated quotation system of the National Association of Securities Dealers—was operational, linking hundreds of market-makers to a pair of Univac 1108 mainframes in Trumbull, CT.[363] NASDAQ did not provide for automated trade execution, but it did provide a centralized, electronic repository of extant dealer quotations. Institutional Networks (later Instinet), by contrast, was a registered broker-dealer with institutional investor subscribers (e.g. pension funds and mutual funds) with dedicated lines to another Univac system in Watertown, MA. Unlike other electronic systems of the early 1970s, Instinet provided the facility for automated execution of anonymous block trades.[364]

---

exchange. SEC (1963, p. 657). Sperry Rand was then one of the "seven dwarves" of computer manufacturing in competition with IBM.

[361]Wells (2000); NYSE (1971).

[362]Keith & Grody (1988).

[363]NASD was the self-regulatory organization (SRO) for over-the-counter (OTC) broker-dealers (J. W. Smith, Selway III, and McCormick 1998).

[364]A "block trade" is simply a large transaction—at least 10,000 shares, but often much more. On the founding of Instinet, see Pardo-Guerra (2013).

In hearings before the House Subcommittee on Commerce and Finance (U.S. House 1972) and a subsequent Senate report (U.S. Senate 1973), the electronically centralized quotations of NASDAQ were taken in part as an inspiration for a proposed "central market system" (later "national market system" or NMS)[365]:

> While the various formulations of the concept [of a central market system] differ in important respects, they have all contemplated the existence of a communication system through which (1) all orders and quotations in a particular security would have an opportunity to meet, and (2) all transactions would be reported (U.S. Senate 1973, 89).

In 1975, Congress passed the Securities Amendments Act of 1975 (U.S. 94th Congress 1975). The act, among other changes, ended the fixed commissions of NYSE members and directed the SEC to establish a National Market System, although details on how such a system was to be implemented were vague.[366] It called for "fair competition among brokers and dealers, among exchange markets, and between exchange markets and markets other than exchange markets".[367]

The existing centralized quotation systems—albeit only used on over-the-counter stock—thus made it possible to imagine a National Market System as a centralized limit order book (CLOB) (Pardo-Guerra and MacKenzie 2014). The National Market System amendment introduced rules to facilitate the construction of an NMS, including the "Last Sale Rule"[368],

---

[365]"We… note our satisfaction with the manner in which the NASDAQ communications system has been operating and intend to continue to monitor its operations and development in order to determine whether any modifications may be necessary as the evolution of a central market system progresses." (U.S. House 1972, 3447–48)

[366]Macey & Haddock (1985).

[367]The amendment relating to the National Market System is section 11A (U.S. 94th Congress 1975, 111–112).

[368]The Last Sale Rule (originally rule 17a-15 in SEC Release 34-9850 in 1972) required the dissemination of trade execution information in exchange-listed and NASDAQ stocks on some real-time reporting system. (The "last sale" is the last transaction price for a security, on any market.)

"Quote Rule" (or "Firm Quote Rule")[369], and "Display Rule".[370] Technological developments subsequent to the 1975 Securities Acts Amendments include the establishment of the Consolidated Tape Association (CTA) (to implement the Last Sale Rule)[371], the Consolidated Quote System (CQS) (to implement the Quote Rule[372]; and the Intermarket Trading System (ITS), which allowed orders placed on the NYSE to be executed on a regional exchange (via networked "chat room"-style terminals).

Electronic trading platforms in the 1990s

An exchange had been defined in the Securities Exchange Act of 1934 in the following way:

> The term "exchange" means any organization, association, or group of persons, whether incorporated or unincorporated, which constitutes, maintains, or provides a market place or facilities for *bringing together purchasers and sellers of securities* or for other wise performing with respect to securities *the functions commonly performed by a stock exchange as that term is generally understood*, and includes the market place and the market facilities maintained by such exchange [emphasis added] (U.S. 73rd Congress 1934, sec. 3.(a)(1)).

Institutions registered as exchanges are classified as *self-regulatory organizations* (SROs), which are obliged by the 1975 Amendments to the Securities Exchange Act to enforce a variety of conditions, to "prevent fraudulent and manipulative acts and practices", to "promote just and equitable principles of trade".[373]

---

[369]The Quote Rule is 240.11Ac1-1, "Dissemination of Quotations". It required brokers/dealers to send its quotes to exchanges, and for those exchanges to make those quotes available.

[370](Lee 1998, pp. 124–126). The Display Rule is 240.11Ac1-2.

[371]Before the Consolidated Tape, information on the last-sale price was provided by NYSE or Amex ticker tapes or electronic displays (Seligman 1984, 86).

[372](Lee 1998, 126).

[373]U.S. 94th Congress (1975, pp. 105–106); Lee (1998, pp. 118–120).

By contrast, a *broker* and *dealer* were defined this way:

> The term "broker" means *any person engaged in the business of effecting transactions in securities for the account of others*, but does not include a bank [emphasis added] (U.S. 73rd Congress 1934, sec. 3.(a)(4)).
>
> The term "dealer" means *any person engaged in the business of buying and selling securities for his own account*, through a broker or otherwise, but does not include a bank, or any person insofar as he buys or sells securities for his own account, either individually or in some fiduciary capacity, but not as a part of a regular business [emphasis added] (U.S. 73rd Congress 1934, sec. 3.(a)(5)).

The distinction between the "exchange" and the "broker" were established in a world where the latter was strictly subjugated to the former. That is to say, brokers needed the exchange to provide them with opportunities for finding counterparties to their trades. Additionally, brokers were subject to the rules and regulations of the exchange. Thus, when these terms were defined, there was never an assumption that any individual broker or broker-dealer might be providing "the functions commonly performed by a stock exchange as that term is generally understood". But by the late 1980s, this was precisely what Instinet had been doing for decades (see Fig. 23).

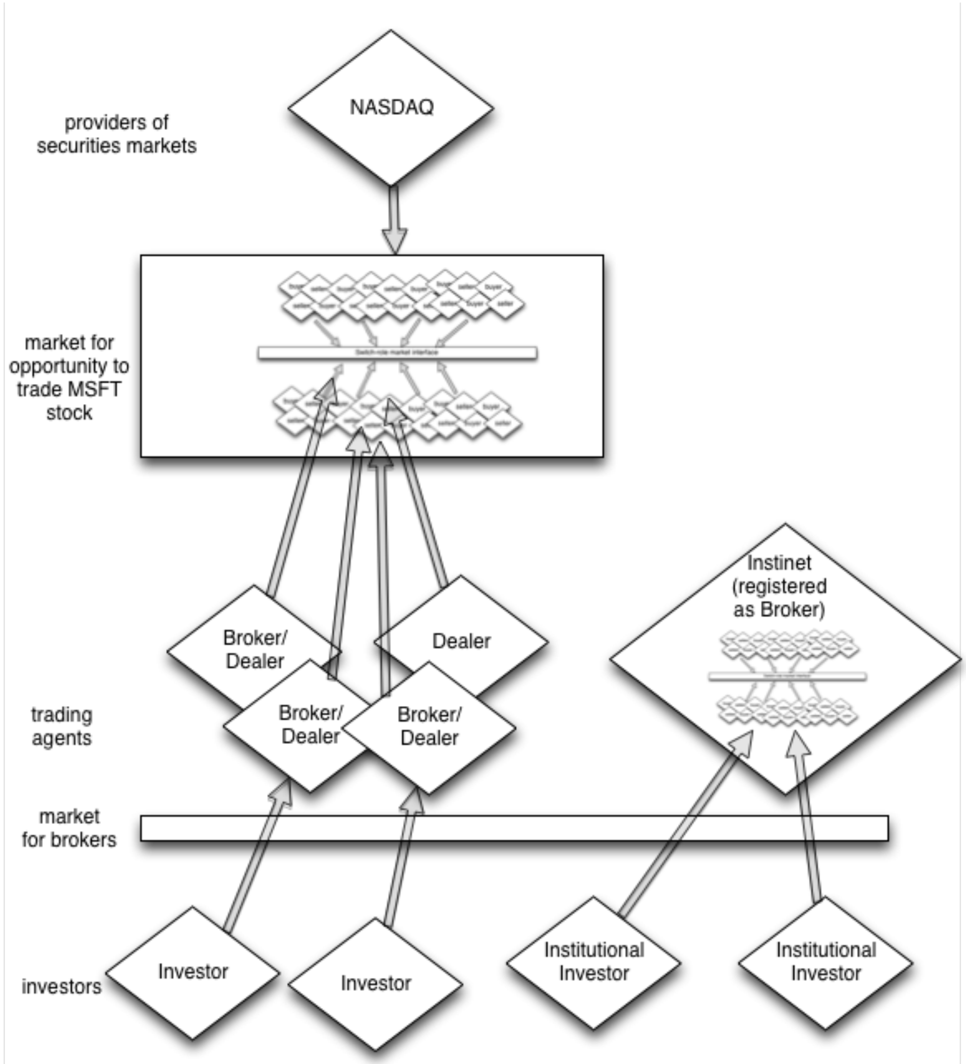

Figure 23: Instinet disrupts the market for financial markets by being registered as a broker, but functioning as an exchange. Source: author.

In the section to follow, we shall examine how the dissolution in the distinction between an exchange and a broker-dealer was, in part, the outcome of technological changes. But automated trade execution platforms had made only a limited impact on the exchange landscape, until a distinct political development—the NASDAQ odd-eighths scandal, described below—motivated the further elaboration of NMS-related regulations (the 1996 Order Handling Rules); these regulations in turn legitimated a variety of competing broker-dealer systems, known as *electronic communication networks* (ECNs). In response to the emergence of the ECNs, the SEC ultimately passed a resolution in 1998, Regulation ATS (for "Automated Trading Systems"), which finally permitted ECNs to choose to be regulated as either exchanges or as broker-dealers, and thereby redrawing the demarcation lines between broker and exchange.

The regulation of a disrupted production market

In 1991, SEC Chairman Richard C. Breeden announced the commencement of a "thorough and comprehensive study of the current market structure", entitled "Market 2000".[374] The subsequent SEC request for comments stated that the SEC believed that "computerized trading systems, whether operated by securities markets or by broker-dealers, are generally consistent with the objective of linking all securities markets through communication and data processing facilities."[375] Another document, co-written by members of the SEC's Division of Market Regulation, admitted the inevitability of such systems, but raised concern:

> …the rate of technological change has become so great that other, equally revolutionary developments seem to follow in almost stupefying rapidity. Thus, we find ourselves attempting to make difficult choices concerning what time and place limitations we will choose to retain, if any, in the absence of any lingering physical or technological necessity, all the while being bombarded by continuing automation advances that sometimes make even our most recent market structure and regulatory decisions seem already archaic (Becker et al. 1992, 328).

It is important to explain why the SEC seems to be ambivalent about a transformation which might be considered consistent with an NMS initiative that, at that point, was over 16 years old; I will do so by simultaneously emphasizing sociotechnical and technopolitical perspectives.

---

[374] Breeden (1991).

[375] SEC (1992, p. 32601).

A *sociotechnical* understanding would emphasize the presence of actors and their associated technologies and techniques as asymmetric prostheses. For example, the innovations by Instinet and other ECNs were definitively interconnected to practitioners and technologies from outside the financial industry; For example, the founders of Instinet (Weeden & Co.) did so not because of an internally developed matching system, but because they had also funded Keydata Corporation in Watertown, MA, which provided time-sharing computing services (founded by Charles Adams, a member of MIT's real-time Project Whirlwind) (Pardo-Guerra 2014).

A *technopolitical* perspective would recognize the (currently understudied) role of relevant patents on the part of Charles Adams and others.[376] But it would also account for the relative ignorance towards technology on the part of the SEC as an organization and institution, historically primarily composed as it is of securities lawyers without formal training in engineering or computer science fields.[377] These 1991 and 1992 discussions followed in the wake of an earlier SEC proposal in 1989—on which the SEC ultimately did not take action—which floated the concept of regulating "proprietary trading systems" like those of Instinet (SEC 1989). The SEC comment letters reveal a strong preference on the part of incumbent exchanges for regulation, and an equally strong preference on the part of the firms running the proprietary systems to remain registered as, e.g., broker-dealers. While these discussions remain at a

---

[376]Adams' 1969 patent is "Instinet communication system for effectuating the sale or exchange of fungible properties between subscribers", US3573747 A.

[377]Khademian (1992).

theoretical and legalistic level and do not actively discuss the materiality of their systems, their positions emphatically indicate the role of these technological systems in the debates to come.[378]

Market 2000: Study of U.S. Equity Market Structure

Noting the technological challenge to their existing regulatory definitions, in July 1992 the SEC released a request for comments on the ongoing study to U.S. equity market structure.[379] In order to frame the parameters of the transformation of the exchange, I will enumerate the most important—and, perhaps, problematic—concepts mentioned in this document release, including:

> *Best execution*: It appears to be assumed that greater transparency (see below) and a "linked market" will lead to better execution.
>
> *Transparency*: This involves the "real-time" dissemination of quotations and trade information.
>
> *Market fragmentation*: the idea that markets are "two-tiered" — one for institutional investors and one for individual investors—is raised.
>
> *Competition*: The document explicitly asks, "is 'fragmentation' simply another word for 'competition'?" (SEC 1992, 32395)

---

[378] By contrast with the above perspectives, it is worth noting the relative weakness of the concept of *performativity of economics* in the case of the transformation of the exchange industry. In the construction of automated quotation and trade execution systems there is little neoclassical economic theory to be found, despite the (incorrect) possibility of imagining these systems as physical manifestations of a hypothetical Walrasian-equilibrium generator (this is to say that, in practice, continuous order matching via a CLOB does not correspond with Walras' depiction.) In fact, Frederick Nymeyer, who submitted a CLOB-style patent around the same time as Smith, was inspired by Austrian economics, which denied the existence of a single market-clearing price (Pardo-Guerra 2014, 22). Moreover, one finds little theory of industrial organization cited in the regulatory debates, besides the abstract invocation of notions of competition and fairness.

[379]SEC (1992)

Liquidity: It is also held that the dispersal of order flow in the situation of fragmentation may "impair liquidity".

Each of these concepts can be considered in turn. "Best execution" is defined most generally as traders receiving favorable outcomes for their trades; in securities law discussions, brokerages are obliged to execute a customer's order at the best available price, though there is no existing definitive statement of what constitutes best execution (J. R. Macey and O'Hara 1997, 190). However, when multiple trading venues are available with different bid-ask spreads, parameters and commissions for trade execution, it is not always clear what constitutes the most favorable trade. For example, one reason held for the moderate success of Instinet and POSIT (another platform for institutional traders) in an era dominated by the incumbent NYSE is that institutional investors could execute large trades while reducing the "price impact" or "market impact"—i.e. the financial market's dynamic response to the elements of *phatic* communication in the act of trading[380]—that such trades would have on the public exchanges. As Larry Harris put it, "Best execution means different things for different people."[381]

Transparency is a word that often indicates a philosophical tendency towards a single, accessible consolidated limit order book (CLOB). As such it represents a comparable paradox to that of "best execution", which is that some traders will be discouraged from the "transparent" exposure of their limit orders. However, it is clear that a lack of transparency on the part of

---

[380] On phatic communication, see Jakobson (1960). In the 2000s, the competitive proliferation of "maker-taker" pricing—which grants various rebates to either "liquidity suppliers" (those "makers" posting marketable limit orders) or, alternatively, to those "takers" submitting the orders which match them—further complicated this notion of best execution (Foucault 2012).

[381] Harris (1996).

market-makers has led to excess spreads and high commissions in some exchanges.[382] One can imagine the sociologically appropriate position to take with respect to transparency is one of ontological heterogeneity, not just of traders (as in the case of "best execution") but of firms in competition with one another. With complete order book transparency, there is little one can do to distinguish oneself as an exchange except to compete on execution speed. But the success of contemporary "dark pools" helps show that transparency is not always a positive feature for traders and exchanges, and that the population of trading services firms in a "fragmented" environment is likely to always include producers of both 'lit' and 'dark' financial markets.

Market fragmentation is an especially slippery phrase, with an inherent pejorative sense for many, and for which my introductory distinction between fixed-role and switch-role markets can be applied. Fragmentation at the level of the exchange industry would seem to be a good thing for those who want to improve competition (as opposed to the monopolistic qualities of the NYSE in the 20th century, for example.) The market microstructure literature refers to the basic fragmentation of "upstairs" trading (executing large blocks in a dealer market as opposed to the NYSE floor) as "rational fragmentation", as it is used to reduce the price impact of large trades.[383] But fragmentation at the level of the switch-role financial market—where the confluence of more buyers and traders results in the "positive externality" of the best prices—it would seem that fragmentation is problematic at best.

Competition is a concept which is unavoidable with respect to switch-role markets but, as per White, somewhat different for fixed-role markets, as one rarely finds a state of "pure

---

[382] The exemplary case of this was, of course, the NASDAQ odd-eighths scandal (Christie and Schultz 1994).

[383]Madhavan (2000, p. 227).

competition" in the analysis of production markets. Some commentators are, indeed, thoroughly aware that competition in switch-role and fixed-role markets must be keenly distinguished:

> The competition among traders to obtain the best price and the competition among exchange service providers to provide exchange services often are incompatible with each other. Policies that would improve one competition typically harm the other. The pro-competitive position on any issue affecting both competitions—which includes most issues—therefore is rarely unambiguous (Harris 2010, 106).

Finally, *liquidity* —referring to the presence of sufficient market interest to be able to transact large amounts of a given security at reasonable prices in a short time frame— is a fascinating category, especially in the modern-day context of high-frequency trading, where debates emerge over whether HFTs are "providing/offering liquidity" or whether they are "taking liquidity".[384] An important aspect here is the facility for high-frequency algorithms to post and then quickly retract limit orders as they became unfavorable due to market conditions elsewhere (Dolgopolov 2014).

National Market Hearings (1993)

Subsequent to the 1992 request for comments, in the Spring and Summer of 1993, the House Committee on Telecommunications and Finance held a series of hearings (U.S. House, 103rd Congress 1993) focusing on the "Market 2000" initiative, inviting representatives from many exchanges and other industry institutions to give remarks and respond to Congressional questions; this included the Presidents and Chairmen of the NYSE, Amex, NASD, various regional exchanges, and various firms engaged in proprietary trading systems (including Instinet, Lattice, ITG, Madoff, and the Arizona Stock Exchange (AZX)[385]).

---

[384]Harris (1991) is an excellent discussion of liquidity.

[385]Steven Wunsch's Arizona Stock Exchange was, at the time, the only proprietary trade execution system actually registered as an exchange.

The published Market 2000 document (January 27, 1994) provides a snapshot of the U.S. securities exchange industry circa 1994. At that time, 97% of the market value for listed companies was at the NYSE, with the Amex and regionals at 3%. Half of NYSE volume were block transactions. Regional exchanges accounted for 20% of NYSE stock trades. The "third market" (OTC trading of NYSE-listed securities) accounted for 9.3% of trade volume; and proprietary trading systems had only 1.4% of NYSE share volume and 13% of NASDAQ share volume. [386]

While many of the actors speaking in the National Market Hearings were of high rank and though (testifying as they were before Congress) one cannot take their comments at face value, the discussions are particularly interesting, especially on contentious issues, and have helped us categorize the main classes of competitors in the market for trading services. Three of the issues are highlighted in Table 9: *fragmentation*, *payment for order flow*, and *regulatory burden*. The provided quotes intend to highlight the extent to which each category of dispute reveals the interests of the institutional actors in question.

