

The Sociality of Computation: From Educational Systems to Persistent Digital Worlds

Flyxion
Independent Researcher

October 2026

Abstract

The conventional identification of social media with feeds, followers, profiles, and algorithmic recommendation obscures a longer history of social computing. Educational systems such as PLATO developed communication and community functions out of shared instructional infrastructure, and contemporary platforms including Wikipedia, GitHub, YouTube, and multiplayer games organize social interaction around different kinds of persistent computational objects. This essay proposes an object-centered and event-centered account of social media, distinguishing publication, annotation, collaboration, and co-presence as overlapping mechanisms of social organization. It examines the community courses, mnemonic annotations, and following system of Memrise as a case study in how platform governance can enable or foreclose emergent social functions. It then considers generative flashcards, source-grounded discussion, spatial interfaces, and auditory environments as possible extensions of social computing beyond the feed, and asks what personalized generation does to the common objects on which sociality depends. The central claim is that sociality emerges not from a particular interface but from the capacity of a computational environment to make contributions, relationships, and shared histories mutually accessible. The essay offers a minimal formal model of this claim, offered as a proposal for analysis and not as an established theorem, and draws institutional consequences for the design of object-centered social systems.

Contents

I	The Category Error	3
1	Introduction: What Counts as Social Media	4
2	From Communication Channels to Shared Computational Objects	5
3	A Minimal Structural Account of Social Media	6

4	Social Consequence Without Persistence	8
5	From Causal Coupling to Social Practice	9
II	Educational Origins	10
6	PLATO as a Historical Counterexample	11
7	From Contributed Lessons to PLATO Notes	12
8	Educational Infrastructure and Political Expression	13
9	Architectural Possibility and the Labor of Social Invention	14
III	An Unrealized Architecture of Social Knowledge	15
10	Community-Created Courses and Disciplinary Openness	16
11	MEMS: When Mnemonics Become Messages	17
12	Following, Ranking, and the Construction of Comparison Groups	18
13	Platform Enclosure and the Removal of Community Courses	19
IV	The Comparative Architecture of Social Platforms	21
14	Wikipedia: Collaborative Authority and Revision History	21
15	GitHub: Software Development as Social Media	22
16	YouTube: Publication, Annotation, and Audience Formation	22
17	Multiplayer Games: Sociality Through Action	23
18	A Comparative Taxonomy	24
V	Platform Governance and the Foreclosure of Possible Communities	24
19	Affordances, Appropriation, and Unintended Uses	24
20	The Politics of Interface Removal	25

21 Authority, Permissions, and the Ownership of Shared Worlds	26
22 Twitter, X, and the Loss of a Conceptual Metaphor	27
VI Generative Learning and the Future of Shared Knowledge	28
23 NotebookLM and the Automated Production of Learning Objects	28
24 From Generated Cards to Collective Annotation	29
25 Learning Networks as Distributed Epistemic Institutions	29
VII Beyond the Screen	30
26 The Historical Contingency of Visual Interfaces	30
27 The Auditory Memory Palace	31
28 SpherePOP and the Inhabitable Computational Environment	32
29 The Problem of Representation and Delegated Interpretation	32
30 The Three Layers of Shared Reference	33
31 Personalized Generation and the Fragility of Shared Worlds	34
VIII Formal Synthesis and Institutional Consequences	36
32 A Theory of Social Emergence in Computational Environments	36
33 History Dependence and the Augmented Social State	38
34 Event Sourcing, Social Memory, and the Identity of Shared Objects	40
35 Sociality Without Profiles, Feeds, or Permanent Identities	42
36 Institutional Design for Object-Centered Social Systems	42
37 Conclusion: Social Media as a Property of Shared Worlds	43

Part I

The Category Error

1 Introduction: What Counts as Social Media

The phrase social media is now used almost exclusively for a family of services organized around a profile, a graph of followers or friends, and a stream of posts ordered by a ranking algorithm. Facebook, Twitter (now X), Instagram, and TikTok are the paradigmatic cases, and much of the scholarly and regulatory literature has been written as though these services exhausted the category, a history that van Dijck (2013) traces in detail. The most influential academic definition of social network sites, that of Boyd and Ellison (2007), describes them as web-based services that let individuals construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection, and view and traverse their lists of connections and those made by others. The definition is careful and has been enormously useful, but it is also a definition of one architecture. It takes the profile and the connection list as constitutive, and so it classifies as social only those systems in which persons present themselves to one another as persons. Kaplan and Haenlein (2010) offer a broader characterization in terms of Web 2.0 ideology and the exchange of user-generated content, which admits wikis, content communities, and virtual worlds, but their taxonomy treats these as sibling categories sharing a family resemblance, and does not say what common mechanism makes them social.

The difficulty becomes visible as soon as one considers systems that are plainly social in their effects and plainly unlike the feed. Wikipedia sustains cooperation among large numbers of people who frequently do not know one another, rarely follow one another, and communicate chiefly through the object they are jointly producing. GitHub organizes reputation, mentorship, dispute, and institutional formation around repositories, issues, and patches. YouTube combines publication with a comment culture and with interpretive communities that gather around particular videos long after their upload. Multiplayer games such as Minecraft and EVE Online generate economies, alliances, wars, and customs without requiring that a single message be sent. And educational platforms, which are rarely counted as social media at all, have repeatedly produced communities as a by-product of shared instruction. These counterexamples are not marginal. Taken together they account for a large fraction of the sustained collaborative activity that occurs online, and a definition of social media that excludes them has mistaken a historical branch for the trunk.

This essay therefore asks a more general question: under what conditions does a computational system become a medium of social organization? The question has to be answered with some care in both directions. A definition that is too narrow, as the feed-centered definitions

are, misclassifies a great deal of social computing. A definition that is too broad, one that counts every interactive website as social media, ceases to say anything. A calculator that two people happen to use in the same room is not a social medium, and neither is a web form that stores a submission in a database nobody else can read. The task is to find criteria that exclude trivial interaction while accommodating systems that lack conventional profiles or feeds.

The approach taken here combines four methods. Historical reconstruction supplies the sequence in which communication functions emerged from instructional systems, beginning with PLATO and continuing through Memrise. Comparative platform analysis tests the proposed account against systems with very different architectures. A conceptual taxonomy distinguishes the mechanisms by which shared objects support social relations. And speculative interface design, treated as an exercise in extending the account instead of as a forecast, considers what the account implies for generative learning systems and for auditory and spatial environments. Throughout, the essay distinguishes documented history from recollection, and flags the claims that depend on sources that remain to be verified.

2 From Communication Channels to Shared Computational Objects

The dominant model of online social life descends from the model of the telephone and the postal service. A sender addresses a message to a recipient, or to a set of recipients, and the medium carries it. Even when the recipient set is open, as in a public post, the unit of analysis is a communicative act directed at an audience. Feed-based platforms refine this model by adding a persistent profile on the sender's side and a ranking function on the receiver's side, but the underlying logic remains that of messages in transit between persons.

A second and older logic is available. Communication can be mediated by persistent objects to which contributions are addressed, instead of by messages that travel between persons. A contribution to an article is not addressed to the next reader of the article, who may not yet exist and whose identity is irrelevant to the contributor. It is addressed to the article, and its significance lies in what it does to the article. A commit is addressed to a codebase, a card annotation to the card, a modification of terrain to the world in which others will walk. In each case later participants encounter the contribution as part of the object and not as a message. The relation between contributor and later participant is mediated entirely by the altered state of the shared thing.

This distinction has a long pedigree in the study of cooperative work. Star and Griesemer (1989) introduced the notion of boundary objects to describe artifacts that are plastic enough to adapt to the local needs of several communities of practice and robust enough to maintain a common identity across them. Their study of a natural history museum showed how specimens, field notes, and maps allowed amateurs, professional biologists, administrators, and trappers to

cooperate without consensus about what the collection was for. The point carries over directly to the present argument: the shared object makes coordination possible among parties whose purposes differ, and it does so by being interpretable in more than one way. Hutchins (1995), in his study of navigation aboard a naval vessel, argued that cognition is distributed across people and artifacts, so that charts, instruments, and procedures are parts of the cognitive system and not mere aids to it. Suchman (1987) argued, against plan-based models of action, that purposeful activity is organized in situ by participants using the resources the situation provides, and that shared artifacts are among those resources. The field of computer-supported cooperative work, from its early statements through its later critical turns, has repeatedly documented how groupware succeeds when it provides common objects that participants can see, change, and refer to, and fails when it provides channels without such objects. A related biological idea, stigmergy, introduced by Grassé (1959) to explain coordination among termites, describes coordination through modifications of a shared environment that guide the subsequent actions of others. Heylighen (2016) has argued that stigmergy provides a general account of coordination in open online collaboration, and the account fits Wikipedia and open-source development closely.

The shared object is therefore a candidate locus of social interaction. Its significance, however, cannot depend on persistence alone. A file that persists on a private disk is persistent and not social. What matters is whether contributions remain available for subsequent interpretation and action by others, that is, whether the object is accessible, whether changes to it are observable, and whether those changes bear on what others can later do. Persistence is a means to this end, and in some systems, such as a live multiplayer match, the relevant shared object may be a transient event whose consequences are carried by the participants and not by a stored record. The account developed below accordingly treats durable artifacts and transient shared events together.

3 A Minimal Structural Account of Social Media

The informal argument can be given a precise form, which is useful chiefly because it forces decisions about what is and is not required. Let a computational environment be described by a tuple

$$\mathcal{M} = (U, O, A, E, H, P, G),$$

where U is a set of participants, O a set of shared objects, A a set of admissible actions, E a set of observable events, H a family of retained histories, P a set of publication and visibility rules, and G a set of governance mechanisms. Participants perform actions on objects, actions produce events, events are recorded into histories according to retention rules, visibility rules determine which participants may observe which events and objects, and governance mechanisms determine who may modify the rules themselves.

Definition 1 (Socially generative environment). *An environment $\mathcal{M}$ is socially generative when there exist participants $u, v \in U$, an action $a \in A$ performed by u , and an object or event x such that a alters x , x is accessible to v under P , and the alteration changes what v can perceive, interpret, or do. The alteration need not persist, and the role of persistence is taken up in the next section.*

The definition is deliberately weaker than any definition in terms of profiles or followers. It requires that one participant's action become meaningful and consequential for another through an accessible shared environment, and it says nothing about whether the participants know each other, whether they are identified, or whether the action is a communicative act. It excludes the isolated calculator, because there is no second participant for whom anything changes, and it excludes the private database, because P withholds the alteration from every participant but one. It includes a wiki, a repository, a flashcard with a shared annotation field, and a game world in which one player's construction alters another's route.

Borderline cases show what the definition does and does not do. Consider two people who edit different parts of a shared document and never see one another's changes, because the visibility rules P hide each editor's work from the other. The definition excludes this case, since no alteration is accessible to the second participant. A public spreadsheet that no one else has opened or altered is a different case, since it is accessible: it is potentially social, and becomes a realized interaction only when someone encounters or alters it, a distinction developed in the next section. A multiplayer game with a single player in a world that others could enter is likewise potentially social until someone enters, and a game with two players who never come within sight of one another is excluded for the same reason as the hidden edits. The definition is therefore sensitive to access and not merely to the existence of shared storage. It is also permissive at the margin: two people who edit a shared document and can see each other's changes satisfy it, even if they never exchange a word, and many readers will judge that case only weakly social. This is intended. The definition names a minimal capacity, the conditions under which social organization can arise, and the question of when that capacity is realized as sustained organization is the subject of Section 32, where a further condition is introduced.

A distinction between necessary conditions and typical features keeps the account from collapsing into a checklist. Persistence, stable identity, and explicit communication all tend to strengthen social organization, and many familiar systems have all three. None of them is necessary in its strongest form. Persistence can be weak, as in a live session whose consequences live in participants' memories and altered positions. Identity can be thin, as in the pseudonymous or anonymous contributions that make up a large part of Wikipedia's edit history. Explicit communication can be absent, as in a multiplayer world where cooperation proceeds by building. What the definition requires is the more abstract condition that actions by some participants bear on the possibilities of others through a shared medium.

The sections that follow use this model in two ways. First they use it descriptively, to recover

from the historical and comparative material the mechanisms by which different systems satisfy the definition. The next section adds a state transition structure so that cross-participant consequence can be stated precisely, and Section 32 uses it to draw a further distinction between ordinary shared computation and sustained social organization.

4 Social Consequence Without Persistence

The structural account developed in this essay distinguishes the capacity for social interaction from the persistence of social organization. The distinction is necessary because an action can be consequential for another participant without leaving a durable record in the computational environment. A gesture in a multiplayer game, a brief sound in a shared auditory space, or the movement of one participant around another can alter what the second participant perceives and does, even when the environment returns immediately to its previous configuration. Such interactions are social in their causal structure, though they need not establish a continuing relationship.

Persistence is therefore not a necessary condition for social generativity in its minimal sense. It is a condition that supports the accumulation of social structure, and it can be realized in several ways. A computational system may retain records of actions. Its current state may embody the consequences of previous actions. Participants may remember interactions that the system itself has forgotten. Or institutional rules may preserve obligations and expectations that originated in earlier encounters. These mechanisms differ in where history is retained, but each allows the past to bear on future conduct.

To distinguish the minimal condition from its stronger historical development, let an environment have a state space X , a set of participants U , and admissible actions A . Let

$$T : X \times U \times A \longrightarrow X$$

describe the transition resulting from an action. For each participant v , let $\text{Perc}_v(x)$ denote the information available to that participant in state x , and let $\text{Poss}_v(x)$ denote the actions available to that participant.

Definition 2 (Cross-participant consequence). *An action a performed by participant u in state x is consequential for a distinct participant v if it changes the information available to v , the actions available to v , or the subsequent behavior of v through a causal pathway mediated by the shared environment.*

This definition does not require that the effect persist. It requires that the action enter the causal situation of another participant. The effect may be fleeting, and it may leave no record accessible to the system after the encounter. The definition also does not require explicit communication. A player who blocks a doorway changes another player's available route, and a

player who moves aside changes it again. Neither action needs to be accompanied by a message for the interaction to be socially consequential.

The distinction between accessibility and actual observation is equally important. A contribution may be available to other participants without having been encountered by any of them. A publicly readable document that has not yet been opened possesses the structural capacity to mediate interaction, but no particular encounter has thereby occurred. It is useful to distinguish a *potentially social* environment, in which cross-participant access is possible, from a *realized social interaction*, in which one participant's action actually enters another's causal situation.

This distinction prevents the account from treating every publicly accessible database as an already constituted community. The existence of an accessible object establishes an opportunity for social interaction; the realization of that opportunity requires participants to encounter and respond to one another, directly or indirectly.

A multiplayer game illustrates the distinction. A world may be designed for thousands of players and yet be occupied by only one. Its architecture permits social interaction, but no interaction among participants is occurring. When another player enters, the potential may be realized. If the players never encounter one another and their actions remain causally independent, the mere fact that they occupy the same server does not establish a social relationship. If one changes terrain through which the other later travels, however, the shared environment mediates an interaction even if the players never meet.

The relevant criterion is thus not simultaneous presence or durable storage alone, but cross-participant consequence. Persistence becomes important at the next level, where repeated consequences accumulate into relationships, reputations, norms, and institutions.

5 From Causal Coupling to Social Practice

Cross-participant consequence, as defined above, is a deliberately minimal condition, and it is broader than ordinary usage of the word social. A participant who consumes a scarce computing resource raises the latency experienced by others, and the action is consequential for them in the sense of the definition, though neither party need recognize the other or act within any shared practice. Counting every multi-user resource allocator as a social medium would trivialize the account. The remedy is to distinguish three levels and to reserve the term social for the second and third. Causal interdependence holds when one participant's action affects another through a shared environment. Social interaction holds when the consequence is situated within a shared and interpretable practice, so that the affected participant can take the action as something done by someone, for some purpose, and respond to it as such. Social organization holds when interactions of this kind establish continuing expectations, roles, relationships, or institutions. The minimal definition of the previous sections then names a capacity, which may

be called social generativity, and not sociality itself.

The step from the first level to the second turns on uptake. Three situations are worth separating. In accidental interference, an action affects another participant without either treating it as communicative, as in the latency example or two players who collide without noticing one another. In intentional communication, the actor designs the action to be understood by another, as in a message or a gesture. In coordination through artifacts, neither of these holds in a simple form: the actor modifies a shared object, the modification is addressed to the object and not to a person, and later participants take it up as part of the practice that surrounds the object. The edit, the commit, and the annotation are of this last kind, and they are the central cases of the account developed here. What makes them social is that the shared object carries conventions of use, so that an action on it is interpretable by anyone familiar with the practice, and the same physical action on an object with no such conventions would be mere interference.

The distinction between access and observation can be made consistent across documents, games, and auditory environments by distinguishing three grades. Structural availability holds when a participant is permitted and technically able to retrieve something. Situated perceptibility holds when the participant can encounter it under the present conditions of the interface, which depend on position, attention, and modality. Actual uptake holds when the participant encounters it and its content influences interpretation or conduct. An unread public document is structurally available and, in a typical case, perceptible, though it has not been taken up. An event in an auditory world that no participant is positioned to hear is not even situated perceptible. Potential sociality requires structural availability, realized interaction requires uptake, and the intermediate grade explains why interface design can remove sociality without changing any permission.