---

[386] Securities and Exchange Commission (1994, pp. 7–9)

	Speaker	Exchange Type	Quote from National Market Hearings (1993)
Fragmentation	James R. Jones (Chairman, AMEX)	Incumbent	"…because SelectNet and other proprietary trading systems do not allow for widespread dissemination of trading interest, they result in increased fragmentation and reduced market transparency."
Fragmentation	Hardiman (President, NASDAQ)	Over-the-counter (OTC)	"…opponents of competition for order flow… must demonstrate that competition for order flow has led to palpable harm and that a monopolistic approach would lead to palpable improvement. We believe neither is possible."
Fragmentation	Bernie Madoff, (Chairman, Madoff Securities)	Broker-dealer / "Third market"	"By definition, any time more than one participant marketplace is involved in trading a particular security that could trade elsewhere, there is fragmentation…"
Payment for Order Flow	Donaldson (Chairman, NYSE)	Incumbent	"I think cash payments should be outlawed."
Payment for Order Flow	Hardiman (President, NASDAQ)	Over-the-counter (OTC)	"…the [discount/regional] firms that are receiving the payment for order flow are, indeed, charging lower commissions to their customers."
Payment for Order Flow	Bernie Madoff (Chairman, Madoff Securities)	Broker-dealer / "Third market"	"[T]he exchanges had offered numerous noncash inducements such as reciprocal order routing arrangements, clearing discounts, credits, and other free services… We found that one of the most effective ways of overcoming the primary exchange monopolies was payment for order flow."
Regulatory Burden	Leopold Korins (Chairman Pacific Stock Exchange (PSE))	Regional	"…The systems that have been developed…. should have to conform to the same type of SRO [self-regulatory organization] requirements that we as exchanges guard very jealously. And to establish entities that appear to be exchanges and operate like exchanges but don't have any of the obligations of exchanges, we think is an unfair burden upon us."
Regulatory Burden	Edward A. Kwalwasser, (Exec VP, NYSE)	Incumbent	"Before any trading system initially begins operation, there should be a thorough review of all aspects of the system and the system should meet certain investor protection standards."
Regulatory Burden	Michael O. Sanderson (President, Instinet Corp.)	Alternative Trading System (ATS)	"Regulation of Instinet as a broker is reasonable and appropriate. Regulation of Instinet's activities other than as a broker would discourage innovation in the securities industry."

Table 9. Arguments regarding fragmentation, payment for order flow, and regulatory burden in 1993 "National Market System" Hearings (U.S. House, 103rd Congress 1993)

The Order Handling Rules (1996): The Limit Order Display Rule and amended Quote Rule

Characteristic of the distinction between dealer-based markets (like NASDAQ) and those based on order matching (e.g. the NYSE) was the absence, in dealer markets, of public limit orders (K. J. Cohen et al. 1986, 19), even if there existed limit orders better than the current market-maker's quote for a security. A well-publicized study in 1994 (Christie and Schultz 1994) revealed the possibility of collusion on the part of NASDAQ dealers to keep quote spreads artificially wide (revealed in their data because the dealers' convention was to stick to even-eighths quotes and avoid odd-eighths quotes.) A subsequent release proposed that quotes be published openly whenever an exchange or market-maker trades more than 1% of a security's aggregate volume (SEC 1995).

Up until the adoption of these "Order Handling Rules" in September 1996, a NASDAQ broker-dealer would have no obligation to alter their quote in the system to reflect an incoming customer limit order.[387] The SEC had found the existence of a "two-tiered market" where market-makers would "routinely trade at one price with retail customers and at better prices with ECN subscribers", and insisted that "*all* investors" should be able to fill orders at the best offered price.[388] The Limit Order Display Rule required that customer limit orders better than a market-maker's quotes must be reflected in those quotes (or forwarded to another [entity] that will display the order).[389] The amendment to the Quote Rule includes the "ECN Amendment" which

[387]The proposed rules are SEC (1995); The final rules are in SEC (1996). The Limit Order Display Rule is Rule 11Ac1-4; the amended Quote Rule ("ECN Alternative" to "Dissemination of Quotations") is Rule 11Ac1-1.

[388]SEC (1996, p. 48308).

[389]Smith et al. (1998).

requires market-makers to publicly post any limit orders sent to ECNs which are better than the extant public quote.[390]

Interestingly, very few of the public comment letters supported this proposal without reservation, and even the ECNs (or the firms investing in future ECNs) had reservations about the new rules. One future ECN investor, Bear Stearns, instead proposed their own Limit Order Book technology (which would make that technology the valuable center of calculation instead of the quote-broadcasting complexities of the SEC proposal). Other ECNs like Instinet appear to have met only in private, with only brief summary memorandums available in the SEC's archives. Broker/dealers, looking forward to better prices for their customers, widely supported the proposal.

The order handling rules, once finalized, "brought the order-driven market into the quoted market" (Schwartz, Byrne, & Schnee (2013, p. 20)), meaning that they allowed ECNs to post orders in the NASDAQ quote montage, and potentially, fill it themselves (at a lower cost).[391] With this situation in place, ECNs were effectively no less powerful than NASDAQ dealers, and potentially more inexpensive for traders. (See Fig. 24.) The industry had changed overnight, and when anyone can run their own exchange with electronic access to the same buyers and sellers, one might ask: just what *did* it mean to be an exchange versus an ECN?[392]

---

[390]Odders-White (2004, pp. 280–281).

[391]Angel, Harris, & Spatt (2010, pp. 33–34). (The NYSE had a higher latency of placing and canceling orders.)

[392]For more on the effect of the Order Handling Rules, see Schwartz & Francioni (2004, pp. 229–230). According to Schwartz & Francioni (2004, p. 241), "A market maker could use a Nasdaq system (SelectNet) to send an order it has received to another market maker or to broadcast the order to all market makers. As quote providers, an ECN could also connect directly into SelectNet. SelectNet included a negotiation feature that allows a participant (market maker or ECN) to accept, reject, or counter a received order."

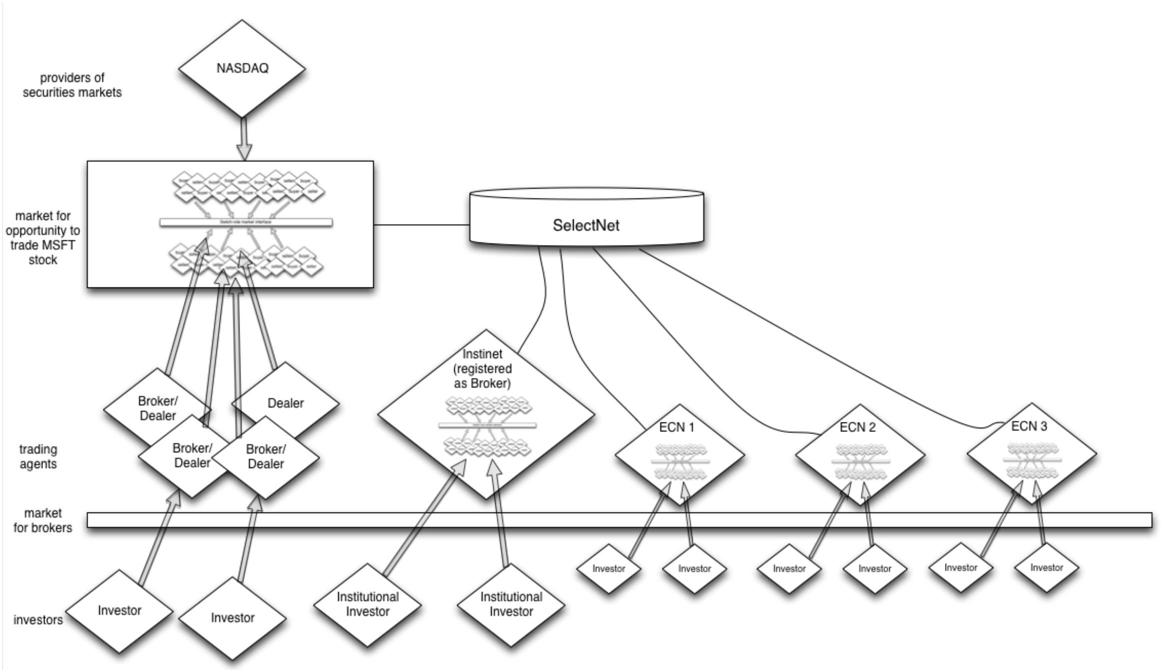

Figure 24: ECN disruption via direct access to NASDAQ quote montage. Source: author.

Regulation ATS (1998) and the semantics of the exchange

In 1997 the SEC issued a Concept Release for what came to be known as *Regulation ATS* (“Regulation of Exchanges” (SEC 1997)); after a comment period, the final rules were released in 1998 (SEC 1998). It provided a new definition of ‘exchange’:

> The statutory definition of “exchange” includes a “market place or facilities for bringing together purchasers and sellers of securities or for otherwise performing with respect to securities the functions commonly performed by a stock exchange.” The new rule interprets these terms to include *any organization, association, or group of persons that: (1) Brings together the orders of multiple buyers and sellers; and (2) uses established, non-discretionary methods (whether by providing a trading facility or by setting rules) under which such orders interact with each other, and the buyers and sellers entering such orders agree to the terms of a trade* [emphasis added] (SEC, 1998, p. 70848).

The primary discursive difference here is from a focus on bringing together *purchasers* over bringing together *orders*. This is not precisely a transformation in the *ontology* of the exchange, because floor-based trading is also characterized by a flow of such orders. However, it is a transformation in the (legal) *semantics* of the exchange: a move from seeing an exchange as a place where buyers and sellers of securities (or, more specifically, their agent intermediaries) are brought together to a place where orders (which may have a variety of origins) are brought together.[393]

Ultimately, as (Karmel 2002a, 89) describes, although the SEC did manage to redefine the "exchange" from its previous interpretations, the goal of Regulation ATS appears to be "to force ATSs with substantial volume in [National Market System] quotation and transaction reporting rules, [and] not to change the way in which exchanges operate or are governed." The transformation of the exchange was thus a legal construction which legally sanctioned a technological shift which had already occurred.

ECN	FOUNDED	ORIGINAL OWNERSHIP	OUTCOME
Instinet	1967	Institutional Networks	Sold to Reuters (1985); Merged with Island ECN (2002); Acquired by NASDAQ (2005)
Redibook	1992	Spear, Leeds & Kellogg; others	Merged into Archipelago, 2002
Tradebook	1996	Bloomberg	Still operating
Island	1997	Datek Online Holdings (majority)	Acquired by Instinet in 2002
Archipelago	1997	Terra Nova Trading	Sold to investors in 2000; Sold to Instinet in 2002, rewrote Instinet's matching engine.
BRUT (Brass Utility)	1998	Multiple firms; later SunGard Data Systems	Sold to Nasdaq (2004)
Strike	1998	Bear Stearns	Merged with BRUT (1999)

Table 10. Outcomes for ECNs in the 2000s. Some data from Liebenberg (2002, p. 77).

[393] The phrasing "non-discretionary methods", it is explained, is meant to distinguish matching algorithms from the activity at traditional block trading desks which would "shop around" and break up a customer order (SEC, 1998, 70851). For general remarks on Reg ATS, see Domowitz and Lee (2001).

Conclusion: the customer as competitor, and the valuation of marketplace platforms

> Over time, exchanges have been behaving more like intermediaries, and intermediaries have been behaving more like traditional exchanges (Cybo-Ottone, Noia, and Murgia 2000, 224).
>
> All natural economic distinctions between stock exchanges and broker dealers have broken down.... Exchanges and brokers are now doing exactly the same thing (Alpert 1999, quoting Benn Steil).

The above quotes indicate the situation at the end of the century: in an exchange industry which now obliged the exposure of orders and quotes, the very foundations of the former production market—in which exchanges would sell the facility to trade downstream to traders via intermediating brokers—had collapsed. The subsequent decade in the exchange industry was dramatic, including the rapid demutualization of major exchanges and waves of mergers (see Table 10 for an enumeration of the acquisitions and mergers of the ECNs of the late 1990s).[394] In 2002, another analysis by Benn Steil concluded:

> The inexorable trend toward securities exchanges operated as for-profit public companies with nonmember ownership is a direct product of the automation of trading systems (Steil 2002, 80).

Such a statement, *prima facie*, would appear to represent the quintessence of technopolitics—a political transformation which is seemingly inextricable from the technological. In this case a major industrial transformation has as its primary causal factor the implementation (and clones thereof) of an electronic version of a trading floor specialist's limit order book. While my work here does not examine the 21st-century exchange landscape, it is difficult to ignore the essential conceptual tension here between a unified, single (monopoly) network and the chaos that ensues when (as with the "National Market System" concept) a regulatory agency attempts to unify (fixed-role) providers of (switch-role) financial markets which, effectively, become fast-paced clones of each other.

[394] Domowitz (1995).

The exchange industry scholar Ruben Lee in 2002 predicted, given the many sources of income of an exchange (i.e., the multiple production markets for which the firm takes the role of a seller)—including "fees for listing, trading, clearing, and settlement, and charges for the provision of company news and for quote and trade data"—that the exchange industry had found itself in a similar position to the media industry (via digital distribution of content and increasingly online advertising marketplaces):

> In the language of the media industry, which they will effectively have joined, exchanges will be content providers. As such, they are likely to mimic the activities of other similar media companies (Lee 2002, 2).

Lee points out that as the marginal cost of executing a transaction gets close to zero, competition between exchanges will lead to increased *payment for order flow*, or "paying for the privilege of executing orders on their trading systems". This had indeed already begun, with Island's introduction of so-called "maker-taker" payments/fees, which gave a rebate to those "makers" submitting standing limit orders, and added a fee to "takers" executing market orders or marketable limit orders. Lee argued, correctly, that this would become the norm (Lee, 2002, pp. 1–2). His use of a media industry analogy is appropriate here, as so-called "two-sided platforms" like newspaper firms subsidize readers (by providing free or inexpensive news) at the expense of advertisers (Evans 2011). Another important remaining source of income, he suggests, would be quotation and sale data; and indeed, the income from these data feeds (as partially revealed in the newly demutualized firms' annual reports) became a prominent source of income for exchanges in today's fragmented, high-frequency markets.

Following the analysis and findings in this chapter, I suggest that a first step can be made towards a new way of thinking about the sociological study of markets. Specifically, by explicitly distinguishing the specialized, distinctive properties of switch-role financial markets from those of fixed-role markets, I have identified a potentially new field of economic processes worthy of investigation; one which is as intriguingly and processually intermingled with economic theory as before, but corresponding with the differing jargon of *multi-sided markets* and *two-sided platforms* from 21st-century industrial organization; and I shall address the potential application of this theory of markets to this latter domain in the concluding chapter.

But in general, this chapter points to an impending theoretical and policy-oriented dilemma. On the one hand, various industries are already confronting the rise of "marketplace platform" startups like Uber and Airbnb—which, like electronic exchanges, bring together buyers and sellers without any of the logistical concerns of materially-mediated supply-chain management. On the other hand, there exists the equally problematic alternative of intensive legal enforcement—in the name of competition and of securing some unified "national market system"—which would oblige competing firms to expose their customers' bids and offers, thus potentially leading to a fragmented production market of various services where firms ruthlessly compete for flows of orders without ever being able to maintain even temporary network dominance. Furthermore, what, in such a technopolitical environment, such as the one which developed in financial exchanges and is only now being realized elsewhere, may stop any customer from implementing their own matching engine, and thus becoming themselves a competitor? In the next and final chapter, I will address how this story of the transformation of the exchange industry can be intriguingly transposed to contemporary concerns of precisely this form, with respect to these marketplace platforms.

CHAPTER 6

THE PLATFORM AS EXCHANGE

Implications and Conclusions: From the Exchange to the Marketplace Platform

In the last chapter, I described the interweaving of technological and regulatory change during the 1990s in the United States, as the increasing technical facility for brokers (at first non-members) to effectively run their own order matching engines—as entirely new exchange-like systems known as *electronic communications networks*, or ECNs—coincided with the SEC's attempt to facilitate competition among the incumbent exchanges (Nasdaq and the NYSE). The decisions made in this period, including the 1996 Order Handling Rules, are in part responsible for certain distinctive aspects of today's exchange industry, an environment in which (for example) every NYSE-listed stock can be traded on many dozens of competing platforms, from public exchanges to dark pools; and which in the 21st century been beset by controversies involving high-frequency-trading (HFT) algorithms which perform arbitrage at high speeds between these competing exchanges. In this chapter, I will draw analogies between this view of the exchange as producer of switch-role market platforms and a different, but technologically and semiotically related, genre of platforms, those of the so-called "collaborative economy".

These terms—"collaborative economy", "sharing economy", "on-demand economy" and "peer economy"—are currently used in media and other popular literature, and increasingly by state regulatory agencies and academic publications—to denote an emerging class of businesses which mediate, via the Internet, buyers and sellers of services. Prominent examples include the "ride-sharing" companies Uber and Lyft (which match requests for rides with providers of rides); the residential-space booking companies Airbnb and HomeAway (which connect requests for non-hotel lodging with renters and homeowners); the "P2P" loan services companies Lending Club and Prosper (matching borrowers with investors); and the freelance services companies oDesk and Elance (now merged as Upwork). These firms, largely funded by venture capitalists, are *not* generally buyers or sellers of goods themselves, as in a traditional production market (H. White 1981a); instead, they produce networked "marketplace platforms" which in turn provide opportunities to buy and sell—skimming a percentage of each transaction as a middleman—and are thus always distinctly less concerned with organizing the supply-chain logistics characteristic of commercial trade.

While platforms of this sort have existed for some time—eBay, after all, was profitably matching buyers and sellers of large varieties of goods online in the 1990s—they have become increasingly prominent in recent years in their overt "disruption" of various service industries, and the high (greater than $1 billion) "unicorn" valuations of Uber, Lyft, Airbnb, WeWork, Instacart, and others. Recently, multiple pop-business books—related to the emerging field of "platform economics" centered around MIT's Sloan School of Management (Evans, Hagiu, and Schmalensee (2006), Evans (2011))—have been published on the subject, with titles like "Matchmakers: the New Economics of Multisided Platforms" and "Platform Revolution: How Networked Markets Are Transforming the Economy—And How to Make Them Work for You".[395]

But should economic sociologists leave the theorization of marketplace platforms solely to economists? In this chapter I will suggest that economic sociology is uniquely positioned to

[395] Evans and Schmalensee (2016); Parker, Van Alstyne, and Choudary (2016). While I do not directly engage with the platform economics or industrial organization literature here, I intend this essay to be a first step towards developing a distinctive alternative to—and coherent critique of—that subfield's emphases on "two-sided" and "multi-sided" markets (Rochet and Tirole (2003); Evans (2003); Rysman (2009); Hagiu and Wright (2015)), which tend to privilege market scenarios featuring indirect network effects.

provide a distinctive interpretation of marketplace-platform phenomena, particularly via the theoretical insights from Patrik Aspers discussed in the last chapter; and, perhaps unexpectedly, via the long tradition of historical and ethnographic research on financial markets ranging from Abolafia (1996) to Cetina and Bruegger (2002) to MacKenzie and Pardo-Guerra (2014). Specifically, I will argue that many of the emergent organizational and regulatory complexities of the marketplace platform—especially with regard to competition, fragmentation, counterparty risk, and the possibility of self-regulation and cooperative ownership—have already been historically realized, in an equally dramatic fashion, in a completely different organizational domain: that of the securities exchange industry. The gradual introduction of electronic stock exchanges, for example, was accompanied by an extended controversy—simultaneously technological and political— over the nature of their relationship with traditional exchanges, and I will argue that this is just one of the intriguing and productive parallels with these newer controversial marketplace platforms.