The terminology of the essay can now be fixed. Social computing is the broad domain of computation in which people's actions bear on one another. Social generativity is the structural capacity defined above. Realized social interaction is an actual consequence taken up within a shared practice. Sustained social organization is a historically stabilized practice, and the following sections give its conditions. Social media, on this usage, names a family of socially generative computational media, of which the feed-based network is one member, and it is not a synonym for any of the four.

Part II

Educational Origins

6 PLATO as a Historical Counterexample

The PLATO system, developed at the University of Illinois at Urbana-Champaign beginning in 1960, when Donald Bitzer built the first system, and later organized under the Computer-based Education Research Laboratory established in 1967, is the historical anchor of this essay for a simple reason. It was built to deliver instruction, its designers thought of it as an instrument for teaching, and it nonetheless became the site of discussion forums, electronic mail, real-time chat, multiplayer games, and communities of users who regarded the system as a place. The principal narrative history is Brian Dear's *The Friendly Orange Glow* (2017), which reconstructs the project and its culture from extensive interviews and documents. The account here follows Dear on the main outline and supplements it where contemporary documentation permits. Smith and Sherwood (1976) give a contemporary overview of the system's educational uses, and Van Meer (2003) treats its funding and commercialization.

PLATO was a mainframe-based system. Students and authors worked at terminals connected to a central computer, and in its fourth major version, introduced in the early 1970s, the terminals incorporated a plasma display panel that combined graphics and text and allowed touch input. The system was designed around the delivery of lessons written in a purpose-built authoring language, TUTOR, which made it relatively easy for instructors, and later for students and amateurs, to write interactive material. TUTOR was designed principally by Paul Tenczar in the mid to late 1960s as a special-purpose authoring language, and encyclopedic accounts of the language, which cite the CERL documentation (Avner and Tenczar 1969), describe its evolution into a general-purpose programming language as unplanned. The language also supplied a substrate for communication. Each user process had a set of student variables that persisted from session to session, and lessons could attach common blocks of shared memory, so that the users of a lesson could exchange information through memory they held in common. Several features of this arrangement bear on the argument. The computing resources were shared, so that actions taken by one user at the central machine could in principle be visible to others. The authoring language was accessible to people who were not professional programmers, so that the population of contributors was far larger than a conventional software development staff. And the terminals were distributed across campuses and, through the commercial expansion of the system by Control Data Corporation, across other institutions, so that the user population extended beyond the original site.

The significance of this arrangement is that instructional infrastructure supplied, without any intention to do so, almost all of the elements of the structural account given above. There

were participants with stable identities, because users had accounts. There were shared objects, namely lessons, programs, and later discussion spaces. There were actions that altered those objects and events that others could observe. There were retained histories in the form of stored lessons and, as will be seen, stored discussions. And there were governance mechanisms, partly formal, in the form of administrative control over accounts and resources, and partly informal, in the form of norms that developed among heavy users. The communication functions that followed were not guaranteed by this arrangement, since particular people still had to conceive and build Notes, Talkomatic, and the games. The more defensible claim is that the educational infrastructure enabled and encouraged social functions beyond its original institutional purpose, and Section 9 distinguishes the technical possibility from the practice that participants built on it.

The historical claim needs to be stated with precision, and the essay adopts the discipline that Dear's narrative itself encourages. PLATO was not simply an early version of Facebook, and describing it as one erases differences that matter. Its institutional purpose was instruction and research and not the maintenance of a public square. Its computing resources were scarce and centrally controlled. Access was restricted to affiliated users and later to paying customers of a commercial service. Its user population was not a cross-section of any national public. What the PLATO case shows is something narrower and more useful: that the functions now associated with social media can be generated by a shared educational environment, and that they appeared there before the personal computer, before the public web, and before any system designed to be social.

7 From Contributed Lessons to PLATO Notes

The development of communication on PLATO followed a sequence that is instructive for what it reveals about the relation between production and conversation. The first stage was user-authored instructional material. Because TUTOR made lesson writing accessible, and because the system was available to many people for long hours, users wrote not only formal courses but also games, demonstrations, and utilities. Multiplayer games, among them dungeon-crawling games, space combat games such as *Empire*, and the flight and combat game *Airfight*, were written by students and enthusiasts and became some of the most heavily used programs on the system. Encyclopedic accounts report thousands of such games from around 1970 through the 1980s, and report that *Avatar* and *Empire* each accumulated more than a million contact hours, figures that should be traced to their sources. These games are important for the present argument because they were social in the sense of the structural account: they placed several participants in a shared simulated environment in which the actions of each altered the situation of the others.

The second stage was messaging. Term-Talk, a one-to-one chat facility, and the real-time

chat program Talkomatic, created by Doug Brown in 1973 with David Woolley credited as a collaborator in some accounts, extended the system's communication functions, and a short subroutine reported to have required about twenty lines of code is credited with enabling communication between terminals. The third stage, and the one most important here, was PLATO Notes, written by David R. Woolley in 1973 as a means of collecting bug reports about the system. Notes allowed users to post comments that were stored and could be read and answered by others, and it developed, with later versions such as Group Notes, into a general-purpose discussion system organized around topics and threads. In Dear's account, as in the broader historical literature on early online communities, the development of Notes is a prime example of discussion emerging within a technical system. The facility was introduced for a narrow instrumental purpose, the reporting of faults, and it was taken up for conversation about anything that people on the system cared to discuss. Woolley's own retrospective (1994) is the principal account of this development and is the usual source for the claim that Notes was an ancestor of the commercial groupware product Lotus Notes, a claim that should be checked there.

The pattern is worth stating in general form. A system provides a persistent shared object, in this case a log of reports to which anyone may add. Participants discover that the object can bear contributions of more than one kind. Repeated encounters with each other's contributions produce recognition, reputation, and relationships. The designers then face a choice between treating this as misuse and treating it as a new function. The Notes case is notable because, at least in the account usually given, the facility was retained and developed, and the reporting function became one use among many. Section 20 considers what happens when the choice goes the other way.

8 Educational Infrastructure and Political Expression

Educational computing resources are institutional property, and institutions have purposes. When participants develop uses that exceed those purposes, a tension arises between the authorized function of the system and the functions its users have given it. In the PLATO case this tension is reported to have taken political forms. The system existed on a university campus during a period of intense political activity, and it is plausible, given what is known about the conduct of campus computing in that era, that discussion on the system touched political questions and that administrators had views about this. A recollection that has circulated in connection with the history of the system holds that users pressured administrators concerning protests against the Vietnam War, and that the system became a vehicle for such organization.

The secondary literature documents related episodes. A political science student, Valarie Lamont, wrote a program called Boneyard Creek that presented the history of a local stream and invited comment on pollution, and Stuart Umpleby contributed to DELPHI, a project

through which users could share views on public issues, rate their importance, and state their own estimates of how likely various outcomes were. Umpleby's later attempt to host a discussion group on the Watergate scandal is reported to have been stopped under a threat to the project's National Science Foundation funding, and the same accounts say that speech on the system was monitored by the university and bounded by institutional and educational rules (Rankin 2018; Latzko-Toth 2017, as summarized in encyclopedic accounts). A program called The Enforcer was also written to run as a background process that regulated or disabled game play at most sites and times, an early case of access control based on use and not on security. These reports concern Watergate and not the Vietnam War, and they should be verified against the cited works before being relied on, but they supply a concrete instance of the tension between institutional purpose and participant purpose, and of an institution constraining a political use through its control of funding.

This essay treats the Vietnam recollection as a historical question and not as an established episode. The relevant passages of Dear's book, and primary sources such as archived Notes files, university administrative records, and contemporary student press, would have to be examined before the episode could be used as evidence for any claim. The distinction is important for method as well as for accuracy. If the episode is documented, it provides a concrete case of the tension between institutional purpose and participant purpose. If it is not documented, the general tension remains a legitimate subject of analysis, since the question of who controls the uses of a shared educational environment arises on any system that supports discussion, but the essay may not rest an empirical claim on it.

Stated in general terms, the question is how institutions respond when educational computing resources become spaces for independent political and social organization. Institutions may respond by restricting the capability, by moderating the content, by ignoring the activity, or by embracing it. Each response alters the kinds of participation the system supports, and the history of online platforms provides many variants. The central tension is between the institution's authorized purposes and the purposes developed by its participants. Part V returns to this tension as a problem of governance, using a better-documented contemporary case.

9 Architectural Possibility and the Labor of Social Invention

The emergence of social functions from educational computing infrastructure should not be understood as an automatic consequence of technical architecture. The presence of shared storage, multiple accounts, and an authoring language makes communication possible, but it does not by itself create a discussion forum, a multiplayer game, or a community.

The history of PLATO illustrates this distinction. The shared mainframe and its instructional environment supplied resources that could be appropriated for communication. Particular participants nevertheless had to recognize these possibilities, design programs, write code,

establish conventions, and persuade others to participate.

PLATO Notes and Talkomatic were inventions within an enabling environment, not inevitable outputs of the environment itself. Their emergence depended on technical affordances and on the initiative of particular users.

This distinction can be expressed in terms of a system’s space of admissible actions and the subset of those actions that participants actually develop into practices.

Let $A(x)$ denote the actions technically available in state x , and let

$$A_{\text{practiced}}(x) \subseteq A(x)$$

denote the actions that have become part of established participant practices.

A capability may exist in $A(x)$ without appearing in $A_{\text{practiced}}(x)$. Conversely, participants may combine existing capabilities in ways that create practices not anticipated by the designers.

The development of a social function therefore requires at least two analytically distinct conditions: an environment that permits the relevant actions and participants who discover, construct, and sustain a practice around them.

Governance affects both conditions. It can remove an action from the admissible set, as when an operator disables an annotation feature. It can also discourage or support practices without changing the underlying capability, as when moderation rules alter what kinds of contributions are tolerated.

The distinction is useful for interpreting the Memrise case. The MEMS interface provided a place for shared annotations, but users had to discover that the same field could serve expressive and conversational purposes. The resulting practice was neither entirely designed by the operator nor independent of the operator’s design.

The social function emerged from the interaction of affordance and appropriation.

This is also why the restoration of a capability does not necessarily restore the community that once used it. The technical possibility may return while the corresponding practice, expectations, and relationships do not.

Social media, on this account, is not simply what a platform permits. It is what participants make of those permissions over time.

Part III

An Unrealized Architecture of Social Knowledge

10 Community-Created Courses and Disciplinary Openness

Memrise, founded in 2010 by Ed Cooke and Greg Detre, began as a spaced-repetition vocabulary service built around mnemonic aids, and it grew into a platform on which users could build courses on any subject. The distinction that matters for this essay is between a platform that offers a curated set of courses authored by its operator and a platform that offers authoring tools to its users. In the second case the range of subjects is bounded only by what users find worth learning and are able to express as items and answers. On Memrise this produced community-authored courses far outside natural-language vocabulary, including courses on programming languages, mathematics, philosophy, scientific terminology, and technical frameworks. The breadth of the community course catalogue is the empirical basis for describing the platform, in its community-authored form, as general-purpose learning infrastructure. Claims about the breadth and use of the catalogue should be checked against archived listings and the dated product documentation, and the discussion here is confined to what is structurally evident from the design of the authoring system.

The unit of that infrastructure is the flashcard, and the flashcard deserves a more serious theoretical treatment than it usually receives. A card is a small, bounded, reusable knowledge object. It has a prompt and an answer, it can be placed in many decks, and it can be copied, revised, and recombined. For independent learning this makes it a convenient container for rehearsal. For social purposes it has a second property: it is a common reference point. When thousands of learners encounter the same card, they share an object about which they can have, in principle, shared experience, shared difficulty, and shared commentary. The card is a boundary object in Star and Griesemer's sense, since it serves the learner who is trying to memorize, the author who is trying to teach, and the community that is trying to collect and improve a body of knowledge, and it maintains enough identity across these uses to coordinate them.

It is important to distinguish what is documented of the platform from what can be known only through use. The structural claims in this part rest on the design of the system as described in its own documentation and as observable in archived interfaces. Claims about particular communities, courses, or episodes of use require their own sources, and where this essay relies on the general character of user behavior it says so.

11 MEMS: When Mnemonics Become Messages

The mnemonic annotation, called a mem on Memrise, is the central case of this essay. A mem is a note attached to a particular learning item that is intended to help the learner retain it, by an image, a pun, an association, or a story. Mems could be contributed by users and viewed by others, and the system presented the contributed mems for a given item to anyone studying that item, ordered in part by community approval. As a pedagogical device the mem is a standard application of the long tradition of mnemonic technique, in which vivid associations are used to anchor otherwise arbitrary pairings.

As a social device the mem has a different significance. Because it is a public free-text field attached to an object that many people see, it can bear any content its authors choose. Users inserted jokes, opinions, personal statements, and material unrelated to the item. This was not, for the most part, what the field was for. Such contributions could interfere with the pedagogical function by cluttering the space in which the learner expected help, and they demonstrate in doing so an unanticipated communicative affordance. The reverse side of the flashcard, the region that appears when the learner has answered, had become a surface on which people could address one another.

It is useful to distinguish four functions that the field could serve. Mnemonic assistance is the intended function: material designed to help a learner retain the item. Interpretive annotation is a nearby function in which the contributor supplies explanation, etymology, usage notes, or correction, and which is about the content of the item and not about retention as such. Public expression is the use of the field as a place to say something, where the item serves only as an occasion. And discussion is the case in which contributions respond to one another, so that the field becomes a thread. The four functions form a gradient and not a partition. A good mnemonic is often also an interpretive annotation, and a joke that makes a word memorable is both expression and assistance. The gradient matters because the pedagogical and the social uses share a mechanism, and governance that treats the social uses as noise will tend to damage the pedagogical ones as well.

The critical observation is that the structural conditions for social emergence were present without any feed. Learners returned to the same items repeatedly, because spaced repetition schedules reviews of the same cards. At each return they encountered the same contributions and the same contributors, so that recognition accumulated. Approval of mems provided a rudimentary reputation signal, in that contributors whose mems were repeatedly endorsed became recognizable, and contributions endorsed by many people became features of the card. The result was a form of social life organized by recurrence and not by broadcast. The relations among participants were mediated by the items they all revisited, in the manner of the object-centered model of Section 2, and the card behaved as a small social world.

This observation has a consequence for how one should think of discovery and community

on such systems. On a feed, discovery is a matter of being shown content by a ranking function, and community is a matter of following persons. On a card-centered system, discovery is a matter of meeting contributions in the course of doing something else, and community forms among people who share a body of items. The second arrangement has a different relation to attention. It does not need to compete for attention, since the participant has already come for the card, and it tends to create ties among people who share a subject and not among people who share a feed.

12 Following, Ranking, and the Construction of Comparison Groups

Memrise, like many learning services, included a competitive element in the form of leaderboards that ranked users by points earned over a period. The leaderboard comparison group was reported to be drawn from the accounts a user followed, and following was reported to be subject to an upper limit of 500 accounts. A screenshot of the separate community site taken on 27 October 2026 shows the interface refusing a further follow, with a message stating that more than 500 users cannot be followed, for an account that was following 487, which confirms that the cap is in force there at that date. A second screenshot of the same site taken the same day shows the weekly leaderboard listing nine accounts, each marked as followed, with weekly scores between roughly 137,000 and 1,021,000, and then the viewing account at rank 20 with a score of zero, followed by a link to further entries. The display is consistent with a population composed of followed accounts plus the viewer, ranked by points within a period selectable as week, month, or all time, but it does not show the intervening entries, so it supports and does not settle the reported rule. A third screenshot, taken a minute later, shows the profile of the same account listing exactly 500 followed accounts and 51 followers, with the followed accounts presented in an alphabetical list across seventeen pages, and the first page includes an account that appears at second place on the leaderboard. The figures show the cap as a binding state and not only a message, since the following count has reached the limit, and the overlap between the followed list and the leaderboard entries is a further point in favor of the reported rule. They also show the asymmetry of the follow relation, since the number of accounts followed is nearly ten times the number following back, which means that the comparison group assembled from followed accounts is chosen by the viewer and is not reciprocal. The cap is reported by long-term users to have stood at 500 for as long as the following feature has existed. The leaderboard rule is confirmed by direct use of the interface, in which only followed accounts appear in the ranking, and it has a practical consequence that the design makes visible. A learner who wants to measure progress against others with large vocabularies has to add those accounts to the followed list, since there is no other route into the comparison group, and a learner who has reached the cap can add a new comparison account only by removing an

existing one. Unfollowing therefore functions as a revision of the reference class and not merely as a social gesture, and the same act that removes a person from a feed removes a benchmark from a ranking. The analysis below is conditional on the reported design, and its value lies in the structural point, which holds whether or not the specific numbers are exactly right.

The structural point can be stated precisely. Two distinct graphs are in play. The social graph S is the relation by which a participant indicates interest in, attention to, or connection with other participants. The comparison graph C is the relation by which a participant is ranked against others, that is, the reference population for competitive standing. There is no necessary connection between S and C . A person may wish to follow many people whose activity they find interesting without wishing to be ranked against them, and a person may wish to be ranked within a group of peers who are not otherwise of interest to them.