But I will also suggest that it is essential that economic sociologists find a place for their traditions of inquiry in the rapidly accelerating contemporary debates on scalable marketplace platforms. The phenomena of "marketization" that these platforms induce—now known in France as "ubérisation"—represent a very different type of "financialization" than the increased centrality and dependence on financial markets articulated by Krippner (2012), and it is clear that many regulatory agencies are at risk of (mis-)regulating marketplace platforms as if they were traditional production firms. Examples of these densely-networked arenas of discussion include the U.S. Federal Trade Commission's workshop "The 'Sharing' Economy: Issues Facing Platforms, Participants, and Regulators" (FTC 2015) and hearings by the UK Parliament's House of Lords (European Union Committee 2016). Additionally, a multitude of debates have taken or are currently taking place within various urban governments, in which municipal representatives and local citizen groups are pitted against multibillion-dollar-valued private corporations to negotiate the ontological character of their services; and some of these debates unconsciously re-rehearse the way that U.S. regulators attempted to simultaneously—and arguably paradoxically—unify markets and enforce competition in the newly-emerging digital stock exchanges of the 1990s.

Switch-Role Markets: Lessons from Finance

Recall from Chapter 5 that in our study of the exchange as firm, we distinguished between *fixed-role markets* and *switch-role markets*, as described by Patrik Aspers (Aspers, 2007; Aspers, 2011). This distinction categorizes markets according to the extent to which actors are strictly assigned the roles of either buyers or sellers ("fixed-role"), or can switch between acting as a buyer or seller ("switch-role") (see Fig. 21 in Chapter 5.) Examples of fixed-role markets—where buyers and sellers are not interchangeable—include production markets (with firms competing to sell comparable products to a disjunct community of buyers) and labor markets; the canonical examples of a switch-role market—where buyers and sellers are interchangeable—are financial markets or other auctions (one can purchase a stock as a buyer, and then turn around to "flip" it as a seller). We saw "the exchange" as a site which had aspects of both fixed-role markets—i.e., multiple exchanges may compete to produce trading services for brokers and dealers—*and* switch-role markets: i.e., the familiar, furious "trading floor"-style buying and selling of shares.

In order, then, to understand the regulatory dynamics of marketplace platforms—which, like securities exchanges, have their primary activity the automated matching of buyers and sellers, and *not* production via a supply chain of upstream-to-downstream commodities—we can look to the much longer history of the financial markets produced by stock exchanges for clues. Specifically, I will focus on issues regarding (1) competition and fragmentation; (2) counterparty risk; and (3) self-regulation. By *competition/fragmentation* I refer to situations in which one can trade the same securities in multiple arenas; until the regulatory changes of the 1990s it was common, for various reasons, for 80% or more of trading in a given stock to occur on a single exchange. By *counterparty risk* I refer to the possibility that a participant on one side of a trade will default on their obligations; stock exchanges act to mitigate this risk in various ways, which I will discuss below. Finally, by *self-regulation* I refer to the governance structure of many exchanges, which deferred various aspects of regulatory action to the institutions themselves.

Competition/fragmentation in financial markets

The New York Stock Exchange (NYSE)—to rely on a prominent example—has a long history of deliberately limiting competition: the original Buttonwood Tree agreement in 1792, for example, fixed the minimum commission rate for member brokers at 0.25%, meaning that no matter how large the volume of shares traded, the brokers got the same non-negotiable cut; it also stipulated that members should deal with each other instead of non-members whenever possible (Harris 2003, 64). Through the 20th century, the NYSE actively prevented its members—the "broker-dealers" which traded on behalf of institutional and individual investors, and/or on their own behalf—from belonging to competing exchanges (such as the Consolidated Stock Exchange, founded in 1885, and the "curb" market which would become the American Stock Exchange.)[396] In response to the crash of 1929, the Securities Exchange Act of 1934 created the Securities and Exchange Commission (SEC) as an independent regulatory agency

[396] Michie (1986).

(primarily due to concerns regarding stock price manipulation), but much of the regulatory activity was left to the exchanges themselves, as so-called Self-Regulatory Organizations (SROs); and so their anti-competitive practices continued during the 20th century.[397] The NYSE's members were also prohibited from trading NYSE-listed securities on other (e.g. regional) exchanges, and while the SEC managed to abolish these restrictions for newly listed stocks after April 26, 1979, the NYSE's "Rule 390" prevented member competition in trading all pre-1979 stocks until 2000.[398]

Perhaps analogously to some of the incumbent "cartels" which various marketplace platforms are now held to be disrupting—such as the regulated "medallion" system for taxicabs in some large cities—the New York Stock Exchange in the early 1970s had a very high "seat price" for brokerage firms who wished to execute trades on the exchange. Moreover, existing rules made it nearly impossible for any new or alternative exchange venue to attract significant trading in NYSE-listed securities. Even after the SEC's 1975 Securities Acts Amendments which eliminated minimum fixed commission rates, the NYSE continued to dominate U.S. trading, with over 80% of the share volume in 1981.[399] But along with the 1975 Amendments came the emphatic call for a so-called National Market System (NMS), a concept which sought to encourage competition among exchanges by allowing traders to get the best price on multiple

---

[397] On the history of the SEC and of exchange self-regulation, see Seligman (2004) and Seligman (1982).

[398] Karmel (2002).

[399] Seligman (1985).

markets; and with that came the beginnings of technological interventions which aimed to link information about quotes for bids and offers, as well as information regarding executed trades.[400]

In the previous chapter I explained that in the late 1990s, when the Order Handling Rules and the Regulation of Exchanges and Alternative Trading Systems (Reg ATS) gave license to the new, broker-dealer-run ECNs to operate in an exchange-like manner, the race was on to draw liquidity away from the incumbent exchanges. These regulations also released the ECNs from the self-regulatory burden of being registered as an exchange. Instead of taking an equal commission from buyers and sellers, for example, ECNs like Island in 1997 began using so-called “maker-taker” pricing schemes which aimed to encourage the posting of orders on their system. If a match was made, the initial “liquidity provider” was rewarded with a high (0.25 cents/share) “liquidity rebate”, while the “taker” on the opposite side was charged a negative “access fee” (0.30 cents/share).[401] This subsidization approach—in which some platforms attracting one group of customers with subsidies at the expense of another group of customers, as in the traditional newspaper industry—was noted by the early platform economics literature (e.g. Rochet and Tirole 2003) as a common strategy to build a “critical mass”.

The effect of these regulatory changes, then, was certainly to “disrupt” an existing state of affairs in which there was little significant trading competition for incumbent exchanges. However, this competition—because it was happening at the firm level of the exchange industry (competing to provide trading services in given securities) rather than the level of a single,

---

[400] These systems emerging from the National Market System mandates include the “consolidated tape” (reporting executed trades), “consolidated quote” (reporting quotes for limit orders), and the Intermarket Trading System (ITS) (allowing, e.g. traders on regional exchanges to forward their orders to the NYSE, or vice versa) (Seligman 1984).

[401] Foucault (2012); Angel, Harris, and Spatt (2010). For a comparison of these U.S. securities rules to the European Union’s Markets in Financial Instruments Directive (MiFID) see Boskovic, Cerruti, and Noel (2010).

unified market for particular stocks (where individual buyers and sellers might thus be concentrated in their "competition" for the best price)—came to be described as "fragmentation", a pejorative term which indicates a move away from an idealized market which finds its Walrasian equilibrium precisely in the participants meeting at a single continuous auction. From the story detailed above, however, it would seem that for switch-role markets, competition is necessarily *also* fragmentation.

The effect of this regulated competition/fragmentation on the exchange industry in the coming decade was extreme, with rapid waves of mergers as well as demutualizations—meaning that these former mutual cooperatives went public (and thus became listed firms on their own trading floors).[402] Recall from Chapter 5 that the exchange industry scholar Ruben Lee saw that in such a competitive environment—with the cost of a transaction headed to zero—that one of the last reliable sources of revenue for exchanges were the quotes and trade data themselves; he predicted that exchanges would thus become, like media companies, "content providers" (Lee 2002). His observation implicitly ties the disruption of the exchanges to the well-known disruption of other platforms like newspapers at the hands of online competition; and thus gives us one perspective on the future of marketplace platforms, which also equally at risk for competition and fragmentation. As Lee predicted, as the commission per transaction decreased in a more competitive environment, these newly public exchanges have increasingly derived their revenue from receiving revenues for market data.[403] Indeed, some ECNs (like Island) which had originally avoided being registered as exchanges later sought to be registered as exchanges

[402] On the demutualized exchange see Macey, Jonathan R. and O'Hara, Maureen (2005).

[403] Hasbrouck (2014). Reg NMS' "market data rule" imposes a weighted formula based on trade volume and frequency, as well as for improving on the visible best bid and offer (Hasbrouck 2007). (In Europe, there is no comparable regulated consolidation of market data.)

instead of broker-dealers, precisely because of the possibility of collecting revenue from their market data under U.S. regulations.[404]

Counterparty risk in financial markets

It is the economic concept of *counterparty risk*—the possibility that the opposing party to a trade will fail to settle their debt—that inspired various medieval financial innovations described by Braudel (1992).[405] These mechanisms included *bills of exchange*, debt instruments which could be redeemed at trusted merchant banks; *fairs*, which at their conclusions took on the role of a clearinghouse, netting bills of exchange among merchants; and finally *stock exchanges* themselves, whose member dealers served as counterparties to both buyers and sellers. The "anonymous" trading we associate with modern stock exchanges—where buyer and seller may never meet in person, and yet manage to trust each other to complete a transaction—is only possible given highly standardized goods (such as stocks); and (especially in the case of forward or futures trading) a form of centralized *clearinghouse* institution which attempts to guarantee payment in the event of default of one party.[406] By limiting its members, exchanges provided an element of trust that the opposing party would not default; by centralizing clearing (in what is called a "centralized counterparty" (CCP)) , it provided further guarantees of ultimate

---

[404] Markham and Harty (2008). In 2009, the CEO of the Direct Edge ECN stated: "As an exchange operator, you follow the money. With exchange status and market penetration you can collect significant market data fees here in the USA" (R. A. Schwartz, Byrne, and Schnee 2013, 18).

[405] On counterparty risk and broker defaults on the Paris Bourse, see Riva and White (2011). For other discussion of financial risk in the economic sociology literature, see Zaloom (2004); Hardie (2004); MacKenzie, Beunza, and Hardie (2009); and Holzer and Millo (2005).

[406] On clearinghouse mechanisms, see Millo et al. (2005).

settlement.[407] The stock exchange is thus an institution that limits the risks of exchange on the financial markets it produces; we will later see important analogies to this state of affairs in marketplace platforms.

Self-regulation in financial markets

The self-regulatory status of stock exchanges—effected as a matter of pragmatic expediency in 1934—was something of a curiosity for mid-century observers: one commentator noted that "stock exchanges seem to have been permitted to function almost as though there were no antitrust problem at all… the technical relationship of the exchange to the state is, roughly, the same as the relationship of a private club."[408] Abolafia, in his ethnographic observations of futures and securities markets, noted that "self-regulators are, in fact, engaged in a delicate balancing act between profits and prudence… they know that the market's legitimacy is essential to their long-term viability."[409] He contrasted the comparatively freewheeling futures pits with the presence of floor governors (SRO officials) on the NYSE floor, noting that "members exhibited a boastful pride in the rules and in the rules' consequences for a fair and equitable marketplace".[410] The occasional large-scale study of the exchange industry in the 20th century (e.g. Securities and Exchange Commission (1963), Securities and Exchange Commission (1994)) raised the various potential problems of combining oversight and competition, without making firm recommendations for significant change to the SRO status quo. The question remains as to

---

[407] On the introduction of centralized clearing to the NYSE, see Bernstein, Hughson, and Weidenmeier (2014).Note that the concept of clearing (bilateral, multilateral) presumes switch-role markets, while the concept of settlement (fund transfer between counterparties) does not.

[408] Westwood and Howard (1952).

[409] Abolafia (1996, 101–102). For a more critical perspective on SROs see Miller (1985).

[410] Abolafia (1996, 104).

which type of industries demand or deserve self-regulatory status, and what precisely about trading services should lead it to remain outside more commercial antitrust regulations: if it is because an exchange is a natural monopoly, why deliberately induce competition? And if it is *not* a natural monopoly, then why delegate enough control to the exchange to permit it to maintain anticompetitive practices? As part of the next section, I will suggest that—whether we know it or not—state legislatures have (perhaps unfairly) granted a kind of self-regulatory status to certain marketplace platforms, and that explicitly expanding or constraining this SRO role will be an important policy prescription of the future.

The Switch-Role Markets of Marketplace Platforms: A Comparison

The current approaches to regulation of firms like Uber/Lyft and Airbnb/VRBO are in part misplaced, as these firms have many qualities that are less like traditional participants in a taxicab or hotel industry and far more like the new electronic stock exchanges of the 1990s; it may be the case that legislators would do better to contend with the "market microstructure" of the businesses in question. See Fig. 25 below for an illustration showing the sharing-economy analogy to Fig. 22 in Chapter 5; for a broad comparison of the various aspects discussed in this section, see Table 11. The interjection of exchange-like logic into commercial domains, I suggest—i.e., the competitive substitution of fixed-role production/consumption markets with switch-role markets which automatically match buyers and sellers—is at the heart of the perfect storm of controversy which these businesses appear to continuously generate. As in the previous section, I will address three aspects of these marketplace platform firms: (1) I will consider the relevance of *competition and fragmentation* by examining the potential (but relative absence at present) for linking "orders" between competing marketplace-platform firms, in an analogy to 1990s-era developments on stock exchanges. (2) I will address *counterparty risk* by discussing the use of reputation feedback systems and other mechanisms for facilitating trust. (3) Finally, I will examine the practices, promises, and potential (or lack thereof) of encouraging a *self-regulatory* approach to marketplace platforms.

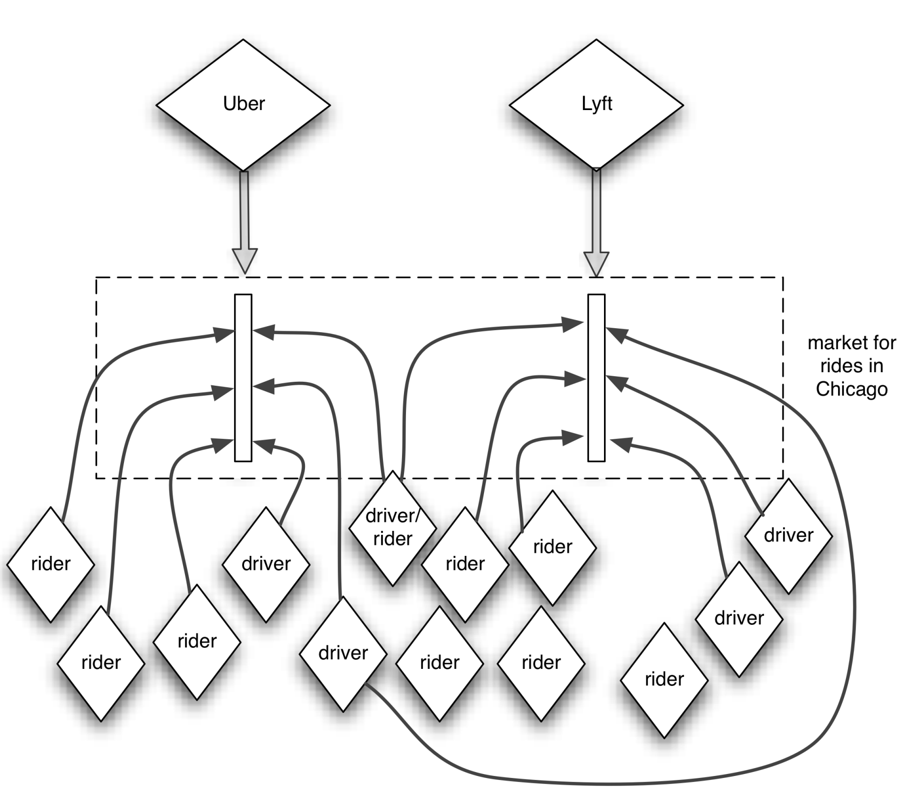

Figure 25. Fixed-role and switch-role markets in the production of ride services. Producers of ride services in a given city include Uber and Lyft (incumbent taxicab services not shown). Drivers and riders are "in the market" for the exchanges' services, which consist of potentially switch-role markets in which they can alternately take the role of a driver or a rider (though not all riders are also drivers). Source: author.

	NYSE (pre-2000s)	*ECNs (in late-1990s securities exchange industry)*	*Uber/Lyft (ride services industry)*	*Airbnb/ Homeaway (hospitality services industry)*	*Instacart/Deliveroo (groceries/food delivery services industry)*
OWNERSHIP STRUCTURE	Member-owned cooperative (became public corporation in 2006)	Privately owned / varying sources of funding	Privately owned / VC funded	Privately owned / VC funded	Privately owned / VC funded
MARKET ROLES	Fixed-role producer of physical switch-role markets (on the trading floor) for various stocks	Fixed-role producers of electronic switch-role markets for various stocks	Fixed-role producers of markets for rides in various cities	Fixed-role producers of markets for short-term rentals in various cities	Fixed-role producer of delivery services for (fixed-role) markets for perishable goods (supermarkets, restaurants)
COMPETITION /FRAGMENT-ATION	Competition limited to "third market" of off-exchange members (after repeal of Rule 390, decline of market share to electronic exchanges)	After Order Handling Rules, ECNs fragmented markets for OTC securities by drawing order flow away from Nasdaq dealers	Competition with incumbent taxicab services and various other ride services startups; markets for rides overtly fragmented, but covertly connected via drivers running multiple apps	Incumbent hotel / B&B industry; other hospitality services startups	Limited due to overt partnership with fixed-role supermarkets and restaurants
SWITCH-ROLE ASPECTS	Buyers and sellers of securities interchangeable	Buyers and sellers of securities interchangeable (but various	Partial/potential (drivers are often periodic riders; less common for riders to be drivers. Cannot "flip" a ride.)	Partial (similar to ride services, hosts are often users, users less often hosts)	Partial (users less likely to also be shoppers/delivery drivers)
TRANSACT-ION FEES	Varies and minimum commission negotiable (since 1975); began as 0.25% commission per share	Varies and minimum commission not fixed	20-25% fixed-rate commission	6-12% fixed-rate commission for guests; 3% fixed-rate commission for hosts	\$3.99-\$9.99 flat delivery fee; 0-15% markup on prices depending on store (Instacart); £2.50 flat fee per delivery (Deliveroo)
COUNTER-PARTY RISK	National Securities Clearinghouse Corp. (NSCC) as central counterparty (CCP)	Also used NSCC (jointly owned by NYSE, Amex, and NASD).	Bilateral ratings system; centralized netting and payment processing	Bilateral ratings system; centralized netting and payment processing	Unilateral ratings system (Instacart); customer service line only (Deliveroo); centralized netting and payment processing

Table 11. Comparison of stock exchanges ca. the 1990s (NYSE and competitor ECNs) with various marketplace platform firms.

Competition/fragmentation in marketplace platforms

Like the NYSE "club" of the 1970s, Uber/Lyft and Airbnb in particular have become notorious in many municipalities for their anti-regulatory attitudes, seeking to halt much nascent legislation through extensive lobbying. But unlike the NYSE throughout most of the 20th century, these firms are more at risk from competition by future platform firms, assuming those competing platforms can reach a sustainable critical mass. To use the phrasing of economists, there are low "switching costs" between, e.g., using Uber versus using Lyft (one simply has to download a new mobile app.) To put it another way, the "off-exchange" trading restrictions that protected the NYSE—preventing the occurrence of equivalent transactions (of e.g., NYSE-listed securities) on other exchanges—are not present in this case (many platforms are available for the same approximate service, a ride from point A to point B). At the same time, the phenomenon of "liquidity attracting liquidity" remains, so that the more drivers/riders use the Uber platform, the more appealing the platform is for future participants (just as a confluence of buyers/sellers attracts other buyers/sellers). No legal barriers prevent the interlinking of the markets, however, only technical ones. Therefore, the apps may deliberately attempt to block external firms from displaying price quotes—as Uber did for Urbanhail, a price comparison startup for ride services in Boston.[411])

We can see then that the most significant difference between stock exchanges and Uber/Lyft is that the former facilitates the buying and selling of perfectly standardized (and thus fungible) *goods*, while the latter facilitates the buying and selling of (more or less standard) *services*; for while one can trivially "flip" a stock, it is harder to see how one can literally "flip" a

---

[411] Woodward (2016).

ride or short-term rental—though many Airbnb hosts, for example, are also Airbnb customers, often simultaneously (e.g. while one is on vacation).[412] To problematize this traditional goods-services distinction, with its origins in Adam Smith's concepts of productive and unproductive labor, requires a return to debates in economic sociology in the early 2000s (Callon, Méadel, and Rabeharisoa (2002); Slater (2002)).[413] Inspired by Gadrey (2000), Callon et. al. find that frames around service activities facilitate "the singularization of products" (Aspers' *standard* market); and it facilitates the consumer's "attachment to and detachment from" products (as in the purchase of a temporary ride from point A to point B; or, perhaps, the *switch-role* character of getting "in and out" of a market by, e.g., buying and quickly selling). Despite this, the ability of goods and services to be conflated for centuries—and why their arguably "sociological" distinction remained unproblematic for late-20th-century economists in many regards—is that their exchange can be represented and recorded by a *transaction* (Hill 1977). As such, marketplace platforms, whether they match buyers and sellers of goods (e.g. eBay, Amazon's used-books marketplace) or buyers and sellers of services (Uber/Lyft, Taskrabbit), have the same basic revenue model at the center of their platforms: to bring together as many buyers and sellers together as possible, and to take a percentage of each facilitated transaction.

Taking the notion of liquidity in a financial market and applying it to these marketplace platforms can be instructive, to see how the analogy can apply to both goods and services. For

---

[412] Adam Smith remarks that the labors of servants, for example, "generally perish in the very instant of their performance, and seldom leave any trace or value behind them for which an equal quantity of service could afterwards be procured" (A. Smith 1776, 358).

[413] For example, it reveals that many "on-demand"-style firms may match buyers and sellers of services, but those services (specifically, *delivery*, a.k.a. the temporary service-like intermediation of goods transactions) are potentially rather closely integrated into traditional fixed-role production markets for goods. Indeed, some on-demand firms (Instacart, Shyp) are closely integrated with producer firms (e.g. supermarkets and shipping carriers, respectively) that they have reclassified some or all of their shoppers/couriers as employees.

example, the claim of Uber's representatives that their prices are a function of "supply and demand" can lead one to ask whether drivers represent supply and riders demand, or vice versa. To use the securities market analogy—in which those who post limit orders are market "makers" and those who post market orders the price "takers"[414]—the driver is ostensibly a "maker" of liquidity, with the rider a "taker"; but from the perspective of the driver, who also needs liquidity, the riders could be the "makers" and her the "taker."[415] On Uber's platform, for example, a driver can be punished for turning down too many rides (being "unmarketable"), and riders can abort their ostensibly "marketable" orders for rides if the estimated price (or estimated "surge" factor) is too high. But note the comparative opacity and discontinuity of this matching process: in a financial market, if offers suddenly and discontinuously "surged" to 1.4 times their previous value, automated circuit breakers would halt trading! There is thus reason to be suspicious of Uber's "Economics 101" claims, when their system is not truly running a continuous auction matching explicit bids and offers. Interestingly, the Uber/Lyft competitor Sidecar, beginning in February 2014, allowed drivers to bid on rides and riders to choose based on price or other driver parameters (e.g. closer drivers, drivers with higher ratings); these competitor features brought the exchange-like character of these systems to the fore, but this pricing system was not enough to sustain Sidecar as a viable competitor.[416]

---

[414] On the distinction between makers and takers in financial markets, see Foucault (2012).

[415] While there is certainly an overall asymmetry between the rider and driver as actors (the former might consummate a ride once in a day, but the latter several times), during their mutual engagement it is not necessarily obvious which one provides liquidity while the other takes it away.

[416] Tam (2014).

Counterparty risk in marketplace platforms

One controversial aspect of marketplace platforms is the use of interactive ratings systems to induce service quality and customer protection by providing a measure of participant reputation; but ratings systems (pioneered in part by eBay, and common in, e.g., Uber/Lyft, Airbnb, and more) are only one way that users of marketplace platforms attempt to mitigate counterparty risk.[417] First, one should note that these ratings systems are often *bilateral*—the rider rates the driver, but the driver also rates the rider—which is suggestive of switch-role markets because the buyer is no different from the seller (i.e., both can be rated in the same manner). By contrast, in production markets it is more common to rate only one side, as in Yelp reviews, which are strictly fixed-role and unilateral (for an analysis of consumer restaurant reviews, see Mellet et al. (2014)).

But the other, less appreciated way these platforms mitigate risk is by providing various guarantees of settlement and protection from other liabilities, much as a stock or futures exchange mitigates credit risk with centralized clearing and settlement procedures, as described above. In the case of many marketplace platform services, one's credit card is not charged (or bank account deposited) until the service is consummated; Airbnb specifically provides $1M liability insurance in the case of accident or death. Much like the transactions processed by clearinghouses, economic transactions "between", e.g., a rider and driver are actually composed of two separate transactions: one from the rider's credit or debit card to Uber/Lyft and one from Uber/Lyft to the driver (with rider payments netted weekly and middleman fees deducted). The

[417] For a prescient comparison of eBay to financial markets, see Kollock (1999).

mitigation of risk on the part of "collaborative economy" marketplace platforms is thus not entirely dependent on collaborative ratings but instead uses traditional centralized clearing and settlement methods recognizable from the exchange industry to facilitate anonymous transactions. We can thus also see how "peer-to-peer" lending firms (e.g. Lending Club, Prosper) could initially be distinguished by their blending of traditional risk management (e.g. FICO credit ratings) with more "collaborative" information about social ties.[418]

Self-regulation in marketplace platforms

Before the waves of demutualization and mergers of the 2000s, exchanges like the NYSE were member-owned, non-profit cooperatives, a fact that is often lost in dismissive discussions about Wall Street and capitalism, and one which is especially lost on the recent critical commentary that private, for-profit, venture-capital-funded marketplace platforms could also be realized as member-owned "platform cooperatives" (Scholz 2016). Given the history of stock exchanges, this perspective is both reasonable (it is, indeed, technically quite possible to imagine a member-owned ride services or short-term rental services platform) but also dismissive of the revenue challenges that can emerge in a technopolitical situation where any of your customers (such as the brokerages of the incumbent stock exchanges) could turn and become a competitor (e.g., by implementing their own order matching system and drawing away order flow with various incentives and rebates).

However, the appropriate regulation of marketplace platforms, whether private or cooperatively owned, remains in question. If, as I have been arguing, marketplace platform firms are like stock exchanges, how can the self-regulatory organization (SRO) status of exchanges

---

[418] Verstein (2011).

inform their regulation? It would appear that by conceiving of these companies as traditional competitors (i.e. as similar to taxicab companies or hotels), many of their practices appear outright to be illegal. But if we conceive of them as exchanges, then we can see that some combination of self-regulation, transparency, and oversight may be more appropriate; an argument like this has recently been proposed by Cohen and Sundararajan (2015). But even given the SRO status of exchanges which provides a measure of day-to-day regulatory autonomy, it should be noted that exchanges are comparatively far more bound by SEC rules than any current marketplace platform firm is by any corresponding agency (such as the Federal Trade Commission (FTC)). Specifically, we can look at the obligations of exchanges to expose market data to facilitate inter-exchange competition, but also for oversight purposes (so that, e.g., the SEC can investigate "flash crashes"); this is precisely the kind of information which some legislators have found very difficult to elicit from Uber/Lyft/Airbnb, especially in any kind of real-time modality.[419] A modest, and yet arguably far-reaching, proposal would be to permit the SRO-like qualities of existing marketplace platform firms—the enforcement of business practices (using internal data) and the use of reputation feedback systems—but to mandate a certain level of data transparency to regulators. The potential also exists to mandate data exposure even to competing platforms, but to do so would be—as in the history of the exchange industry—to trade anticompetition for hypercompetition (i.e. from one or two major exchanges to dozens of competing exchanges and dark pools). Just as with the exchanges, it will be increasingly necessary to step back and determine a sustainable combination of regulation and

---

[419] On the increasing importance of data monitoring for financial regulators, see Flood, Mendelowitz, and Nichols (2013).

self-regulation; but it will not be possible for legislators to move forward until the current level of opacity of operational data is explicitly reduced.

Coda: Time, Space, Value and the Primacy of Finance

With this final chapter, I projected into 21st century regulatory policy the perspective developed across this dissertation: the relevance of the long history of financial sociotechnics and practice to the history and sociology of computing, as well as to the history of the interventions of computing practitioners into other forms of markets and marketplaces. In order to argue this, it was necessary to ontologically and semiotically distinguish between those practices which abstracted away from temporality—namely, the relational database and the transaction concept—and those which abstracted away from spatiality, namely messaging middleware and distributed systems. The practice of finance, which is distinguished by its socially organized transduction between time and value, was thus always closely involved with these conceptual and pragmatic interventions.

In order to make this argument, it was necessary to draw on a wide diversity of interdisciplinary material and sources, drawing from social theory and semiotics, to the philosophy of technology, to the history of computing and economic sociology. This story involves a mix of many actors who, for the most part, never overtly made any of these connections, but nevertheless took part in a radical transformation which can be seen, increasingly, as a financialization and marketization of everyday life. I hope that this combination of perspectives can convince the reader that just as there can be no society without technics, there can be no technics without politics: and transitively, that there can be no intentional institutional change independent of a sociotechnical and conceptual scaffolding to make those intentions realizable.

At the same time, however, this work should not be misinterpreted as an argument towards totalized holism. It should be seen instead as insistently disambiguating certain concepts and categories at each step which are often conflated by others. In Chapter 2, Charles Bachman reasoned that tabular representation might be subsumed by network representation, instead of seeing the organizational benefit that the relational model's elimination of indexicality provided. In Chapter 3, the formalization of transaction was successful in its conceptual erasure of the temporality and potential complexities of change. In Chapter 4, that erased temporality returns to the fore, because one cannot have a remote 'transaction' without delivering a message that moves through space. This is not because transactions can be 'reduced to' messages or vice versa (cf. the reduction of time to space which was so offensive to Bergson), but instead because they are intertwined in a duality. In Chapter 5, the electronic exchanges made possible by a combination of tabular representation, transactions, and message-oriented data communication were arguably misunderstood by the Securities and Exchange Commission, who did not coherently distinguish between fixed-role production markets and switch-role financial markets (the latter of which were the product of exchanges). As shown in Chapter 6, this conflation—in the case of other industries in which firms' products are platforms for exchange—continues to the present day, and remains the source of significant controversy.

References

Abbate, Janet. 2000. *Inventing the Internet*. MIT Press.

Abbott, Andrew. 2001. *Time Matters: On Theory and Method*. 1 edition. Chicago: University of Chicago Press.

———. 2016. *Processual Sociology*. University of Chicago Press.

Abolafia, Michel. 1996. *Making Markets: Opportunism and Restraint on Wall Street*. Cambridge: Harvard University Press.

Adam, Barbara. 1994. *Time and Social Theory*. Cambridge: Polity.

Agar, Jon. 2006. "What Difference Did Computers Make?" *Social Studies of Science* 36 (6):869–907.

Ahrne, Göran, Patrik Aspers, and Nils Brunsson. 2015. "The Organization of Markets." *Organization Studies* 36 (1):7–27.

Akera, Atsushi. 2001. "Voluntarism and the Fruits of Collaboration: The IBM User Group, Share." *Technology and Culture*, 710–36.

Akidau, Tyler, Robert Bradshaw, Craig Chambers, Slava Chernyak, Rafael J. Fernandez-Moctezuma, Reuven Lax, Sam McVeety, et al. 2015. "The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing." *Proceedings of the VLDB Endowment* 8 (12).

Alexander, Christopher. 1999. "The Origins of Pattern Theory: The Future of the Theory, and the Generation of a Living World." *IEEE Software* 16 (5):71–82.

Allix, Andre. 1922. "The Geography of Fairs: Illustrated by Old-World Examples." *Geographical Review* 12 (4):532–69.

Alpert, Bill. 1999. "Wall Street Revolution: Technology Is Toppling Trading's Berlin Wall." *Barron's* 79 (30):19–22.

Amihud, Yakov, Thomas S. Y. Ho, and Robert A. Schwartz. 1984. "Overview of the Changing Securities Markets." In *Market Making and the Changing Structure of the Securities Industry*, edited by Yakov Amihud, Thomas S. Y. Ho, and Robert A. Schwartz. Lexington Books.

Ananda, A. L., B. H. Tay, and E. K. Koh. 1992. "A Survey of Asynchronous Remote Procedure Calls." *SIGOPS Oper. Syst. Rev.* 26 (2):92–109. https://doi.org/10.1145/142111.142121.

Anderson, Benedict. 1983. *Imagined Communities: Reflections on the Origin and Spread of Nationalism*. London: Verso.

Angel, James J., Lawrence E. Harris, and Chester S. Spatt. 2011. "Equity Trading in the 21st Century." *Quarterly Journal of Finance* 1 (1):1–53.

Anon., et al. 1985. "A Measure of Transaction Processing Power." *Datamation*, no. 7 (April):112–18.

A.R.D. Mathias. 1992. "What Is Mac Lane Missing?" In *Set Theory of the Continuum*, by Haim Judah, Winfried Just, and Hugh Woodin, 113–18. Springer.

Aspers, Patrik. 2005. "Performativity, Neoclassical Theory and Economic Sociology." *Economic Sociology European Electronic Newsletter* 6 (2).

———. 2007a. "Theory, Reality, and Performativity in Markets." *American Journal of Economics and Sociology* 66 (2):379–398.

———. 2007b. "Theory, Reality, and Performativity in Markets." *American Journal of Economics and Sociology* 66 (2):379–98.

———. 2009. “How Are Markets Made?” MPIfG Working Paper 09/2. Cologne: Max Planck Institute for the Study of Societies.
———. 2011. *Markets*. Cambridge: Polity.
Astrahan, M. M., B. Housman, J. F. Jacobs, R. P. Mayer, and W. H. Thomas. 1957. “Logical Design of the Digital Computer for the SAGE System.” *IBM Journal of Research and Development* 1 (1):76–83. https://doi.org/10.1147/rd.11.0076.
Atten, Mark van. 2012. “Kant and Real Numbers.” In *Epistemology versus Ontology Essays on the Philosophy and Foundations of Mathematics in Honour of Per Martin-Löf*, edited by Dybjer et. al.
Audoze, Françoise. 2002. “Leroi-Gourhan, a Philosopher of Technique and Evolution.” *Journal of Archaelogical Research* 10 (4):277–306.
Austrian, Geoffrey. 1982. *Herman Hollerith, Forgotten Giant of Information Processing*. Columbia University Press.
Avizienis, A., H. Kopetz, and J. C. Laprie, eds. 1987. *The Evolution of Fault-Tolerant Computing*. Vol. 1. Springer-Verlag.
Babbage, Charles. 1832. *On the Economy of Machinery and Manufactures*.
Bach, A. Francis. 1998. “Wall Street Takes Stock of IP Multicast.” *Business Communications Review*, January.
Bachelard, Gaston. 2000. *The Dialectic of Duration*. Rowman and Littlefield.
———. 2001. *The Formation of the Scientific Mind*. Manchester: Clinamen Press Ltd.
Bachman, Charles W. 1973. “The Programmer as Navigator.” *Communications of the ACM* 16 (11):653–658.
———. 2009. “The Origin of the Integrated Data Store (IDS): The First Direct-Access DBMS.” *IEEE Annals in the History of Computing*, 42–54.
Baker, Wayne. 1984. “The Social Structure of a National Securities Market.” *American Journal of Sociology* 82 (4):775–811.
Baldonado, Danny. 2015. *George Turin Memorial - 08 Harvey Wagner*. https://vimeo.com/138053141.
Ball, Richard A. 1978. “Sociology and General Systems Theory.” *The American Sociologist* 13 (1):65–72.
Baran, Paul. 1964. “On Distributed Communications: I. Introduction to Distributed Communication Networks.” RAND.
Barley, Stephen. 1986. “Technology as an Occasion for Structuring:Evidence from Observation of CT Scanners and the Social Order of Radiology Departments.” *Administrative Science Quarterly* 31:78–108.
Bartlett, Joel. 1981. “A NonStop Kernel.” *ACM SIGOPS Operating Systems Review* 15 (5):22–29.
Bartlett, Joel, Wendy Bartlett, Richard Carr, Dave Garcia, Jim Gray, Robert Horst, Robert Jardine, Doug Jewett, Dan Lenoski, and Dix McGuire. 1998. “The Tandem Case: Fault Tolerance in Tandem Computer Systems.” In *Reliable Computing Systems*, edited by Daniel P. Siewiorek and Robert S. Swarz, 3rd ed.
Bartlett, Wendy, and Lisa Spainhower. 2004. “Commercial Fault Tolerance: A Tale of Two Systems.” *IEEE Transactions on Dependable and Secure Computing* 1 (1).
Baskett, Forest, John H. Howard, and John T. Montague. 1977. “Task Communication in DEMOS.” In *Proceedings of the Sixth ACM Symposium on Operating Systems*

Principles, 23–31. SOSP ’77. New York, NY, USA: ACM. https://doi.org/10.1145/800214.806544.
Bátiz-Lazo, Bernardo. 2009. “Emergence and Evolution of ATM Networks in the UK, 1967-2000.” *Business History* 51 (1):1–27.
Bátiz-Lazo, Bernardo, Thomas Haigh, and David L. Stearns. 2013. “How the Future Shaped the Past: The Case of the Cashless Society.” *Enterprise & Society* 15 (1).
Bauman, Richard, and Charles L. Briggs. 1990. “Poetics and Performance as Critical Perspectives on Language and Social Life.” *Annual Review of Anthropology* 19:59–88.
Bayer, Rudolf. 2002. “B-Trees and Databases, Past and Future.” In *Software Pioneers*, edited by M. Broy and E. Denert, 233–44. Springer-Verlag Berlin Heidelberg.
Bayer, Rudolf, and Edward M. McCreight. 1972. “Organization and Maintenance of Large Ordered Indices.” *Acta Informatica* 1:173–89.
Becker, Brandon, Eugene Lopez, Victoria Berberi-Doumar, Richard Cohn, and Alden S. Adkins. 1992. “Automated Securities Trading.” *Journal of Financial Services Research* 6 (4):327–41.
Benn, Alec. 2000. *The Unseen Wall Street of 1969-1975: And Its Significance for Today*. Quorum Books.
Bergson, Henri. 1910. *Time and Free Will*. London, New York: Allen & Unwin.
———. 1944. *Creative Evolution*. The Modern library.
Berkeley, Edmund Callis. 1949. “Measuring; Massachusetts Institute of Technology’s Differential Analyzer No. 2.” In *Giant Brains, or Machines That Think*. New York: John Wiley and Sons.
Bernstein, Asaf, Eric N. Hughson, and Marc Weidenmeier. 2014. “Counterparty Risk and the Establishment of the New York Stock Exchange Clearinghouse.” http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2482472.
Bernstein, Philip, Vassos Hadzilacos, and Nathan Goodman. 1987. *Concurrency Control and Recovery in Database Systems*. Addison Wesley.
Bertalanffy, Ludwig von. 1968. *General Systems Theory: Foundations, Development, Applications*. New York: George Braziller.
Beunza, Daniel, and David Stark. 2012a. “From Dissonance to Resonance: Cognitive Interdependence in Quantitative Finance.” *Economy and Society*, 1–35.
———. 2012b. “From Dissonance to Resonance: Cognitive Interdependence in Quantitative Finance.” *Economy and Society* 41 (3):383–417.
Bijker, Wiebe E., Thomas P. Hughes, and Trevor Pinch, eds. 1987. *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*. MIT Press.
Birman, Ken, Coimbatore Chandersekaran, Danny Dolev, and Robbert van Renesse. 2006. “How the Hidden Hand Shapes the Market for Software Reliability.”
Birman, Kenneth. 2012. *Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services*. 2012 edition. London: Springer.
Birman, Kenneth, and Robert Cooper. 1990. “The ISIS Project: Real Experience with a Fault Tolerant Programming System.” In *Proceedings of the 4th Workshop on ACM SIGOPS European Workshop*, 1–5. EW 4. New York, NY, USA: ACM. https://doi.org/10.1145/504136.504153.