Definition 3 (Coupled graphs). *A platform couples its social and comparison graphs when, for every participant u , the reference population in C is a function of u 's neighbors in S , so that a change in u 's social relations is a change in u 's competitive reference group.*

Coupling has consequences in four areas. For discovery, it makes the act of following a competitive act. Following a prolific learner raises the standard against which one is ranked, so that the decision to broaden one's social attention can be penalized, and the decision to follow only inactive accounts can be rewarded. For reputation, it makes the composition of one's following list a strategic variable, and so contaminates the social meaning of the list. For social scalability, any cap on following, whether imposed to bound ranking computation or to limit the burden of the social graph, becomes a cap on both functions at once, and the pressure to stay below the cap is a pressure on social participation. For measurement validity, the ranking no longer measures effort relative to a stable population, since the population is chosen by the participant and varies with social behavior.

The case illustrates a general principle of the design of object-centered social systems: the mechanisms by which people relate to one another and the mechanisms by which they are measured against one another should be separable, and the platform that identifies them has made a governance decision about the shape of its community, whether or not it recognized it as such. Section 36 develops the principle into a design recommendation.

13 Platform Enclosure and the Removal of Community Courses

The decisions by which Memrise moved community-created courses out of its main application constitute a documented case of affordance foreclosure, a term introduced in Section 20. Memrise's own announcements, as cited in the sources listed at the end of this essay, describe a sequence in which the retirement of mems was announced in November 2021, postponed from an original mid-June 2022 deadline, and carried out on the website on 5 September 2022, with

the mobile apps following in the weeks after; the community forum closed on December 8, 2023; and community courses moved out of the main application on March 31, 2024 to a separate site that was originally to remain available until at least the end of 2024 and whose availability was later extended to the end of 2025. The retirement notice urged users to save their mems beforehand, and user reports state that the original mems were not restored when the feature later returned. The sequence is described here as reported in those announcements, and the dates should be checked against the archived originals in any version of this essay prepared for publication.

Several institutional rationales are available for decisions of this kind, and the analysis does not require choosing among them. Product specialization is one: an operator may conclude that the platform is best at a particular kind of learning, such as foreign-language instruction, and that general-purpose features dilute it. Moderation cost is another: user-generated content requires review, spam control, and dispute resolution, and the burden grows with the size of the community. Instructional consistency is a third: community content varies in quality, and an operator concerned with learning outcomes may prefer content it has authored or vetted. Commercial positioning is a fourth: a subscription product is easier to describe and sell when its content is curated. Interface simplification is a fifth: each feature has a maintenance and design cost, and removal reduces both. These rationales are not exclusive, and in practice they reinforce one another.

The argument of this section does not depend on showing that any of them was mistaken, nor on showing that the decision was commercially unsuccessful. The claim is narrower. The decision changed the kinds of participation and knowledge production the platform supported. Users who had treated the service as a general-purpose learning environment, in which they could study programming languages or philosophical vocabulary or their own specialized fields, lost that capability in the main application. Contributors whose mems had accumulated recognition lost the surface on which the recognition had been earned. Communities that had formed around particular courses lost their shared space. These are consequences for the social architecture within the application, as the structural account predicts, since removing shared objects and the rules that made them visible removes the conditions under which relations among their users were mediated there. Whether the affected communities dispersed or migrated to other venues is not something the operator's announcements can show, and the claim here is limited to the loss of the shared space within the application.

The history, moreover, does not end there. According to Memrise's own later announcements and to user reports, user-created word lists began to return to the main product in 2026, together with a revised form of mems that is reported to apply to official courses and to include voting and reporting mechanisms, in contrast to the earlier open annotation surface. This development provides an opportunity to examine restoration and redesign, and in particular to ask whether recovering a feature recovers its former social function. The structural account suggests that

it need not. A restored feature may enter an environment in which the former community has dispersed or moved elsewhere, the reputation accumulated under the earlier arrangement has been lost or frozen, and the governance of the feature has been redesigned. Whether the revived feature supports the same kinds of recognition and discussion is an empirical question that the account helps to formulate without answering.

Part IV

The Comparative Architecture of Social Platforms

14 Wikipedia: Collaborative Authority and Revision History

Wikipedia exhibits the object-centered logic in its purest large-scale form. The article is the shared object, and the principal social mechanism is editing. Each article has a public face, the current text that readers see, and a set of supporting structures: the revision history, which records every change with its author and time, the talk page, on which editors discuss changes, and the policies and guidelines that regulate both. The canonical public article is therefore the visible surface of a process, and the process is available for inspection.

The distinction between the article and the negotiation behind it is the key to Wikipedia's social function. Viégas, Wattenberg, and Dave (2004) showed, in their history flow visualizations, that the evolution of articles is marked by cooperation, conflict, reversion, and long periods of stability, and that the record of these events is a resource for the people who participate. Editors do not need to agree about the interpretation of a subject to cooperate on its article, since the rules of verifiability and neutrality provide a basis for working together while holding different views. The shared object thus plays exactly the role that Star and Griesemer attribute to boundary objects: it is robust enough to maintain an identity, and plastic enough to be used differently by different parties.

Wikipedia also illustrates stigmergic coordination, in that edits left in the text guide the subsequent edits of others without any need for direct communication, and illustrates the necessity of governance. The community has developed elaborate procedures for dispute resolution, protection of contested pages, and the sanctioning of misconduct, which in Ostrom's (1990) terms are institutions for governing a commons. Identity on Wikipedia is thin and persistent at once. Contributors may be pseudonymous or identified only by network address, and yet a contributor's history of edits is a durable record that confers standing. This combination is a counterexample to the assumption that social organization requires rich profiles.

15 GitHub: Software Development as Social Media

GitHub organizes its social life around repositories and the operations that can be performed on them. Commits record changes, issues record problems and proposals, pull requests offer changes for review, forks create independent copies that can later be reconciled, and discussions provide a place for conversation that is not tied to a specific change. Each of these is a shared object in the sense developed above, and each is addressed to a project and not to a person.

The consequence is that technical work on GitHub functions at once as production, as communication, as reputation formation, and as community organization. A pull request is a unit of work and also a message to the maintainers, and its review is a conversation. A contribution history is a record of labor and also the basis on which a developer is trusted. The graph of forks and pull requests among repositories is a map of institutional relationships that no one drew. Empirical studies of the platform report that visible activity histories shape how contributions are evaluated and how collaborators form impressions of one another (Dabbish et al. 2012; Tsay et al. 2014). The platform does provide following and starring, which resemble the features of feed-based services, but the distinction between following a developer and participating in a project is revealing. Following is a relation among persons, and it is weak. Participating is a relation to a shared object, and it is strong. GitHub makes especially clear that social relationships can be organized around shared work and not around personal broadcasting.

The platform also demonstrates the value of retained histories. The version control system on which it is built can preserve an inspectable record of changes, so that contributions remain attributable and earlier states recoverable, subject to the retention and access conditions of the repository, since histories can be rewritten, repositories deleted, and access restricted. The history is not an accessory to the social life of a repository but its foundation, since it is what allows participants to disagree productively, to revert errors, and to trace the origin of decisions.

16 YouTube: Publication, Annotation, and Audience Formation

A video is a persistent social object. Once published, it can be viewed, linked, cited, discussed, and re-edited by others, and it accumulates around itself a body of comments, responses, and derived works. YouTube combines several distinct relations that are easily conflated. The relation between creator and audience is one of publication: the creator addresses an audience that may be large and anonymous, and the subscription mechanism formalizes a durable tie. The relations among audience members are conversational, and they take place in the comment threads attached to videos, which are annotations in the sense developed above, contributions addressed to an object and read by later visitors. And the relation between the platform and both groups is mediated by the recommendation system, which allocates visibility.

These three relations have different structures and different governance problems, and an adequate account of the platform has to keep them apart. The creator-audience relation is asymmetric and tends toward broadcasting. The conversation among audience members is symmetric in form, though in practice it is shaped by the visibility rules that decide which comments are seen. The recommendation system decides which videos become occasions for either. Collaborative production, in the form of reaction videos, response videos, remixes, and channel collaborations, is a further layer in which videos become objects of other videos, and interpretive communities form around particular creators, genres, and recurring problems. The platform thus contains feed-like and object-like architectures at once, which is why it serves as a useful test of any account that tries to separate them.