Birman, Kenneth, A. El Abbadi, W.C. Dietrich, Thomas Joseph, and T. Raeuchle. 1985. "An Overview of the ISIS Project." IEEE Distributed Processing Technical Committee Newsletter.

Birrell, Andrew D., and Bruce Jay Nelson. 1984. "Implementing Remote Procedure Calls." *ACM Trans. Comput. Syst.* 2 (1):39–59. https://doi.org/10.1145/2080.357392.

Bjork, L. A., and C. T. Davies Jr. 1972. *The Semantics of the Preservation and Recovery of Integrity in a Data System*.

Bjork, Lawrence A. 1973. "Recovery Scenario for a DB/DC System." In *Proceedings of the ACM Annual Conference*, 142–146. ACM '73. New York, NY, USA: ACM.

Black, Fischer. 1971a. "Toward a Fully Automated Stock Exchange." *Financial Analysts Journal* 27 (4):28–35+44.

———. 1971b. "Toward a Fully Automated Stock Exchange." *Financial Analysts Journal* 27 (6):24-28+86-87.

Blackford, Victoria. 1988. "The Financial Trading Game in 1988." *Network World*, October.

Blau, P., C. McHugh-Falbe, W. McKinley, and T. Phelps. 1976. "Technology and Organization in Manufacturing." *Administrative Science Quarterly* 21:20–40.

Boggs, David Reeves. 1982. "Internet Broadcasting." Stanford, CA, USA: Stanford University.

Borr, Andrea. 1981. *Transaction Monitoring in ENCOMPASS*.

Boskovic, Tanja, Caroline Cerruti, and Michel Noel. 2010. "Comparing European and U.S. Securities Regulations MiFID versus Corresponding U.S. Regulations." World Bank Working Paper No. 184. Washington, D.C.: World Bank.

Bowker, Geoffrey, and Susan Leigh Star. 1999. *Sorting Things Out*. MIT Press.

Brady Commission. 1988. *Report of the Presidential Task Force on Market Mechanisms*. Government Printing Office.

Brady, Geraldine. 2000. *From Peirce to Skolem: A Neglected Chapter in the History of Logic*. Elsevier.

Braudel, Fernand. 1992. *The Wheels of Commerce (Civilization and Capitalism: 15th-18th Century -Volume 2)*. University of California Press.

Breeden, Richard C. 1991. *Past Successes, Future Challenges*. Boca Raton, Florida: Securities and Exchange Commission.

Burg, Urs von. 2002. *The Triumph of Ethernet: Technological Communities and the Battle for the LAN Standard*. 1 edition. Stanford, Calif: Stanford Business Books.

Burman, Michael. 1985. "Aspects of a High-Volume Production Online Banking System." *Compcon Spring 1985, February 25-28, Thirtieth IEEE Computer Society International*, 244–48.

Caliskan, Koray, and Michel Callon. 2010. "Economization, Part 2: A Research Programme for the Study of Markets." *Economy and Society* 39 (1):1–32.

Çalıskan, Koray, and Michel Callon. 2010. "Economization, Part 2: A Research Programme for the Study of Markets." *Economy and Society* 39 (1):1–32.

Callaghan, Brent. 2000. *NFS Illustrated*. Addison-Wesley Professional.

Callon, Michel. 1998. "Introduction: The Embeddedness of Economic Markets in Economics." In *Laws of the Market*, edited by Michel Callon. Blackwell.

———. 2005. "Why Virtualism Paves the Way to Political Impotence: A Reply to Daniel Miller's Critique of The Laws of the Markets." *Economic Sociology: European Electronic Newsletter* 6 (2):3–20.

Callon, Michel, Cécile Méadel, and Vololona Rabeharisoa. 2002. "The Economy of Qualities." *Economy and Society* 31 (2):194–217.
Callon, Michel, and Fabian Muniesa. 2005. "Economic Markets as Calculative Collective Devices." *Organization Studies* 26 (8):1229–50.
Calvin, Donald L. 1984. "The National Market System: A Successful Adventure in Industry Self-Improvement." *Virginia Law Review* 70 (4):785–811.
Campbell-Kelly, Martin. 2003. *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*. MIT Press.
———. 2010. "Victorian Data Processing." *Communications of the ACM*, no. 10 (October):19–21.
Campbell-Kelly, Martin, William F. Aspray, Nathan Esmenger, and Jeffrey R. Yost. 2014. *Computer: A History of the Information Machine*. Westview Press.
Campbell-Kelly, Martin, and Daniel D. Garcia-Swartz. 2008. "Economic Perspectives on the History of the Computer Time-Sharing Industry, 1965–1985." *IEEE Annals of the History of Computing*.
Cantillon, Estelle, and Pai-Ling Yin. 2011. "Competition between Exchanges: A Research Agenda." *International Journal of Industrial Organization* 29 (3):329–36.
Capurro, Rafael. 2011. "Angeletics – A Message Theory." In *Messages and Messengers: Angeletics as an Approach to the Phenomenology of Communication*, edited by Rafael Capurro and John Holgate, 33–42. Wilhelm Fink Verlag.
Carruthers, Bruce. 1996. *City of Capital: Politics and Markets in the English Financial Revolution*. Princeton: Princeton University Press.
Carruthers, Bruce G., and Arthur L. Stinchcombe. 1999. "The Social Structure of Liquidity: Flexibility, Markets, and States." *Theory and Society* 28 (3):353–82.
Cetina, Karin Knorr. 2003a. "From Pipes to Scopes: The Flow Architecture of Financial Markets." *Distinktion: Journal of Social Theory* 4 (2):7–23. https://doi.org/10.1080/1600910X.2003.9672857.
———. 2003b. "From Pipes to Scopes: The Flow Architecture of Financial Markets." *Distinktion: Journal of Social Theory* 4 (2):7–23. https://doi.org/10.1080/1600910X.2003.9672857.
Cetina, Karin Knorr, and Urs Bruegger. 2002a. "Global Microstructures: The Virtual Societies of Financial Markets." *American Journal of Sociology* 107 (4):905–50.
———. 2002b. "Global Microstructures: The Virtual Societies of Financial Markets." *American Journal of Sociology* 107 (4):905–50.
———. 2002c. "Traders' Engagement with Markets." *Theory, Culture & Society* 19 (5–6):161–85. https://doi.org/10.1177/026327602761899200.
Chamberlin, Don. 2012. "Early History of SQL." *IEEE Ann. Hist. Comput.* 34 (4):78–82. https://doi.org/10.1109/MAHC.2012.61.
Chamberlin, Donald D., Morton M. Astrahan, Mike W. Blasgen, Jim Gray, W. Frank King III, Bruce G. Lindsay, Raymond A. Lorie, et al. 1981. "A History and Evaluation of System R." *Commun. ACM* 24 (10):632–46.
Chamberlin, Donald D., Raymond F. Boyce, and Irving L. Traiger. 1974. "A Deadlock-Free Scheme for Resource Locking in a Data-Base Environment." In *Information Processing 74: Proceedings of IFIP Congress 74*.
Chamberlin, Edward. 1933. *The Theory of Monopolistic Competition*. Cambridge, MA: Harvard University Press.

Chandler, Alfred D. 1977. *The Visible Hand : The Managerial Revolution in American Business*. Belknap Press.
Chaplin, Tamara. 2007. *Turning On the Mind: French Philosophers on Television*. University of Chicago Press.
Cheriton, D. R. 1984. “The V Kernel: A Software Base for Distributed Systems.” *IEEE Softw.* 1 (2):19–42. https://doi.org/10.1109/MS.1984.234046.
Cheriton, David R., and Willy Zwaenepoel. 1985. “Distributed Process Groups in the V Kernel.” *ACM Transactions on Computer Systems* 3 (2):77–107.
Chimisso, Cristina. 2000. “Introduction.” In *The Dialectic of Duration*, by Gaston Bachelard, 1–13. Rowman and Littlefield.
———. 2001. *Gaston Bachelard: Critic of Science and the Imagination*. Routledge.
Christie, William G., and Paul H. Schultz. 1994. “Why Do NASDAQ Market Makers Avoid Odd-Eighth Quotes?” *The Journal of Finance* 49 (5):1813–40.
Church, Alonzo. 1936. “A Note on the Entscheidungsproblem.” *The Journal of Symbolic Logic* 1 (1):40–41.
Clemson, Gaye I. 2012. *Tandem Computers Unplugged: A People’s History*. FastPencil.
Clos, C. 1953. “A Study of Non-Blocking Switching Networks.” *The Bell System Technical Journal* 32 (2):406–24. https://doi.org/10.1002/j.1538-7305.1953.tb01433.x.
Coase, R. H. 1937. “The Nature of the Firm.” *Economica* 4 (16):386–405.
Codd, E. F. 1970. “A Relational Model of Data for Large Shared Data Banks.” *Communications of the ACM* 13 (6):377–87.
———. 1971. “Normalized Data Structure: A Brief Tutorial.” In *Proceedings of 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control, San Diego, California, November 11-12, 1971*, edited by E. F. Codd and A. L. Dean, 1–17. ACM.
———. 1972. “Relational Completeness of Data Base Sublanguages.” In *Courant Computer Science Symposium 6 May 24-25 1971: Data Base Systems*, edited by Randall Rustin, 33–96. Prentice-Hall.
Codd, E. F., and C. J. Date. 1974. “Interactive Support for Non-Programmers: The Relational and Network Approaches.” In *Proceedings of the 1974 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control: Data Models: Data-Structure-Set versus Relational*, 11–41. SIGFIDET ’74. New York, NY, USA: ACM.
Coffee, John C. 2002. “Racing Towards the Top? The Impact of Cross-Listings and Stock Market Competition on International Corporate Governance.” *Columbia Law Review* 102 (7):1757–1831.
Cohen, Kalman J., Steven F. Maier, Robert A. Schwartz, and David K. Whitcomb. 1986. *The Microstructure of Securities Markets*. Englewood Cliffs, New Jersey: Prentice-Hall.
Cohen, Kalman J., and Robert A. Schwartz. 1989. “An Electronic Call Market: Its Design and Desirability.” In *The Challenge of Information Technology for the Securities Markets: Liquidity, Volatility, and Global Trading*, edited by Henry C. Lucas and Robert A. Schwartz. Dow Jones-Irwin.
Cohen, Molly, and Arun Sundararajan. 2015. “Self-Regulation and Innovation in the Peer-to-Peer Sharing Economy.” *University of Chicago Law Review* 82:116–33.
Comer, Douglas. 1979. “The Ubiquitous B-Tree.” *ACM Comput. Surv.* 11 (2):121–37.
Copeland, Duncan G., Richard O. Mason, and James L. McKenney. 1995. “Sabre: The Development of Information-Based Competence and Execution of Information-Based Competition.” *IEEE Annals of the History of Computing*, September, 30–57.

Copeland, Duncan G., and James L. McKenney. 1988. "Airline Reservations Systems: Lessons From History." *MIS Quarterly*, September.
Cortada, James W. 1993. *Before the Computer: IBM, NCR, Burroughs, and Remington Rand and the Industry They Created, 1865-1956*. Princeton University Press.
———. 1996. *Information Technology as Business History: Issues in the History and Managementof Computers*. ContributionsinEconomicsandEconomicHistory 177. GreenwoodPress.
———. 2003. *The Digital Hand: How Computers Changed the Work of American Manufacturing, Transportation, and Retail Industries*. Oxford University Press.
———. 2005. *The Digital Hand, Volume 2: How Computers Changed the Work of American Financial, Telecommunications, Media, and Entertainment Industries*. Oxford University Press.
———. 2007. *The Digital Hand, Vol 3: How Computers Changed the Work of American Public Sector Industries*. Oxford University Press.
Crocker, Stephen D. 2009. "How the Internet Got Its Rules." *The New York Times*, April 7, 2009, sec. Opinion. http://www.nytimes.com/2009/04/07/opinion/07crocker.html.
Cronon, William. 1992. *Nature's Metropolis: Chicago and the Great West*. Reprint edition. New York: W. W. Norton & Company.
Cybo-Ottone, Alberto, Carmine di Noia, and Maurizio Murgia. 2000. "Recent Development in the Structure of Securities Markets." In *Brookings-Wharton Papers on Financial Services: 2000*, edited by Robert E. Litan and Anthony M. Santomero, 223–73. Washington, D.C.: Brookings Institution Press.
Cypher, Robert, and Eric Leu. 1994. "The Semantics of Blocking and Nonblocking Send and Receive Primitives." In *Proceedings of the 8th International Symposium on Parallel Processing*, 729–735. Washington, DC, USA: IEEE Computer Society. http://dl.acm.org/citation.cfm?id=645604.662605.
Darwen, H. 2012. "The Relational Model: Beginning of an Era." *IEEE Annals of the History of Computing* 34 (4):30–37. https://doi.org/10.1109/MAHC.2012.50.
Date, C. J. 1975. *An Introduction to Database Systems*. The Systems Programming Series; Variation: Addison-Wesley Systems Programming Series. Reading, Mass.: Addison-Wesley Pub. Co.
Date, C. J., and E. F. Codd. 1974. "The Relational and Network Approaches: Comparison of the Application Programming Interfaces." In *Proceedings of the 1974 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control: Data Models: Data-Structure-Set versus Relational*, 83–113. SIGFIDET '74. New York, NY, USA: ACM.
Davies, Charles T. 1973. "Recovery Semantics for a DB/DC System." In *Proceedings of the ACM Annual Conference*, 136–41. New York, NY, USA: ACM.
Davies Jr, C. T. 1972. *A Recovery/Integrity Architecture for a Data System*.
Davis, Jeffry L. 1985. "The Intermarket Trading System and The Cincinnati Experiment." In *Market Making and the Changing Structure of the Securities Industry*. Lexington Books.
Davis, L. E., and J. C. Taylor. 1986. "Technology, Organization and Job Structure." Edited by R. Dubin. *Handbook of Work, Organization, and Society*, 379–419.
DeBlois, Diane, Robert Dalton Harris, and Sune Christian Pedersen. 2012. "Hermes: Message and Messenger." In *The Winton M. Blount Postal History Symposia: Select Papers, 2010–2011*, edited by Thomas Lera. Washington, D.C: Smithsonian Institution Press.

DeCarlo, C.R. 1955. "Application of Data Processors in Production." *Western Joint Computer Conference*.
Decock, Lieven. 2005. "The Taming of Change." In *After Whitehead: Rescher on Process Metaphysics*, edited by Michel Weber. Ontos Verlag.
Demsetz, Harold. 1968. "The Cost of Transacting." *The Quarterly Journal of Economics* 82 (1):33–53.
Derrida, Jacques. 1976. *Of Grammatology*. Johns Hopkins University Press.
Desmond, John. 1984. "NYSE System Bears up under Record Stock Trading." *Computerworld*, August.
Desrosières, Alain. 1998. *The Politics of Large Numbers: A History of Statistical Reasoning*. Harvard University Press.
DeWitt, David J., and Charles Levine. 2008. "Not Just Correct, but Correct and Fast: A Look at One of Jim Gray's Contributions to Database System Performance." *SIGMOD Record* 37 (2):45–49.
Dijkstra, E. W. 1965. *Cooperating Sequential Processes*.
DiMaggio, Paul, Eszter Hargittai, W Russell Neuman, and John Robinson. 2001. "Social Implications of the Internet Paul DiMaggio, Eszter Hargittai, W Russell Neuman, John Robinson." *Annual Review of Sociology* 27.
Dobbin, Frank. 1994. *Forging Industrial Policy*. Princeton University Press.
Dodge, George M. 1921. *The Telegraph Instructor*. Valparaiso, Ind. https//catalog.hathitrust.org/Record/008606131.
Dolgicer, Max. 1993. "The Middleware Muddle/, International Systems Group.Pdf." *Data Communications* 22 (14):33.
Dolgopolov, Stanislav. 2014. "The Maker-Taker Pricing Model and Its Impact on the Securities Market Structure: A Can of Worms for Securities Fraud?" *Virginia Law Review* 8 (2):231–72.
Domowitz, Ian. 1990. "The Mechanics of Automated Trade Execution Systems." *Journal of Financial Intermediation* 1:167–94.
———. 1995. "Electronic Derivatives Exchanges: Implicit Mergers, Network Externalities and Standardization." *Quarterly Review of Economics and Finance* 35 (4):163–75.
Domowitz, Ian, and Ruben Lee. 2001. "On the Road to Reg ATS." *International Finance* 4 (2):279–302.
Domowitz, Ian, and Benn Steil. 1999. "Automation, Trading Costs, and the Structure of the Securities Trading Industry." In *Brookings-Wharton Papers on Financial Services: 1999*, edited by Robert E. Litan and Anthony M. Santomero, 33–92.
Dornian, Alan. 1994. "ReserVec: Trans-Canada Air Lines' Computerized Reservation System." *IEEE Annals of the History of Computing* 16 (2):31–41.
Du Boff, Richard B. 1980. "Business Demand and the Development of the Telegraph in the United States, 1844-1860." *The Business History Review* 54 (4):459–79.
DuBoff, Richard B. 1982. "The Telegraph and the Structure of Markets in the United States, 1845-1890." In *Research in Economic History*, edited by Paul Uselding, 8:253–78. Greenwich, CT: JAI Press Inc.
Durr, Kenneth. 2007. *Securities and Exchange Commission Historical Society Interview with Donald Calvin*.
Edwards, Paul N. 1996. *The Closed World: Computers and the Politics of Discourse in Cold War America*. MIT Press.

Einstein, A. 1905. “Zur Elektrodynamik Bewegter Körper.” *Annalen Der Physik* 322 (10):891–921. https://doi.org/10.1002/andp.19053221004.
Eisenstein, Elizabeth L. 2005. *The Printing Revolution in Early Modern Europe*. 2nd ed. Cambridge University Press.
Elias, Norbert. 1939. *The Civilizing Process*. Blackwell Publishing.
Emirbayer, Mustafa, and Ann Mische. 1998. “What Is Agency?” *American Journal of Sociology* 103 (4):962–1023. https://doi.org/10.1086/231294.
Engles, Robert W. 1971. “An Analysis of the April 1971 Data Base Task Group Report.” In *Proceedings of the 1971 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control*, 69–91. SIGFIDET ’71. New York, NY, USA: ACM.
Eswaran, K. P., J. N. Gray, R. A. Lorie, and I. L. Traiger. 1976. “The Notions of Consistency and Predicate Locks in a Database System.” *Communications of the ACM*, no. 11 (November).
European Union Committee. 2016. “Online Platforms and the Digital Single Market.” House of Lords.
Evans, David S. 2003. “The Antitrust Economics of Multi-Sided Platform Markets.” *Yale Journal on Regulation* 20 (2):325–81.
———. 2011. *Platform Economics: Essays on Multi-Sided Businesses*. Competition Policy International.
Evans, David S., Andrei Hagiu, and Richard Schmalensee. 2006. *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*. MIT Press.
Evans, David S., and Richard Schmalensee. 2016. *Matchmakers: The New Economics of Multisided Platforms*. Boston, Massachusetts: Harvard Business Review Press.
Everett, R.R., C.A. Zraket, and H.D. Benington. 1957. “SAGE — A Data-Processing System for Air Defense.” *Proceedings of the Eastern Joint Computer Conference*, 148–55.
Fischer, Michael M.J., and Alyssa Gabbay. 2009. “Bazaars.” In *The Oxford Encyclopedia of the Islamic World*.
Flaherty, Dermot. 2011. *WebSphere® MQ – A Short History of the Past 15 Years*. http://web.archive.org/web/20110102005405/http://www-01.ibm.com/software/integration/wmq/MQ15Anniversary.html.
Fligstein, Neil. 1996. “Markets as Politics: A Political-Cultural Approach to Market Institutions.” *American Sociological Review* 61 (4):656–73.
Fligstein, Neil, and Luke Dauter. 2007. “The Sociology of Markets.” *Annual Review of Sociology* 33 (1):105–28.
Flood, Mark, Allan I. Mendelowitz, and William Nichols. 2013. “Monitoring Financial Stability in a Complex World.” In *Financial Analysis and Risk Management: Data Governance, Analytics and Life Cycle Management*, edited by Victoria Lemieux, 15–46. Springer-Verlag.
Foucault, Thierry. 2012. “Pricing Liquidity in Electronic Markets.” Foresight Driver Review 18. The Future of Computer Trading in Financial Markets. London: United Kingdom Government Office for Science.
Fourcade, Marion. 2007. “Theories of Markets and Theories of Society.” *American Behavioral Scientist* 50 (8):1015–34.
———. 2009. *Economists and Societies: Discipline and Profession in the United States, Britain, and France, 1890s to 1990s*. Princeton University Press.
Frana, Philip L. 2003. *An Interview with Jim Gray, 3 Jan. 2002*. http://purl.umn.edu/113899.

Frege, Gottlob. 1879. “Begriffsschrift, a Formula Language, Modeled upon That of Arithmetic, for Pure Thought.” In *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*, edited by Jean van Heijenoort, 1–83.
Friedman, Daniel. 1993. “The Double Auction Market Institution: A Survey.” In *The Double Auction Market: Institutions, Theories and Evidence*, edited by Daniel Friedman and Jon Rust. Reading, MA: Addison-Wesley.
Friess, Bernhard, and Sean Greenaway. 2006. “Competition in EU Trading and Post-Trading Service Markets.” *Competition Policy International* 2 (1).
FTC. 2015. “The ‘Sharing’ Economy: Issues Facing Platforms, Participants, and Regulators: June 9, 2015 Workshop Transcript.” Federal Trade Commission.
Gadrey, Jean. 2000. “The Characterization of Goods and Services: An Alternative Approach.” *Review of Income and Wealth* 46 (3):369–87.
Galison, Peter. 2003. *Einstein’s Clocks, Poincaré’s Maps: Empires of Time*. New York: W.W. Norton and Company.
Galison, Peter Louis. 1979. “Minkowski’s Space-Time: From Visual Thinking to the Absolute World.” *Historical Studies in the Physical Sciences* 10:85–121. https://doi.org/10.2307/27757388.
Galloway, Alexander R. 2010. “Bernard Stiegler, or Our Thoughts Are With Control.” In *French Theory Today: An Introduction to Possible Futures*.
Gaukroger, Stephen W. 1976. “Bachelard and the Problem of Epistemological Analysis.” *Studies in History and Philosophy of Science Part A* 7 (3):189–244.
Geertz, Clifford. 1978. “The Bazaar Economy: Information and Searching in Peasant Marketing.” *Supplement to the American Economic Review* 68 (2):28–32.
Gibson, George Rutledge. 1889. *The Stock Exchanges of London, Paris, and New York: A Comparison*. G.P. Putnam.
Gilfillan, S. C. 1935. *The Sociology of Invention*. Follett Publishing Company.
Gille, Bertrand. 1986. *History of Techniques*. Gordon and Breach Science Publishers.
Glaeser, Andreas. 2010. *Political Epistemics: The Secret Police, the Opposition, and the End of East German Socialism*. Chicago, IL: University of Chicago Press.
Godin, Benoît. 2010. *Innovation Without the Word: William F. Ogburn’s Contribution to Technological Innovation Studies*.
Goetzmann, William N. 2016. *Money Changes Everything: How Finance Made Civilization Possible*. Princeton: Princeton University Press.
Goetzmann, William N., and K. Geert Rouwenhorst, eds. 2005. *The Origins of Value: The Financial Innovations That Created Modern Capital Markets*. Oxford University Press.
Goffman, Erving. 1953. “Communication Conduct in an Island Community.” University of Chicago.
———. 1983. “The Interaction Order.” *American Sociological Review* 48 (1):1–17.
Grad, Burton. 2007. *RDBMS Workshop: IBM*. Computer History Museum.
Graeber, David. 2001. *Toward An Anthropological Theory of Value: The False Coin of Our Dreams*. Palgrave.
Granovetter, Mark. 1985. “Economic Action and Social Structure: The Problem of Embeddedness.” *American Journal of Sociology* 91 (3):481–510.
Gray, J., R. Lorie, G. F. Putzolu, and I. L. Traiger. 1976. “Granularity of Locks and Degrees of Consistency in a Shared Data Base.” In *Modeling in Data Base Management Systems*, edited by G. M. Nijssen, 364–94. North Holland Pub.

Gray, Jim. 1970. “Locking.” In *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation*, edited by Jack B. Dennis, 169–176. New York, NY, USA: ACM.
———. 1978. “Notes on Data Base Operating Systems.” In *Operating Systems: An Advanced Course*. Vol. 60. Springer-Verlag Lecture Notes in Computer Science. Springer-Verlag.
———. 1981. “The Transaction Concept: Virtues and Limitations (Invited Paper).” In *Very Large Data Bases, 7th International Conference, September 9-11, 1981, Cannes, France, Proceedings*, 144–54. IEEE Computer Society.
———. 1990. “A Census of Tandem System Availability between 1985 and 1990.” *IEEE Trans. Reliability* 34 (October):409–18.
Gray, Jim, Butler Lampson, Bruce Lindsay, and Harold Sturgis. 1972. *The Control Structure of an Operating System*.
Gray, Jim, and Charles Levine. 2005. *Thousands of DebitCredit Transactions-Per-Second: Easy and Inexpensive*.
Gray, Jim, Paul McJones, Mike Blasgen, Bruce Lindsay, Raymond Lorie, Tom Price, Franco Putzolu, and Irving Traiger. 1981. “The Recovery Manager of the System R Database Manager.” *ACM Comput. Surv.* 13 (2):223–242.
Gray, Jim, and Andreas Reuter. 1992. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann.
Gray, Jim, and Daniel P. Siewiorek. 1991. “High-Availability Computer Systems.” *Computer*, September.
Gruber, Thomas R. 1993. *A Translation Approach to Portable Ontology Specifications*.
Haerder, Theo, and Andreas Reuter. 1983. “Principles of Transaction-Oriented Database Recovery.” *ACM Computing Surveys* 15:287–317.
Hagino, Tatsuya. 1987a. *A Categorical Programming Language*.
———. 1987b. *A Categorical Programming Language*.
Hagiu, Andrei, and Julian Wright. 2015. “Multi-Sided Platforms.” *International Journal of Industrial Organization* 43:162–74.
Haigh, Thomas. 2001a. “Inventing Information Systems: The Systems Men and the Computer, 1950-1968.” *Business History Review* 75:15–61.
———. 2001b. “The Chromium-Plated Tabulator: Institutionalizing an Electronic Revolution, 1954-1958.” *IEEE Annals of the History of Computing*, October, 75–104.
———. 2006. “A Veritable Bucket of Facts: Origins of the Data Base Management System.” *SIGMOD Record* 35 (2):33–49.
———. 2007. *Oral History of C. J. Date*. Computer History Museum.
———. 2009. “How the Data Got Its Base: Information Storage Software in the 1950s and 1960s.” *IEEE Annals in the History of Computing*, 7–25.
———. 2011. “The History of Information Technology.” *Annual Review of Information Science and Technology* 45 (1):431?487.
———. 2014. “Actually, Turing Did Not Invent the Computer.” *Communications of the ACM* 57 (1):38–41.
Hall, Richard H. 1963. “The Concept of Bureaucracy: An Empirical Assessment.” *American Journal of Sociology* 69 (1):32–40.
Hansen, Per Brinch. 1967. “The RC 4000 Real-Time Control System at Pulawy.” *BIT Numerical Mathematics* 7 (4):279–88. https://doi.org/10.1007/BF01939322.

———. 2000. “The Evolution of Operating Systems.” In *Classic Operating Systems: From Batch Processing to Distributed Systems*, edited by Per Brinch Hansen. Springer New York.
Härder, Theo, and Wolfgang Lehner, eds. 2005. *Data Management in a Connected World*. Springer.
Hardie, Iain. 2004. “‘The Sociology of Arbitrage’: A Comment on MacKenzie.” *Economy and Society* 33 (2):239–54.
Harris, Larry. 1991. *Liquidity, Trading Rules, and Electronic Trading Systems*. NYU Stern.
———. 1996. *The Economics of Best Execution*. http://www-bcf.usc.edu/~lharris/ACROBAT/BESTEXEC.PDF.
———. 2003. *Trading and Exchanges: Market Microstructure for Practitioners*. New York: Oxford University Press.
———. 2010. “The Economics of Trading and of Regulated Exchanges.” In *Regulated Exchanges: Dynamic Agents of Economic Growth*, edited by Larry Harris, 70–108. Oxford: Oxford University Press.
Hasbrouck, Joel. 2007. *Empirical Market Microstructure: The Institutions, Economics and Econometrics of Securities Trading*. New York: Oxford University Press.
———. 2014. “Securities Trading: Procedures and Principles. Draft Teaching Notes.”
Hauser, P., and O. D. Duncan, eds. 1959. “Human Ecology and Population Studies.” In *The Study of Population*, 678–716. University of Chicago Press.
Hautcoeur, Pierre-Cyrille, and Angelo Riva. 2012. “The Paris Financial Market in the 19th Century: Complementarities and Competition in Microstructures.” *Economic History Review* 65 (4):1326–53.
Hayward, Mark, and Bernard Dionysius Geoghegan. 2012. “Introduction: Catching Up With Simondon.” *SubStance* 41 (3):3–15.
Head, Robert V. 2002. “Getting Sabre off the Ground.” *IEEE Annals of the History of Computing*.
Hecht, Gabrielle, and Paul Edwards. 2010. “The Technopolitics of Cold War: Toward a Transregional Perspective.” In *Essays on Twentieth-Century History*, edited by Michael Adas, 271–314. Philadelphia, PA: Temple University Press.
Heide, Lars. 2009a. *Punched-Card Systems and the Early Information Explosion, 1880-1945*. Johns Hopkins University Press.
———. 2009b. *Punched-Card Systems and the Early Information Explosion, 1880-1945*. Johns Hopkins University Press.
Heijenoort, Jean van. 1967. *From Frege to Godel: A Source Book in Mathematical Logic*. First Edition edition. Cambridge, Mass: Harvard University Press.
Held, G. D., M. R. Stonebraker, and E. Wong. 1975. “INGRES: A Relational Data Base System.” In *Proceedings of the May 19-22, 1975, National Computer Conference and Exposition*, 409–416. AFIPS ’75. New York, NY, USA: ACM. https://doi.org/10.1145/1499949.1500029.
Hill, Peter. 1977. “On Goods and Services.” *The Review of Income and Wealth* 23 (4):315–38.
Hochfelder, David. 2006. “‘Where the Common People Could Speculate’: The Ticker, Bucket Shops, and the Origins of Popular Participation in Financial Markets, 1880-1920.” *The Journal of American History* 93 (2):335–58. https://doi.org/10.2307/4486233.
———. 2012. *The Telegraph in America, 1832–1920*. Johns Hopkins Studies in the History of Technology. Baltimore: Johns Hopkins University Press.

Holzer, Boris, and Yuval Millo. 2005. "From Risks to Second-Order Dangers in Financial Markets: Unintended Consequences of Risk Management Systems." *New Political Economy* 10 (2):223–45. https://doi.org/10.1080/13563460500144777.
Houston, Peter. 1998. "Choosing between the Synchronous and Asynchronous." *Financial MiddlewareSpectra* 12 (3):30–35.
IBM Software Group. 2004. "CICS: An Introduction."
"IBM, Stratus, DEC Talk To Each Other With DMQ." 1988. *Sell-Side Technology*, September.
Israel, J., J. Mitchell, and H. Sturgis. 1978. "Issues in the Design and Use of a Distributed File System." Xerox Corporation.
Jakobson, Roman. 1960. "Closing Statement: Linguistics and Poetics." In *Style in Language*, edited by Thomas A. Sebeok, 350–77. Cambridge, MA: MIT Press.
Jakobson, Roman, C. Gunnar M. Fant, and Morris Halle. 1961. *Preliminaries to Speech Analysis: The Distinctive Features and Their Correlates*. Cambridge, Mass: The MIT Press.
Jardine, Donald A., ed. 1974. *Proceedings of the SHARE Working Conference on Data Base Management Systems, Montreal, Canada, July 23-27, 1973*. North-Holland.
Jarema, D. R., and E. H. Sussenguth. 1981. "IBM Data Communications: A Quarter Century of Evolution and Progress." *IBM Journal of Research and Development* 25 (5):391–404. https://doi.org/10.1147/rd.255.0391.
Joas, Hans. 1997. *G. H. Mead: A Contemporary Re-Examination of His Thought*. Reprint edition. Cambridge, Mass: The MIT Press.
Joel, A.E. 1982. "A Better Understanding of Switching." In *A History of Engineering and Science in the Bell System: Switching Technology (1925-1975)*, 87–118. Bell Telephone Laboratories, Inc.
Johnson, P. R., and R. Thomas. 1975. "Maintenance of Duplicate Databases." January 27, 1975. https://tools.ietf.org/html/rfc677.
Kahn, R., D. Walden, F. Heart, S. Ornstein, and W. Crowther. 1970. "The Interface Message Processor for the ARPA Computer Network." In *Managing Requirements Knowledge, International Workshop On*, 551. Los Alamitos, CA, USA: IEEE Computer Society. https://doi.org/10.1109/AFIPS.1970.69.
Kant, Immanuel. 1965. *Critique of Pure Reason*. St. Martin's Press.
Karmel, Roberta S. 2002a. "Regulation ATS and Related Market Structure Issues in the United States." In *Exchanges and Alternative Trading Systems*, edited by Dick Frase and Helen Parry, 81–96. London: Sweet & Maxwell.
———. 2002b. "Turning Seats into Shares: Causes and Implications of Demutualization of Stock and Futures Exchanges." *Hastings Law Journal* 367.
Keith, Christopher, and Allan Grody. 1988. "Electronic Automation at the New York Stock Exchange." In *Managing Innovation: Cases from the Services Industries*, edited by Bruce R. Guile and James Brian Quinn, 82–107. Washington, D.C.: National Academy Press.
Khademian, Anne. 1992. *The SEC and Capital Market Regulation: The Politics of Expertise*. Pittsburgh, PA: University of Pittsburgh Press.
Kirk, Robert A. van. 1989. "How SIAC Survived." In *The Challenge of Information Technology for the Securities Markets: Liquidity, Volatility, and Global Trading*, edited by Jr. Henry C. Lucas and Robert A. Schwartz.
Kitchin, Rob. 2014. "Big Data, New Epistemologies and Paradigm Shifts." *Big Data & Society*, June, 1–12.

Kline, Ronald. 1995. "Construing 'Technology' as 'Applied Science': Public Rhetoric of Scientists and Engineers in the United States, 1880-1945." *Isis* 86 (2):194–221.
Kline, Ronald R. 2006. "Cybernetics, Management Science, and Technology Policy: The Emergence of 'Information Technology' as a Keyword, 1948-1985." *Technology and Culture* 47 (3):513–35.
Knuth, Donald. 1973. *The Art of Computer Programming, Vol 3: Sorting and Searching*. Addison-Wesley.
Kockelman, Paul. 2005. "The Semiotic Stance." *Semiotica* 157 (1):233–304.
———. 2006. "Agent, Person, Subject, Self." *Semiotica* 162 (1):1–18.
———. 2013. *Agent, Person, Subject, Self: A Theory of Ontology, Interaction, and Infrastructure*. Oxford University Press.
———. 2015. "Four Theories of Things: Aristotle, Marx, Heidegger, and Peirce." *Signs and Society* 3 (1).
Kollock, Peter. 1999. "The Production of Trust in Online Markets." In *Advances in Group Processes*, edited by E. J. Lawler, M. Macy, S. Thyne, and H. A. Walker. Vol. 16. JAI Press.
Kondo, Annette, and Ignatius Chithelen. 1988. "Financial Workstations Emerge as 'Modern Marvels.'" *Wall Street Computer Review*, December, 42–64, 90–91.
Krippner, Greta R. 2012. *Capitalizing on Crisis: The Political Origins of the Rise of Finance*. Cambridge, MA: Harvard University Press.
Lagneau-Ymonet, Paul, and Angelo Riva. 2015. *Histoire de La Bourse*. La Découverte.
Lamport, Leslie. 1978. "Time, Clocks, and the Ordering of Events in a Distributed System." *Commun. ACM* 21 (7):558–565. https://doi.org/10.1145/359545.359563.
Laplante, Phillip A., Eileen P. Rose, and Maria Gracia-Watson. 1995. "An Historical Survey of Early Real-Time Computing Developments in the U.S." *Real-Time Systems* 8 (2–3):199–213. https://doi.org/10.1007/BF01094343.
Latour, Bruno. 1986. "Visualisation and Cognition: Drawing Things Together." In *Knowledge and Society Studies in the Sociology of Culture Past and Present*, edited by H. Kuklick, 1–40. Greenwich, CT: Jai Press.
———. 1987. *Science in Action*. Harvard University Press.
Lee, Ruben. 1998. *What Is an Exchange?: The Automation, Management and Regulation of Financial Markets*. New York: Oxford University Press.
———. 2002. "The Future of Securities Exchanges." In *Brookings-Wharton Papers on Financial Services: 2002*, 1–33. Washington, D.C.: Brookings Institution Press.
LeLann, Gerard. 1981. "Chapter 1. Motivations, Objectives and Characterization of Distributed Systems." In *Distributed Systems — Architecture and Implementation*, 1–9. Springer. https://link-springer-com.proxy.uchicago.edu/chapter/10.1007/3-540-10571-9_1.
Lenglet, Marc. 2011. "Conflicting Codes and Codings How Algorithmic Trading Is Reshaping Financial Regulation." *Theory, Culture & Society* 28 (6):44–66.
Lenoir, Timothy. 1998. "Inscription Practices and Materialities of Communication." In *Inscribing Science: Scientific Texts and the Materiality of Communication*, edited by Timothy Lenoir, 1–19. Stanford University Press.
Leroi-Gourhan, André. 1993. *Gesture and Speech*. The MIT Press.
Lévi-Strauss, Claude. 1962. *The Savage Mind*. University of Chicago Press.
Levy, Steven. 1991. "A Spreadsheet Way of Knowledge." In *Computers in the Human Context*, edited by Tom Forester, 318–26. MIT Press.