17 Multiplayer Games: Sociality Through Action

The preceding examples concern texts and recordings, and a theory of social media that was confined to them would leave out the most important case of action as a social medium. In multiplayer games, players communicate through construction, movement, trade, cooperation, competition, and the modification of a shared world. Minecraft, EVE Online, and Roblox are contrasting examples. Minecraft provides a persistent block world that players may modify, on servers with widely varying rules and communities. EVE Online maintains a single persistent universe in which player-run corporations, alliances, markets, and wars shape outcomes, and in which the game's economy is substantially produced by players. Roblox is a platform of games, most of them created by users, together with a creator economy. They differ in persistence, governance, player-created content, and economic structure, and the differences matter for what kinds of social organization they support. The scholarly literature on virtual worlds supports the general claim that such environments sustain real economies, institutions, and communities (Castronova 2005; Taylor 2006; Boellstorff 2008; Nardi 2010). Claims specific to the player economies of EVE Online and Roblox rest on operator reports, and a published version should cite them to those sources.

What they share is the feature that matters here: social action can be meaningful without being translated into a message. A player who builds a bridge changes the route of others. A player who destroys a structure alters a relationship. A fleet that moves to a system conveys intention to those who observe it. In each case the participants are co-present in a world whose state is the medium of their interaction, and the world supplies the shared object, the observable events, and the retained consequences that the structural account requires. Communication in the narrow sense, chat and voice, is an additional layer that supports coordination but is not what makes the system social. The theory of Section 3 accommodates these cases without strain, and the feed-centered definitions do not.

18 A Comparative Taxonomy

The systems discussed can be summarized by the shared object or environment around which they organize interaction and by the principal social mechanism they use.

System	Shared object or environment	Principal social mechanism
PLATO	Lessons and shared computing environment	Discussion, games, collaboration
Memrise	Courses, cards, mems	Annotation and learning
Wikipedia	Articles and revisions	Collaborative editing
GitHub	Repositories and code histories	Joint production
YouTube	Videos and channels	Publication and interpretation
Multiplayer games	Shared simulated worlds	Co-presence and consequential action
Twitter/X	Posts and social graph	Broadcasting and circulation

Table 1: Shared objects and principal social mechanisms in selected systems.

The table is an analytical simplification, and its value is comparative and not classificatory. Most systems support several mechanisms. Wikipedia has chat and user pages, GitHub has following and starring, YouTube has live chat, games have forums, and Twitter has long-running threads that function as shared objects. The table shows the dominant mechanism in each case and, more importantly, shows that the feed is one row among several. It also suggests a classification of mechanisms into four overlapping kinds: publication, in which contributions are addressed to an audience through an object; annotation, in which contributions are attached to an existing object; collaboration, in which contributions jointly produce or modify an object; and co-presence, in which participants act within a shared environment whose state is the medium of interaction.

Part V

Platform Governance and the Foreclosure of Possible Communities

19 Affordances, Appropriation, and Unintended Uses

The distinction between designed function and appropriated function is central to the analysis of platforms. Gibson’s notion of affordance, as adapted to design by Norman and extended to technology studies, refers to the possibilities for action that an artifact offers, and the later literature on appropriation emphasizes that users routinely find possibilities the designers did not intend, and Jenkins (2006) treats such appropriation as a defining feature of participatory

culture. A mnemonic field can become a message board. A course platform can become a political forum. A software repository can become a collaborative institution. A multiplayer simulation can become an economy. In each case the appropriated function was latent in the structure of the shared object, and it was activated by participants who found that the object could bear it.

Appropriation should not be celebrated automatically. The same mechanism that allows a mnemonic field to host a joke allows it to host harassment, spam, manipulation, and material that interferes with the original activity. The pattern is not confined to recent platforms: secondary accounts of PLATO's forums report harassment of women by a predominantly male user base (Rankin 2018). Gillespie (2018) has documented in detail how platforms govern such content, how moderation is a constitutive activity of platforms and not an accessory to them, and how the choices made in moderation shape the character of the community. The relevant question is not whether appropriation is good or bad but how governance can preserve productive emergent uses without sacrificing the original function.

Several general principles can be drawn. Governance should distinguish between the capability and its abuses, so that the response to harassment in a field is not the deletion of the field. It should make the rules of the shared object legible, since users who can predict how the system will treat their contributions can adapt to it. It should provide for the participation of the affected community in rule-making, a principle that Ostrom (1990) lists among the conditions for durable commons. And it should preserve histories, since the record of how a community used a capability is the evidence on which later decisions about the capability should be based. Section 20 considers what happens when these principles are violated.

20 The Politics of Interface Removal

The concept of affordance foreclosure is proposed here to name the elimination or restriction of a platform capability that supports forms of activity beyond the operator's preferred use. The concept is intended to be descriptive and not accusatory. Operators have legitimate reasons to remove features, and the concept does not presuppose that any particular removal was unjustified. It identifies a class of decisions that shares a structure: they change the space of possible social organization on the platform, and the people most affected are those who were using the capability in ways the operator did not foresee or did not value.

The Memrise sequence described in Section 13 is a documented instance. The retirement of community mems, the closure of the community forum, and the removal of community courses from the main application are three removals, each of which narrowed the means by which participants could address one another through shared objects. The staging of these removals, from the November 2021 announcement to the March 2024 move, over roughly two and a half years is also instructive. Each removed a different layer of the social architecture,

the annotation surface, the discussion space, and the community-authored content, and the cumulative effect was the elimination of the conditions that the structural account identifies as necessary for object-centered sociality on the platform.

The later restoration of user-created word lists and a revised form of mems, as reported by the operator, provides a second kind of evidence. It suggests that the operator concluded that the capabilities had value, and it raises the question of what is recovered when a capability is restored. The analysis above indicates three things that restoration may fail to recover. The first is the accumulated contributions, the mems and courses whose authors had built recognition, which on user reports were not restored for the mems. The second is the community, which may have dispersed or moved elsewhere. The third is the governance context, since a feature redesigned within a different set of rules may not support the same uses. A feature is a mechanism, and a social function is a relation between the mechanism and a community that uses it. Restoring the mechanism need not restore the relation.

The PLATO case supplies a contrasting instance of continuity after the loss of a platform. After the commercial service wound down, a version corresponding to the final release was resurrected online in August 2004 on an open-source emulation of the original hardware, and encyclopedic accounts report that more than 500 former users signed up within six months by word of mouth, with many describing a social bond with the community of users, a last claim that is reported without a citation and should be verified. What made the revival possible was that the software, the lessons, and the communication tools were preserved together, which suggests that community continuity depends on the preservation of the mechanism, the objects, and access to them, and on none of them alone.

21 Authority, Permissions, and the Ownership of Shared Worlds

The account so far has treated shared objects as if participants stood in comparable relation to them. They do not. A repository owner may reject contributions that others have written. Wikipedia administrators hold powers unavailable to ordinary editors. A platform operator may end an entire community's access to its objects, and a ranking algorithm may leave an annotation technically public but practically invisible. A theory of social computing has to represent asymmetric authority over shared objects and not merely shared access to them, and the Memrise case cannot be described adequately without it, since the operator controlled the conditions of the community's continued existence while contributors controlled only their individual contributions.

The governance component of the environment tuple can be made explicit as a family of functions, $G = (\text{Auth}, \text{Rev}, \text{Appeal}, \text{Exit})$. The function *Auth* determines whether a participant u may perform an action a on an object o . The function *Rev* describes the authority to revise permissions and rules, including *Auth* itself. The function *Appeal* describes the mechanisms

available for contesting decisions made under the other two, and Exit describes what participants can preserve or transfer when they leave. The decomposition separates things that are often run together. The ability to contribute to an object, which is a matter of Auth, is different from the authority to preserve the conditions under which the contribution remains socially meaningful, which is a matter of Rev, and a community whose members hold the first without the second depends on the stewardship of whoever holds the second.

Several roles can then be distinguished. Authorship is the production of contributions. Stewardship is the maintenance of an object and its conventions, which may be held by many or few. Moderation is the application of rules to contributions, with the power to hide or remove them. Ownership is control over the infrastructure in which the object exists and is the holder of the ultimate Rev. Unilateral termination is the exercise of that ownership to end the object or the community's access to it. These roles may coincide, as when a single developer owns, maintains, and writes a small repository, or diverge sharply, as on a commercial platform whose operator owns everything while authors contribute without any claim on the conditions of continuation.

The comparison is useful across kinds of system. On a centralized learning platform, contributors hold limited Auth, the operator holds Rev, Appeal is minimal, and Exit is whatever the operator provides, which in the Memrise case was a notice urging users to save their mems before a date and later an expression of interest in how courses could be exported. On a collaboratively governed wiki, Rev is distributed among a community through procedures, Appeal is substantial, and Exit is strong because the content is freely licensed and copyable. On a federated repository network, Exit is built into the architecture, since a fork preserves the object and its history under new stewardship. These arrangements are not interchangeable, and their differences account for much of the variation in how the foreclosure of a capability affects a community.

Authority also interacts with visibility in a way that the permission model alone does not capture. An annotation that a ranking function consistently places out of view is available under Auth and not situated perceptible in the sense of the previous section, and the party who controls the ranking holds an effective authority over the object that appears in no list of permissions. A full account of the authority structure of a system therefore has to include the rules by which contributions are surfaced, which belong to the visibility component P and are often controlled separately from the permissions that attract attention in governance debates.

22 Twitter, X, and the Loss of a Conceptual Metaphor

A different kind of foreclosure concerns the symbolic identity of a platform and not its capabilities. Twitter built, over its first decade and a half, a coherent metaphorical vocabulary of birds, tweets, followers, and a timeline, and in July 2023 the service was rebranded as X, with the bird

removed and the vocabulary of tweeting progressively replaced by that of posting. Van Dijck (2013) argues that platform vocabularies are not neutral labels but part of how a platform shapes the practices it hosts, and the rebranding is an unusually clear case in which one such vocabulary was replaced while the underlying architecture of accounts, posts, and following remained in place. The case illustrates the distinction between a platform’s symbolic identity and its social architecture, and it shows that the first is itself something participants share and can lose. This essay takes no position on the commercial effects of the change, which are a separate empirical matter. A conceptual essay composed before the rebranding, *What the Birds Are Tweeting* (Flyxion 2023, April), treated the bird as a structural element of the platform’s social world, and it is cited as a dated artifact of that period and not as evidence about what followed.

Part VI

Generative Learning and the Future of Shared Knowledge

23 NotebookLM and the Automated Production of Learning Objects

Generative systems can now transform documents, textbooks, lectures, and research papers into flashcards, summaries, quizzes, and audio overviews. Tools such as NotebookLM exemplify the capability by grounding generation in sources supplied by the user. The immediate application is the one most commonly discussed: more efficient memorization and study. For the present argument that application is the less interesting one.

The more significant development is the reduction in the labor required to create structured objects around which discussion can form. The community courses of earlier platforms depended on the willingness of authors to spend hours converting material into items. A system that produces a deck from a source in seconds changes the economics of that conversion, so that the supply of structured learning objects can in principle exceed anything that human authorship could provide. Whether this supply becomes a resource for social learning or a source of undifferentiated clutter depends on what is done with it.

The distinction that matters is between generating an individual study aid and establishing a persistent, publicly addressable knowledge object. A deck generated for one learner and discarded is a private artifact, and by the criteria of Section 3 it is not social. A deck that is published, versioned, and open to annotation is a shared object. The difference lies not in the generation but in the publication and governance rules, which is to say in P and G . The same

generative capability can therefore serve either isolated or social uses, and the design decisions that determine which it serves are decisions about the architecture of sharing.

24 From Generated Cards to Collective Annotation

A hypothetical system suggests itself that would combine source-grounded generation, community annotations, threaded discussions, and reusable conceptual references. A learner would supply a source, the system would generate cards grounded in it, and the cards would be published to a shared space in which others could annotate them, discuss them, and link them to related cards. The system would be a successor to the card-centered architecture of Memrise, with generation reducing the cost of creation and with annotation and discussion providing the social layer.

Several problems would have to be solved. The first is the identification of equivalent concepts across independently generated decks. Two systems given different sources may produce cards that concern the same concept in different words, and a shared annotation space is valuable only if it can recognize this without erasing differences in wording, sources, or interpretations. The task is a form of entity resolution that must avoid forcing a canonical formulation, since the differences among formulations are often what makes comparison instructive. A plausible approach is to treat equivalence as a relation that is asserted and revisable and not as an identity, so that cards can be linked as related without being merged, and annotations can attach at the level of the link.

The second problem is provenance. Every generated card should carry a record of its source, the system that produced it, and the revisions it has undergone, so that readers can evaluate it and correct it. The third is versioning, since cards will be revised and annotations that were apt for an earlier version may be misleading for a later one. The fourth is moderation, which for a generative system has a new character, since a large volume of plausible but erroneous cards can be produced faster than they can be reviewed. The fifth is contributor reputation, which should reflect the quality of corrections and explanations and not merely volume. And the sixth is the portability of annotations between platforms, so that the social layer is not captive to a single operator, a consideration that the Memrise case makes concrete: a community whose annotations lived on one platform lost them when the platform removed the capability.

25 Learning Networks as Distributed Epistemic Institutions

It is tempting to say that a community of learners annotating shared objects will converge on better understanding. The claim is not well supported as a general matter, and the history of online discussion contains as many examples of the spread of error as of its correction. The aim here is therefore more modest: to identify the conditions under which distributed annotation

can support correction, explanation, and cumulative understanding.

Three distinctions help. The first is between the authority of the source material, the interpretations supplied by contributors, and the platform’s decisions about visibility. A card derived from a textbook has a relation to the textbook that is different from its relation to a contributor’s explanation of it, and both are different from the position the card occupies in a ranking. A system that collapses these distinctions makes it difficult for readers to evaluate what they are reading. The second is between the correction of errors and the accumulation of understanding. Correction is local, and it requires only that someone notice a mistake and that the system make the correction visible and durable. Accumulation is global, and it requires that explanations build on one another, which requires structure that links them. The third is between agreement and the capacity to disagree productively. Wikipedia and GitHub show that communities can sustain cooperative work among people who disagree, provided that the shared object supplies a procedure for resolving disputes about it.

If these conditions are met, a learning network could function as a distributed epistemic institution, a community that collectively evaluates knowledge by procedures embedded in its shared objects. If they are not met, the same architecture produces a forum in which assertions accumulate without discipline. The design question is therefore not whether to have annotation but how the annotation is tied to sources, to corrections, and to procedures.

Part VII

Beyond the Screen

26 The Historical Contingency of Visual Interfaces

The assumption that computation must present its results visually is a historical contingency and not a necessity. Early computers presented results as printed output, and the graphical interface was a later development tied to particular hardware and commercial choices. Visual presentation is so entrenched that its alternatives are rarely considered, and yet it has costs: it requires the user’s attention to be directed at a surface, it excludes users who cannot see, and it reduces the range of information that can be conveyed ambiently.

Four activities need to be distinguished, which are often conflated. Observation is the acquisition of information about a situation. Interpretation is the assignment of meaning to what is observed. Representation is the presentation of the interpretation in some medium. Communication is the delivery of the representation to someone. A system may perform observation and interpretation without any representation to the user, as when a program inspects a film, analyzes an image, or evaluates a spatial environment and reports only a conclusion or an alert. The separation establishes a conceptual basis for alternative interfaces,

since if observation and interpretation can be done by the system, the choice of presentation becomes a free design variable, and the visual medium is one option among several.

27 The Auditory Memory Palace

A 2023 proposal by the present author describes an auditory operating system built on this separation (Flyxion 2023). In it, subjects are assigned trees, and birds announce their identities through sound. The user moves through an acoustic space in which entities are located, and their relations are conveyed by the qualities of the sounds, so that the interface is a spatial information environment and not a conventional voice assistant. The mnemonic tradition of the memory palace, in which items are placed at locations in an imagined space and recalled by moving through it, is the model.

The proposal is useful for the argument because it forces consideration of what a nonvisual shared environment would be. Acoustic identity, the characteristic sound of an entity, plays the role of a name or an icon. Distance, conveyed by loudness and filtering, plays the role of relevance or proximity. Urgency is conveyed by rhythm and pitch. Movement is conveyed by changes in position. Recurrence is conveyed by repetition. And environmental sound, the background of the scene, conveys the state of the system at large. The birds, trains, factories, and approaching vehicles of the proposal serve as examples of how sound can represent activity and proximity and not merely label objects. Such an interface would not be a translation of a visual one, since the properties of the medium suggest different primitives. Sound is naturally temporal and directional, and it is perceived without the need to look.

For a social environment, the implications are direct. If the entities in an auditory world include other participants and their contributions, then the world can convey presence, activity, and approach through sound, which is how physical social spaces often convey them. A shared auditory environment would be a social medium in the sense of Section 3 without any text or image.

The auditory case also tests the definition instead of merely extending it. The clause requiring observable events raises the question of whether a sound-only environment can satisfy it, since sound is transient and leaves no trace unless the environment keeps one. The answer is that the environment need not rely on the sound for retention: the state that sounds report, the positions, objects, and histories of participants, can persist while the sounds themselves are ephemeral, so that a returning participant hears the altered world. The visibility rules P take an acoustic form, since audibility depends on distance, masking, and attention, and an event that no one is positioned or attending to hear is not accessible in the required sense. The case thus shows that accessibility in Definition 1 has to be read as perceptual access and not mere availability, a reading that also bears on any visual system used by participants who cannot see it.