Lewis, Michael. 2015. *Flash Boys*. 01 edition. Penguin.
Licklider, J.C.R. 1960. "Man-Computer Symbiosis." *IRE Transactions on Human Factors in Electronics*, March.
Lindsay, Bruce G. 2008. "Jim Gray at IBM: The Transaction Processing Revolution." *SIGMOD Record* 37 (2).
Liskov, Barbara. 1979. "Primitives for Distributed Computing." In *Proceedings of the Seventh ACM Symposium on Operating Systems Principles*, 33–42. SOSP '79. New York, NY, USA: ACM. https://doi.org/10.1145/800215.806567.
Lomet, D. B. 1977. "Process Structuring, Synchronization, and Recovery Using Atomic Actions." In *ACM SIGSOFT Software Engineering Notes - Proceedings of an ACM Conference on Language Design for Reliable Software*.
Lucas, J.R. 2000. *The Conceptual Roots of Mathematics*. International Library of Philosophy. Routledge.
Mac Lane, Saunders. 1986. *Mathematics Form and Function*. Springer-Verlag.
Macey, Jonathan, and Hideki Kanda. 1989. "The Stock Exchange as a Firm: The Emergence of Close Substitutes for the New York and Tokyo Stock Exchanges." *Cornell Law Review* 75 (5):1006.
Macey, Jonathan R., and David D. Haddock. 1985. "Shirking at the SEC: The Failure of the National Market System." *University of Illinois Law Review* 1985 (2):315–62.
Macey, Jonathan R., and Maureen O'Hara. 1997. "The Law and Economics of Best Execution." *Journal of Financial Intermediation* 6 (3):188–223.
Macey, Jonathan R., and O'Hara, Maureen. 2005. "From Markets to Venues: Securities Regulation in an Evolving World." http://digitalcommons.law.yale.edu/cgi/viewcontent.cgi?article=2403&context=fss_papers.
MacKenzie, Donald. 1984. "Marx and the Machine." *Technology and Culture*.
———. 1993. "Negotiating Arithmetic, Constructing Proof: The Sociology of Mathematics and Information Technology." *Social Studies of Science*.
———. 2001. *Mechanizing Proof: Computing, Risk, and Trust*. MIT Press.
———. 2002. "A View from the Sonnenbichl: On the Historical Sociology of Software and System Dependability." In *History of Computing: Software Issues*, edited by Ulf Hashagen, Reinhard Keil-Slawik, and Arthur L. Norberg, 97–122.
———. 2004. "The Big, Bad Wolf and the Rational Market: Portfolio Insurance, the 1987 Crash and the Performativity of Economics." *Economy and Society* 33 (3):303–34.
———. 2005. "Opening the Black Boxes of Global Finance." *Review of International Political Economy*, October, 555–76.
———. 2006. *An Engine, Not a Camera: How Financial Models Shape Markets*. MIT Press.
———. 2009. "Derivatives: The Production of Virtuality." In *Material Markets: How Economic Agents Are Constructed*, 63–84.
———. 2012. "Mechanizing the Merc: The Chicago Mercantile Exchange and the Rise of High-Frequency Trading," October.
———. 2014. "A Sociology of Algorithms: High-Frequency Trading and the Shaping of Markets."
MacKenzie, Donald, Daniel Beunza, and Iain Hardie. 2009. "The Material Sociology of Arbitrage." In *Material Markets: How Economic Agents Are Constructed*, by Donald MacKenzie and Iain Hardie.

MacKenzie, Donald, and Juan Pablo Pardo-Guerra. 2014. "Insurgent Capitalism: Island, Bricolage and the Re-Making of Finance." *Economy and Society* 43 (2):153–82.
Macneil, Ian R. 1973. "The Many Futures of Contracts." *Southern California Law Review* 47:691–816.
Madhavan, Ananth. 2000. "Market Microstructure: A Survey." *Journal of Financial Markets* 3 (3):205–58.
Mahoney, Michael S. 2005. "The History of Computing(S)." *Interdisciplinary Science Reviews* 3 (2):119–35.
Mancke, Richard B., Franklin M. Fisher, and James W. McKie. 1980. "US vs IBM Exhibit 14971, Part 1." https://archive.org/details/bitsavers_charlesRivit14971part1Jul80_14957277.
Markham, Jerry W., and Daniel J. Harty. 2008. "For Whom the Bell Tolls: The Demise of Exchange Trading Floors and the Growth of ECNs." *Journal of Corporation Law* 33 (4):865–939.
Martin, Ian. 2012. "Too Far Ahead of Its Time: Barclays, Burroughs, and Real-Time Banking." *IEEE Annals of the History of Computing* 34 (2):5–19.
Martin, J. 1965. *Programming Real-Time Computer Systems*. Prentice-Hall.
Marx, Karl. 1990. *Capital, Vol. 1*. Harmondsworth.
Mauss, Marcel. 1934. "Techniques of the Body." *Economy and Society* 2 (1):70–88.
———. 1954. *The Gift*.
Mayo-Wilson, Conor. 2011. "Peirce and Brouwer." http://www.mayowilson.org/Papers/Peirce_Brouwer.pdf.
McGee, W. M. 1959. "Generalization: Key to Successful Electronic Data Processing." *Journal of the ACM* 6 (1):1–23.
McGee, William C. 2009. "The Information Management System (IMS) Program Product." *IEEE Annals of the History of Computing*, October, 66–75.
McJones, Ed Paul. 1995. *The 1995 SQL Reunion: People, Project, and Politics, May 29, 1995*.
Mead, George H., and Merritt H. Moore. 1936. "Science Raises Problems for Philosophy: Vitalism; Henri Bergson." In *Movements of Thought in the Nineteenth Century*, 292–325. Chicago IL: The University of Chicago Press.
Mead, G.H. 1932. *The Philosophy of the Present*. Open Court.
Mellet, Kevin, Thomas Beauvisage, Jean-Samuel Beuscart, and Marie Trespeuch. 2014. "A 'Democratization' of Markets? Online Consumer Reviews in the Restaurant Industry." *Valuation Studies* 2 (1):5–41.
Mendelson, Morris, Junius W. Peake, and Jr. R.T. Williams. 1979. "Toward a Modern Exchange: The Peake-Mendelson-Williams Proposal for an Electronically Assisted Auction Market." In *Impending Changes for Securities Markets: What Role for the Exchanges?*, edited by Ernest Bloch and Robert A. Schwartz. Jai Press Inc.
Metcalfe, R.M., and D.R. Boggs. 1976. "Ethernet: Distributed Packet Switching for Local Computer Networks." *Communications of the ACM* 19:395–403.
Michie, Ranald C. 1986. "The London and New York Stock Exchanges, 1850-1914." *The Journal of Economic History* 46 (1):171–87.
Michie, R.C. 1999. *The London Stock Exchange*. Oxford University Press.
Miller, Michael W. 1987. "Were Computers a Help or a Hindrance? Securities Industry Asks After the Crash." *Wall Street Journal*, December.

Miller, Sam Scott. 1985. "Self-Regulation of the Securities Markets: A Critical Examination." *Washington and Lee Law Review* 42 (3):853–87.
Millikin, Michael D. 1992. "An Object-Oriented Middleware Strategy That Works." *Data Communications* 21 (6):35.
Millo, Yuval, Fabian Muniesa, Nikiforos S. Panourgias, and Susan V. Scott. 2005a. "Organised Detachment: Clearinghouse Mechanisms in Financial Markets." *Information and Organization* 15:229–46.
———. 2005b. "Organised Detachment: Clearinghouse Mechanisms in Financial Markets." *Information and Organization*, Technology as Organization/Disorganization, 15 (3):229–46. https://doi.org/10.1016/j.infoandorg.2005.02.003.
Mills, Nat. 1972. "NASDAQ—A User-Driven, Real-Time Transaction System." *Spring Joint Computer Conference*, 1197–1206.
Minkowski, Hermann. 2012. "Space and Time." In *Space and Time: Minkowski's Papers on Relativity*, edited by Vesselin Petkov, 1 edition, 37–52. Minkowski Institute Press.
Mitcham, Carl. 1994. *Thinking About Technology: The Path Between Engineering and Philosophy*. Chicago, IL: University of Chicago Press.
Moore, Robbie. 2016. "Ticker Tape and the Superhuman Reader." In *Writing, Medium, Machine*, edited by Sean Pryor and David Trotter. Open Humanities Press CIC.
Morrison, Alan D., and William J. Wilhelm Jr. 2008. "The Demise of Investment Banking Partnerships: Theory and Evidence." *The Journal of Finance* 63 (1):311–50.
Mounce, D.C. 1994. *CICS: A Light Hearted Chronicle*. Winchester: IBM U.K. Laboratories.
Mueller, Milton. 1989. "The Switchboard Problem: Scale, Signaling, and Organization in Manual Telephone Switching, 1877-1897." *Technology and Culture* 30 (3):534–60. https://doi.org/10.2307/3105950.
Mumford, Lewis. 1934. *Technics and Civiization*. London: Lowe and Brydon.
Muniesa, Fabian. 2000. "Un Robot Walrasien. Cotation Électronique et Justesse Découverte Des Prix." *Politix* 13 (52):121–54.
———. 2011. "Is a Stock Exchange a Computer Solution?: Explicitness, Algorithms and the Arizona Stock Exchange." *International Journal of Actor-Network Theory and Technological Innovation (IJANTTI)* 3 (1):1–15. https://doi.org/10.4018/jantti.2011010101.
Muniesa, Fabian, Dominique Chabert, Marceline Ducrocq-Grondin, and Susan V. Scott. 2011. "Back-Office Intricacy: The Description of Financial Objects in an Investment Bank." *Industrial and Corporate Change* 20 (4):1189–1213.
National Academy of Engineering. 1973. "Telecommunications Research in the United States and Selected Foreign Countries : A Preliminary Survey. Volume II, Individual Contributions : Report to the National Science Foundation. Panel on Telecommunications Research, Committee on Telecomunications, National Academy of Engineering." Washington: National Academy of Engineering.
NATO. 1970. "Software Engineering Techniques: Report on a Conference Sponsored by the NATO Science Committee." Edited by J.N. Buxton and Brian Randell. Scientific Affairs Division.
Naur, Peter, and Brian Randell. 1969. "Software Engineering; Report on a Conference Sponsored by the NATO Science Committee, 7–11 October 1968." NATO.

Nayak, Ajit. 2016. "Heraclitus (540-480 BC)." In *The Oxford Handbook of Process Philosophy and Organization Studies*, edited by Jenny Helin, Tor Hernes, and Daniel Hjorth, Reprint edition, 32–47. Oxford New York, NY: Oxford University Press.

Nelson, Bruce. 1981. "Remote Procedure Call." Palo Alto Research Center.

Newsbytes News Network. 1993. "IBM 'Middleware' Links Applications."

Nijssen, G. M. 1974. "Data Structuring in the DDL and Relational Data Model." In *Data Base Management: Proceedings of the IFIP Working Conference Data Base Management*, edited by J. W. Klimbie and K. L. Koffeman, 363–84.

Nolan, Richard L. 2000. "Information Technology Management Since 1960." In *A Nation Transformed by Information*, edited by Alfred Chandler and James Cortada, 217–56. Oxford University Press.

Norberg, Arthur L. 1990. "Punched Card Machinery in Business and Government." *Technology and Culture* 31 (4).

Norman, Peter. 2011. *The Risk Controllers Central Counterparty Clearing in Globalised Financial Markets /*. Wiley.

NYSE. 1971. *Crisis in the Securities Industry; a Chronology: 1967-70*. New York: New York Stock Exchange.

Odders-White, Elizabeth R. 2004. "Third Market Reforms: The Overlooked Goal of the SEC's Order Handling Rules." *Journal of Financial and Quantitative Analysis* 39 (2):277–304.

Ogburn, William F., and S. C. Gilfillan. 1933. "The Influence of Invention and Discovery." In *Recent Social Trends in the United States: Report of the President's Research Committee on Social Trends*. McGraw Hill.

Oldenziel, Ruth. 1999. *Making Technology Masculine: Men, Women and Modern Machines in America, 1870-1945*. Amsterdam University Press.

Orlikowski, Wanda J. 1992. "The Duality of Technology: Rethinking the Concept of Technology in Organizations." *Organization Science* 3 (3).

Orlikowski, Wanda J., and Susan V. Scott. 2008. "Sociomateriality: Challenging the Separation of Technology, Work and Organization." *Academy of Management Annals* 2 (1):433–74.

Pacewicz, Josh. 2013. "Tax Increment Financing, Economic Development Professionals, and the Financialization of Urban Politics." *Socio-Economic Review* 11 (3):413–40.

Papadimitriou, Christos H. 1979. "The Serializability of Concurrent Database Updates." *Journal of the ACM* 26 (4):631–53.

Pardo-Guerra, Juan Pablo. 2011. "The Automated House: The Digitalization of the London Stock Exchange, 1955-1990." In *Technological Innovation in Retail Finance: International Historical Perspectives*, edited by Bernardo Bátiz-Lazo, J. Carles Maixé-Altés, and Paul Thomes. New York: Routledge.

———. 2014. *Making Markets: Infrastructures, Engineers and the Moral Technologies of Finance*. https://pardoguerra.files.wordpress.com/2014/11/141124-making-markets2.pdf.

Pardo-Guerra, Juan Pablo, Daniel Beunza, Yuval Millo, and Donald MacKenzie. 2010. "Impersonal Efficiency and the Dangers of a Fully Automated Securities Exchange." Monograph. Foresight. 2010. http://www.bis.gov.uk/foresight/our-work/projects/current-projects/computer-trading/working-paper.

Pardo-Guerra, Juan Pablo, and Donald MacKenzie. 2014. *The Politics of Fragmentation: Liberalism, Market Equality, and the Technological Re-Configuration of American Finance*. https://pardoguerra.files.wordpress.com/2014/03/pardo-guerra-and-mackenzie-the-politics-of-fragmentation1.pdf.

Parker, Geoffrey, Marshall W. Van Alstyne, and Sangeet Paul Choudary. 2016. *Platform Revolution: How Networked Markets Are Transforming the Economy—And How to Make Them Work for You*. New York: W.W. Norton and Company.
Parker, R.W. 1965. "The SABRE System." *Datamation*, September.
Peirce, Charles S. 1931. *Collected Papers of Charles Sanders Peirce*. Vol. 2.
Perrow, Charles. 1986. *Complex Organizations: A Critical Essay*. 3rd ed. McGraw-Hill.
Peterson, James L. 1979. "Notes on a Workshop on Distributed Computing." *ACM SIGOPS Operating Systems Review* 13 (3):18–30.
Petruschell, R. L., D. J. Dreyfus, L. E. Knollmeyer, and J. Y. Lu. 1970. *Reducing Costs of Stock Transactions : A Study of Alternative Trade Completion Systems. Vol. I: Summary of Results*. Rand.
Phister, Montgomery. 1976. *Data Processing Technology and Economics*. Santa Monica Publishing Co.
———. 1989. "Quotron II: An Early Multiprogrammed Multiprocessor for the Communication of Stock Market Data." *Annals of the History of Computing* 11 (2).
Polanyi, Karl. 1944. *The Great Transformation: The Political and Economic Origins of Our Time*. Farrar & Rinehart.
Pollock, Neil, and Robin Williams. 2008. *Software and Organizations: The Biography of the Enterprise-Wide System or How SAP Conquered the World*. Routledge.
Posy, Carl. 2008. "Brouwerian Infinity." In *One Hundred Years of Intuitionism (1907–2007)*, edited by Mark van Atten, Pascal Boldini, Michel Bourdeau, and Gerhard Heinzmann, 21–36. Publications Des Archives Henri Poincaré / Publications of the Henri Poincaré Archives. Birkhäuser Basel. https://doi.org/10.1007/978-3-7643-8653-5_2.
Preda, Alex. 2006. "Socio-Technical Agency in Financial Markets : The Case of the Stock Ticker." *Social Studies of Science* 36 (5):753–82.
Priestley, Mark. 2011. *A Science of Operations Machines, Logic and the Invention of Programming*. Springer.
Pugh, Emerson W., Lyle R. Johnson, and John H. Palmer. 1986a. *IBM's 360 and Early 370 Systems*. The MIT Press.
———. 1986b. *IBM's 360 and Early 370 Systems*. The MIT Press.
Ranadive, Vivek. 1999. *The Power of Now*. Mc-Graw Hill.
Ransom, John. 2014. "Stockmaster - the Service That Revolutionised Reuters." http://www.thebaron.info/archives/technology/stockmaster-the-service-that-revolutionised-reuters.
Rashid, Richard F., and George G. Robertson. 1981. "Accent: A Communication Oriented Network Operating System Kernel." In *Proceedings of the Eighth ACM Symposium on Operating Systems Principles*, 64–75. SOSP '81. New York, NY, USA: ACM. https://doi.org/10.1145/800216.806593.
Rayns, Chris. 2011. *CICS Transaction Server from Start to Finish*. IBM.
Redmond, K.C., and Smith, T.S. 1980. *Project Whirlwind - The History of a Pioneer Computer*. Bedford, Massachusetts: Digital Press.
"Report on the Workshop on Fundamental Issues in Distributed Computing." 1980. www.sigops.org/ew-history/1980/report.pdf.
Rescher, Nicholas. 1996. *Process Metaphysics: An Introduction to Process Philosophy*. SUNY Press.

R.F. Meyer. 1967. “American Airlines Sabre (A). Case No. EA-C 758.” Harvard Business School.
Ritchie, D. M., and K. Thompson. 1974. “The UNIX Time-Sharing System.” In . http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.413.
Ritchie, Dennis M. 1984. “A Stream Input-Output System.” *AT&T Bell Laboratories Technical Journal* 63 (8):1897–1910.
———. 1999. “Advice from Doug Mcilroy.” http://web.archive.org/web/19990508221104/http://cm.bell-labs.com/cm/cs/who/dmr/mdmpipe.html.
Riva, Angelo, and Eugene N. White. 2011. “Danger on the Exchange: How Counterparty Risk Was Managed on the Paris Exchange in the Nineteenth Century.” *Explorations in Economic History* 48 (4):478–93. https://doi.org/10.1016/j.eeh.2011.05.002.
Robbins, Sidney M., Walter Werner, Craig G. Johnson, and Aron Greenwald. 1969. *Paper Crisis in the Securities Industry: Causes and Cures. Is the Stock Certificate Necessary?* Lybrand, Ross Bros. & Montgomery.
Rochet, Jean-Charles, and Jean Tirole. 2003. “Platform Competition in Two-Sided Markets.” *Journal of the European Economic Association* 1 (4):990–1029.
———. 2006. “Two-Sided Markets: A Progress Report.” *The RAND Journal of Economics* 37 (3):645–67.
Rojas, Raúl, and Ulf Hashagen, eds. 2002. *The First Computers--History and Architectures*. Revised edition. Cambridge, Mass: The MIT Press.
Roman, Monica. 1987. “Trading Rooms Stay Abreast of Technology.” *American Banker* 152 (148).
Russell, Andrew L. 2014. *Open Standards and the Digital Age: History, Ideology, and Networks*. New York, NY: Cambridge University Press.
Rustin, Richard. 1975. “Brokers’ Back Offices Are Handling Well The Surge in Trading, Big Board Finds.” *Wall Street Journal*, February.
Rysman, John. 2009. “The Economics of Two-Sided Markets.” *Journal of Economic Perspectives* 23 (3):125–43.
Sacks, Harvey, Emanuel A. Schegloff, and Gail Jefferson. 1974. “A Simplest Systematics for the Organization of Turn-Taking for Conversation.” *Language* 50 (4):696–735. https://doi.org/10.2307/412243.
Sassen, Saskia. 2007. *A Sociology of Globalization*. W.W. Norton and Company.
Saunders, Ray. 2004. “A Dinosaur after Forty+ Years.” http://web.archive.org/raylsaunders.com/asmwork.html.
Schantz, R. E. 2006. “BBN’s Network Computing Software Infrastructure and Distributed Applications (1970-1990).” *IEEE Annals of the History of Computing* 28 (1):72–88. https://doi.org/10.1109/MAHC.2006.4.
Schantz, Richard. 1975. “RFC 684. A Commentary on Procedure Calling as a Network Protocol.” http:// www.rfc-editor.org/rfc/rfc684.txt.
Schatzberg, Eric. 2006. “Technik Comes to America: Changing Meanings of Technology before 1930.” *Technology and Culture* 47 (3).
Schmalensee, Richard, and Robert Willig, eds. 1989. *Handbook of Industrial Organization*. Amsterdam: North-Holland.
Schmerken, Ivy. 1989. “What’s Next on Wall Street’s Automation Agenda?” *Wall Street Computer Review*, April, 45–48, 62–64.

Scholz, Trebor. 2016. “Platform Cooperativism: Challenging the Corporate Sharing Economy.” Rosa Luxemburg Siftung, New York Office.
Schulte, Roy. 1996a. “Batch Is Dead. Long Live Batch.” Gartner Research Note SPA-200-159. Gartner.
———. 1996b. “Online Is Dead. Long Live Online.” Gartner Research Note SPA-200-112. Gartner.
———. 1997. “Putting MOM to Work.” Gartner Research Note TU-401-1296. Gartner.
———. 1998. “A Publish-and-Subscribe Primer.” Gartner Research Note TU-04-3769. Gartner.
Schumpeter, Joseph A. 1942. *Capitalism, Socialism and Democracy*. Harper.
Schutz, Alfred. 1962. *Collected Papers I: The Problem of Social Reality*. The Hague: Martinus Nijhoff.
Schwartz, J.T. 1962. *Relativity in Lllustrations*. New York: New York U. Press,.
Schwartz, Jules I. 1968. “Interactive Systems — Promises, Present and Future.” *Fall Joint Computer Conference*.
Schwartz, Robert A., John Aidan Byrne, and Gretchen Schnee, eds. 2013. *Rethinking Regulatory Structure*. New York: Springer.
Schwartz, Robert A., and Reto Francioni. 2004. *Equity Markets in Action: The Fundamentals of Liquidity, Market Structure & Trading*. Hoboken, NJ: John Wiley and Sons.
SEC. 1963a. *Report of Special Study of Securities Markets of the Securities and Exchange Commission, Part 1*. Washington, D.C.: U.S. Government Printing Office.
———. 1963b. *Report of Special Study of Securities Markets of the Securities and Exchange Commission, Part 2*. Washington, D.C.: U.S. Government Printing Office.
———. 1971a. *Institutional Investor Study Report of the Securities and Exchange Commission*. New York: Arno Press.
———. 1971b. *Securities and Exchange Commission Study of Unsafe and Unsound Practices of Brokers and Dealers, H.R. Doc. No. 92-231, 92d Cong., 2d Sess. 13*.
———. 1988. *The October 1987 Market Break: A Report by the Division of Market Regulation, U.S. Securities and Exchange Commission*.
———. 1989. “Proprietary Trading Systems.” *Federal Register* 54 (73):15420–40.
———. 1992. “U.S. Equity Market Study.” *Federal Register* 57 (141):32587–605.
———. 1995. “Order Execution Obligations: Proposed Rules.” *Federal Register* 60 (195):52792–815.
———. 1996. “Order Execution Obligations: Final Rules.” *Federal Register* 61 (178):48290–332.
———. 1997. “Regulation of Exchanges: Concept Release.” *Federal Register* 62 (107):30485–535.
———. 1998. “Regulation of Exchanges and Alternative Trading Systems: Final Rules.” *Federal Register* 63 (245):70844–951.
Securities and Exchange Commission. 1963. “Ch. 12.” In *Report of Special Study of Securities Markets of the Securities and Exchange Commission*.
———. 1994. *Market 2000 - An Examination of Current Equity Market Developments*. Washington, D.C.: Division of Market Regulation.
Seligman, Joel. 1982. *The Transformation of Wall Street: A History of the Securities and Exchange Commission and Moderm Corporate Finance*. Houghton Mifflin Company.
———. 1984. “The Future of the National Market System.” *Journal of Corporation Law* 10 (1):79–139.

———. 1985. *The SEC and the Future of Finance*. New York: Praeger Publishers.
———. 1995. “Another Unspecial Study: The SEC’s Market 2000 Report and Competitive Developments in the United States Capital Markets.” *The Businesss Lawyer* 50 (2):485–526.
———. 2004. “Cautious Evolution or PerenniaI Irresolution: Stock Market Self-Regulation During the First Seventy Years of the SEC.” *The Business Lawyer* 59 (August):1347–87.
Serlin, Omri. 1993. “The History of DebitCredit and the TPC.” In *The Benchmark Handbook for Database and Transaction Processing Systems*, edited by Jim Gray, 2nd ed.
Serres, Michel. 1995. *Angels: A Modern Myth*. Translated by Francis Cowper. Flamarrion.
Shannon, Claude E. 1948. “A Mathematical Theory of Communication.” *The Bell System Technical Journal* 27 (3):379–423.
Shoch, J.F., Y.K. Dalal, and D.D. Redell. 1982. “The Evolution of the Ethernet Local Area Network.” *IEEE Computer* 15 (8):10–28.
Silberschatz, Abraham, Henry Korth, and S. Sudarshan. 2010. *Database System Concepts*. 6th ed. McGraw-Hill Science/Engineering/Math.
Silverstein, Michael. 1976. “Shifters, Linguistic Categories and Cultural Description.” In *Meaning in Anthropology*, edited by Keith H. Basso and Henry A. Selby, 11–55. Albuquerque: University of New Mexico Press.
Simmel, Georg. 2004. *The Philosophy of Money*. Routledge.
Simondon, Gilbert. 1958. *Du Mode d’existence Des Objets Techniques*. Aubier / Montaigne Analyse et Raisons.
———. 1980. *On the Mode of Existence of Technical Objects*.
Siwiec, J. E. 1977. “A High Performance DB/DC System.” *IBM Systems Journal* 16 (2).
Skeen, Dale. 1992. “An Information Bus Architecture for Large-Scale, Decision-Support Environments.” *Proceedings of the Winter USENIX Conference*, 183–95.
Sklar, Lawrence. 1977. *Space, Time, and Spacetime*. New Ed edition. Berkeley, Calif.: University of California Press.
Slater, Don. 2002. “Markets, Materiality and the New Economy.” In *Market Relations and the Competitive Process*, edited by Stan Metcalfe and Alan Warde. Manchester University Press.
Smith, Adam. 1776. *An Inquiry into the Nature and Causes of the Wealth of Nations*. Dublin: Whitestone.
Smith, Jeffrey W., James P. Selway III, and D. Timothy McCormick. 1998. “The Nasdaq Stock Market: Historical Background and Current Operation.” NASD Working Paper 98-01. Washington, D.C.: NASD Economic Research Department.
Smith, Richard B. 1970. “A Piece of Paper.” *The Business Lawyer*, no. 3 (April):923–30.
Soare, Robert I. 1996. *The History and Concept of Computability*.
Solomon, Eliot. 1998. “Middleware at SIAC and in the Securities Industry.” *Financial MiddlewareSpectra* 12 (3):2–11.
Sombart, Werner. 2005. “Technology and Culture.” In *Sociological Beginnings: The First Conference of the German Society for Sociology*. Liverpool University Press.
Spragins, John D., Joseph Hammond, and Krzysztof Pawlikowski. 1991. *Telecommunications: Protocols and Design*. First Edition edition. Reading, Mass: Prentice Hall.
Standage, Tom. 1998. *The Victorian Internet: The Remarkable Story of the Telegraph and the Nineteenth Century’s On-Line Pioneers*. 1 edition. New York: Walker & Company.

Steel, Thomas B., Jr. 1968. “Multiprogramming: Promise, Performance and Prospect.” In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, 99–103. AFIPS ’68 (Fall, Part I). New York, NY, USA: ACM. https://doi.org/10.1145/1476589.1476607.

Steil, Benn. 2002. “Changes in the Ownership and Governance of Securities Exchanges: Causes and Consequences.” In *Brookings-Wharton Papers on Financial Services: 2002*, 61–91. Washington, D.C.: Brookings Institution Press.

Stiegler, Bernard. 1998. *Technics and Time 1: The Fault of Epimetheus*. Stanford University Press.

———. 2011. *Decadence of Industrial Democracies*. 1 edition. Polity.

———. 2012. “Relational Ecology and the Digital Pharmakon.” *Culture Machine* 13:1–19.

Stiegler, Bernard, and Ars Industrialis. 2008. *R’eenchanter Le Monde : La Valeur Esprit Contre Le Populisme Industriel*. Editions Flammarion.

Stocker, James. 2011. *Securities and Exchange Commission Historical Society Interview with Dennis Dirks*.

Sturgis, H., J. Mitchell, and J. Israel. 1980. “Issues in the Design and Use of a Distributed File System.” *SIGOPS Oper. Syst. Rev.* 14 (3):55–69. https://doi.org/10.1145/850697.850705.

Sussenguth, E. H. 1978. “Systems Network Architecture: A Perspective.” In *Evolutions in Computer Communications: Proceedings of the Fourth International Conference on Computer Communication, Kyoto, 26-29 September 1978*, edited by H. Inose, 353–58. Amsterdam: North-Holland.

Swedberg, Richard. 2000. “Afterword: The Role of the Market in Max Weber’s Work.” *Theory and Society* 29:379–84.

———. 2003. *Principles of Economic Sociology*. Princeton University Press.

———. 2007. “The Tool Kit of Economic Sociology.” In *The Oxford Handbook of Political Economy*, edited by Donald A. Wittman and Barry R. Weingast, 937–50.

———. 2011. “The Role of the Senses and Signs in the Economy: More on the Centrality of Materiality.” *Journal of Cultural Economy* 4 (4):423–37.

Tam, Donna. 2014. “Ride-Sharing Service Sidecar Lets Drivers Name Their Own Prices.” CNET. February 19, 2014. http://www.cnet.com/news/ride-sharing-service-sidecar-lets-drivers-name-their-own-prices/.

Tarr, Joel A., Thomas Finholt, and David Goodman. 1987. “The City and the Telegraph: Urban Telecommunications in the Pre-Telephone Era.” *Journal of Urban History* 14 (1):38–80. https://doi.org/10.1177/009614428701400103.

Tay, B. H., and A. L. Ananda. 1990. “A Survey of Remote Procedure Calls.” *SIGOPS Oper. Syst. Rev.* 24 (3):68–79. https://doi.org/10.1145/382244.382832.

Taylor, E.F., and J.A. Wheeler. 1966. *Space-Time Physics*. San Francisco: W.H. Freeman.

Teitelman, Warren. 1972. “”DO WHAT I MEAN”: The Programmer’s Assistant.” *Computers and Automation* 21 (4):8–11.

“Teknekron Debuts New TIB; Reuters Opts To Use Old One.” 1995. *Sell-Side Technology*, September. http://www.waterstechnology.com/sell-side-technology/feature/1629102/teknekron-debuts-new-tib-reuters-opts-to-use-old-one.

Thornton, Bill. 2000. “Tibco Fidelity Investments: The Internet Trading Room.” Strategic Directions 2000. http://web.archive.org/web/20010409045219/http://media.tibco.com/sd2000/video/index.html.

Tieszen, Richard. 2008. "The Intersection of Intuitionism (Brouwer) and Phenomenology (Husserl)." In *One Hundred Years of Intuitionism (1907–2007)*, edited by Mark van Atten, Pascal Boldini, Michel Bourdeau, and Gerhard Heinzmann, 78–95. Publications Des Archives Henri Poincaré / Publications of the Henri Poincaré Archives. Birkhäuser Basel. https://doi.org/10.1007/978-3-7643-8653-5_6.
"TST 10th Anniversary, Leaders Of The Pack." 1997. *Sell-Side Technology*, October.
Turing, A. M. 1950. "Computing Machinery and Intelligence." *Mind* 59 (236):433–460.
Turing, Alan M. 1936. "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proceedings of the London Mathematical Society* 2 (42):230–65.
U.S. 73rd Congress. 1934. *Securities Exchange Act*.
U.S. 94th Congress. 1975. *Securities Acts Amendments of 1975*.
U.S Congress Office of Technology Assessment. 1990. *Electronic Bulls and Bears: U.S. Securities Markets and Information Technology*. OTA-CIT-469.
U.S. House, 92nd Congress. 1972. *Study of the Securities Industry: Hearings Before the Subcommittee on Commerce and Finance*.
U.S. House, 103rd Congress. 1993. *National Market System: Hearings Before the Subcommittee on Commerce and Finance*.
U.S. Senate, 93rd Congress. 1973. *Securities Industry Study*.
Uzzi, Brian. 1997. "Social Structure and Competition in Interfirm Networks - The Paradox of Embeddedness." *Administrative Science Quarterly* 42 (1):35–67.
Verstein, Andrew. 2011. "The Misregulation of Person-to-Person Lending." *UC Davis Law Review* 45:445–530.
Vinoski, Steve. 2008. "[Erlang-Questions] Rpc Is Bad? (Was Re: Facebook Chat Server)." May 22, 2008. http://erlang.org/pipermail/erlang-questions/2008-May/035209.html.
Wade, Bradford W., and Donald D. Chamberlin. 2012. "IBM Relational Database Systems: The Early Years." *IEEE Annals of the History of Computing* 34 (4):18–29.
Walker, Donald A. 2001. "A Factual Account of the Functioning of the Nineteenth-Century Paris Bourse." *European Journal of the History of Economic Thought* 8 (2):186–207.
Walras, Léon. 1954. *Elements of Pure Economics : Or, The Theory of Social Wealth*.
Weber, Max. 1922. *Economy and Society*. University of California Press.
———. 1968. *Economy and Society*. University of California.
———. 2005. "Remarks on Technology and Culture." *Theory, Culture & Society* 22 (4):23–38.
Welles, Chris. 1975. *The Last Days of the Club*. Dutton.
Wells, Wyatt. 2000. "Certificates and Computers: The Remaking of Wall Street, 1967 to 1971." *Business History Review* 74 (2):193–235.
Westrum, Ron. 1991. *Technologies & Society: The Shaping of People and Things*. Wadsworth Publishing Company.
Westwood, Howard C., and Edward G. Howard. 1952. "Self-Government in the Securities Business." *Law and Contemporary Problems* 17:518–44.
White, Harrison. 1981a. "Production Markets as Induced Role Structures." *Sociological Methodology* 12:1–57.
———. 1981b. "Where Do Markets Come From?" *American Journal of Sociology* 87 (3):517–47.
———. 1988. "Varieties of Markets." In *Social Structures: A Network Approach*, edited by B. Wellman and S. D. Berkowitz. New York: Cambridge University Press.

———. 1992. "Markets, Networks and Control." In *Interdisciplinary Perspectives on Organization*, edited by S. Lindenberg and Hein Schroeder, 223–39. Oxford: Pergamon Press.
———. 2002. *Markets From Networks: Socioeconomic Models of Production*. Princeton University Press.
White, Harrison, and Robert Eccles. 1987. "Producers' Markets." In *The New Palgrave: A Dictionary of Economics*, edited by John Eatwell, Murray Milgate, and Peter Newman, 984–86. London: Macmillan.
White, Harrison, and Eric M. Leifer. 1988. "A Structural Approach to Markets." In *Structural Analysis of Business*, edited by Mark Mizruchi and Michael Schwartz. New York: Cambridge University Press.
White, James E., and Jon Postel. 1974. "RFC 674: Procedure Call Protocol Documents." https://tools.ietf.org/pdf/rfc674.pdf.
William Bernhard Tegetmeier. 1871. *The Homing or Carrier Pigeon*. http://archive.org/details/homingorcarrier00tegegoog.
Williamson, Oliver E. 1981. "The Economics of Organization: The Transaction Cost Approach." *American Journal of Sociology* 81 (3):548–77.
Wingfield, Nick. 1999. "Tibco Software 'Pushes' Wares Beyond Its Wall Street Roots." *Wall Street Journal*, March.
Winslet, Marianne. 2003. "Jim Gray Speaks Out on Chasing the Object-Relational Rainbow, Why Performance Is a Nonissue, Bad Ideas That Went Good, Reinventing the Field, Sailboats, Lunatic Fringe Papers, Whether to Try for a Home Run, and More." *SIGMOD Record* 32 (1).
Woodward, Curt. 2016. "New App Gives Uber a Little Disruption of Its Own." *Boston Globe*, May.
Wooton, Charles W., and Carel M. Wolk. 2000. "The Evolution and Acceptance of the Loose-Leaf Accounting System, 1885-1935." *Technology and Culture* 41 (1):80–98.
Yates, JoAnne. 1989. *Control through Communication: The Rise of System in American Management*. Johns Hopkins University Press.
Yates, Joanne. 2006. "How Business Enterprises Use Technology: Extending the Demand-Side Turn." *Enterprise & Society* 7 (3):422–455.
Yates, JoAnne. 2008. *Structuring the Information Age: Life Insurance and Technology in the Twentieth Century*. Johns Hopkins University Press.
Yelavich, B. M. 1985. "Customer Information Control System-An Evolving System Facility." *IBM Systems Journal* 24 (3/4).
Yelavich, Bob. 2003. "Evolution of CICS."
Zalamea, Fernando. 2016. "Multilayered Sites and Dynamic Logics for Transits between Art and Mathematics." *Glass Bead* Site 0: Castalia, the game of ends and means (February).
Zaloom, Caitlin. 2004. "The Productive Life of Risk." *Cultural Anthropology* 19 (3):365–91.
Zelizer, Viviana A. 1988. "Beyond the Polemics on the Market: Establishing a Theoretical and Empirical Agenda." *Sociological Forum* 3 (4):614–34.
Zwick, Detlev, and Nikhilesh Dholakia. 2006. "The Epistemic Consumption Object and Postsocial Consumption: Expanding Consumer-Object Theory in Consumer Research" 9 (1):17–43.