28 SpherePOP and the Inhabitable Computational Environment

The argument can be extended toward a computational world composed of regions, events, histories, and transformations, as in the SpherePOP framework developed in the author's related work, which is used here only through the generic features listed in this paragraph. In such a world, entities have persistent identities, regions bound scopes of activity, events are recorded in histories, and transformations are operations on regions that produce new states. These elements correspond to the components of the structural account: participants and objects with identities, admissible actions as transformations, observable events, and retained histories.

Persistent identities and event histories could support social interaction through shared activity, in the manner of multiplayer worlds, without the need for a feed. A participant who modifies a region changes the situation of those who inhabit it, and the history records the change. The same underlying structure could be represented through visual, auditory, textual, or mixed interfaces, so that a participant who sees the world, a participant who hears it, and a participant who reads a description of it share one environment and not three.

The key claim is that the social environment is not identical to its sensory presentation. The social structure is constituted by objects, actions, events, and histories, and presentation is a mapping from this structure to a medium. Different mappings serve different participants and purposes, and the choice among them leaves the social facts unchanged only when the presentation preserves what each participant can perceive and do, which is a substantive requirement and not a free choice. Put in the terms of Section 32, a presentation preserves the social structure when an action is consequential for a participant under one presentation exactly when it is consequential under the other. This is the sense in which the distinction between representation and participation is drawn in the title of this part: a representation is a view of a world, and participation is action within it. A design that confuses them, treating the screen as the world, builds the social environment into one presentation and excludes those for whom that presentation is unavailable.

29 The Problem of Representation and Delegated Interpretation

The separation of observation, interpretation, representation, and communication raises a problem that becomes acute in environments that use automated interpretation. When a system interprets a signal on behalf of a user, the interpretation is a product of the system, and the user may mistake it for a property of the signal. The distinction between a subject's actions and an observer's interpretation of those actions must therefore be preserved.

A few examples make the point. A bird’s call does not literally express an urban planning proposal, though an acoustic interface may map the call to a message about a proposal. A photograph does not disclose a person’s motivations, though a system may attribute motives to the person in it. A synthetic voice does not establish that the represented person uttered its words, though a listener may hear it that way. In each case a signal has been transformed into a representation, and a meaning has been attributed to it, and the transformation and the attribution are acts of the system and not facts about the world.

These examples motivate a theory of interpretive restraint for social computational environments. The system should preserve the difference among three things: the observed signal, the generated representation, and the attributed meaning. Each should be identifiable as such by participants, so that a participant can see what was actually observed, how it was transformed, and what meaning the system has assigned. Where meaning is attributed, the attribution should be labeled as an interpretation, with its provenance. This restraint is a precondition of trust in a social environment in which much of what participants encounter has been processed by systems they cannot inspect, and it connects directly to the problems of provenance raised in Section 24.

30 The Three Layers of Shared Reference

The discussion of representation and of auditory environments distinguished an observed signal, a generated representation, and an attributed meaning. The same distinction can be restated as three layers of shared reference, which give the conceptual bridge from nonvisual interfaces to personalized generative environments. The first is the authoritative object o , the thing the community treats as the common referent, such as an article, a card, a repository, or a state of a world. The second is the participant-specific presentation $p_v(o)$, the form in which the object reaches participant v through a particular interface under particular conditions. The third is the interpretation $i_v(p_v(o))$, the meaning that v assigns to what was presented. Shared reference is secured when participants can establish that their presentations are presentations of the same object, whatever differences in presentation and interpretation follow.

This structure explains why differences of presentation are often harmless. A card read aloud and the same card read on a screen are different presentations of one object, and a person using a screen reader and a person using a visual display share the object so long as the presentations preserve what each can perceive and do. Differences of interpretation are the normal condition of discussion, and they are the reason a shared object is useful: people disagree about it because there is something identifiable to disagree about. Neither kind of difference threatens shared reference.

A third kind of difference does. If the system supplies different objects to different participants while presenting them as identical, so that $o_u \neq o_v$ while both are labeled as the same work,

then the participants may disagree about what they believe is a single thing and be unable to discover why. This differs from disagreement about a shared object, in which the object can be consulted to adjudicate, and it differs from an inconsistency in generated content, which is a quality problem and not a problem of reference. The failure is that the layer on which the community relies, the authoritative object, has been silently replaced by a family of private objects. The translation maps introduced in the discussion of personalization are a way of stating the condition for repair: they exist when the system retains a record relating each participant’s object to a common one, and they fail to exist when it does not.

The three layers also sharpen the requirement of interpretive restraint. A system that supplies a presentation should preserve the identity of the object it presents, so that participants can ask what they were shown, and a system that attributes meaning should label the attribution as an interpretation. Both requirements are ways of keeping the layers distinct and inspectable, and both depend on retained provenance, since a participant cannot verify that two presentations refer to one object if no record of the object and of the presentations survives.

31 Personalized Generation and the Fragility of Shared Worlds

The preceding sections treated generative systems chiefly as a means of lowering the cost of creating shared objects. They also pose the opposite possibility, that the cost of producing a different object for every participant falls so far that the shared object disappears. The structural account makes the difficulty easy to state. Definition 1 requires an object or event x that is accessible to more than one participant, and a system that generates a distinct narrative, vocabulary, or world for each participant may preserve every other feature of social media, including accounts, messages, and feeds, while removing the common referent on which the definition depends. The question is whether anything in the account can distinguish harmless personalization from the loss of the conditions for sociality.

It helps first to see how widely the same computational substrate can support different media. Let a computational world be $W = (X, A, T, C, H)$, with states X , actions A , a transition relation T , constraints C , and histories H . A novel, a film, a game, and a recording of play can then be treated as projections of such a world that expose different parts of it. A novel presents a linguistically organized account of selected events, a film presents audiovisual observations along one trajectory, a game exposes some transitions to the control of participants, and a recording of play fixes one realized trajectory as a noninteractive sequence. The projections are not isomorphic. Some preserve choices and others only outcomes, and some add information that the underlying world does not contain. The claim needed here is only that they can share a generative substrate while differing in what they expose, which is also what allows the same social structure to be presented through visual, auditory, or textual interfaces, as argued above.

A second consideration is that the observable qualities of an artifact do not establish how

it was produced. Let $O : H \rightarrow Y$ be a function taking each possible history to its observable result. When $O(h_1) = O(h_2)$ for distinct histories, observing the result cannot identify which history occurred. A convincing voice does not show whether it was spoken by a performer, assembled from recordings, or generated, and a fluent account of a past conversation does not show whether the conversation was retrieved or reconstructed from a title. This is the general problem of inverse inference, and it connects directly to the interpretive restraint discussed earlier: a system that produces a plausible result is not thereby evidence of the history that would make the result accountable. For social environments the consequence is that provenance has to be recorded as such and cannot be recovered from the artifact, since an acceptable output and an authenticated record may be observationally equivalent while differing entirely in evidential status.

Personalization can now be stated precisely. Suppose participants i and j receive personalized worlds W_i and W_j . They need not experience the same things in order to communicate, but they need some way to translate between the parts of their experience that overlap. Let D_{ij} denote the portion over which the two worlds are meant to coincide, and let

$$g_{ij} : W_i|_{D_{ij}} \longrightarrow W_j|_{D_{ij}}$$

be a translation between their restrictions to it. For three participants, compatibility ideally requires $g_{jk} \circ g_{ij} = g_{ik}$ on the overlap $D_{ij} \cap D_{jk} \cap D_{ik}$, where all three maps are defined. The formulation is a proposal for analysis and is not a result, and introducing translation maps does not by itself demonstrate a formal obstruction in any actual system. Its use is to separate two things that are easily confused. Diversity of experience is not a failure of intersubjectivity, since people routinely discuss a book they read differently. The criterion proposed here is that failure occurs when the maps needed to maintain shared reference become unreliable or mutually inconsistent, so that two people believe they have seen the same work while holding incompatible evidence about its contents, and attempts to resolve the disagreement are themselves personalized. Literary treatments such as C. S. Lewis's *The Great Divorce* and the film *What Dreams May Come* dramatize versions of this concern, in which unconstrained construction of a satisfying private environment does not secure the ability to share it. Not every cultural disagreement is an obstruction of this kind, and the model should be read as describing one failure mode and not as a theory of social fragmentation in general.

The consequence for design follows from the structural account and from the earlier discussion of provenance and portability. Personalization should be accompanied by the retention of a common layer: shared objects to which personalized presentations refer, published conventions that make translation possible, and records of what each participant was actually shown. Constraints in this sense are not only external limits on creativity. A common language, a musical notation, and a programming interface all restrict what can be expressed while enabling

independently produced things to interact, and the analogous requirement for generative systems is freedom of generation compatible with shared reference and accountable continuation.

Part VIII

Formal Synthesis and Institutional Consequences

32 A Theory of Social Emergence in Computational Environments

Section 4 introduced the state-transition notation and the notion of cross-participant consequence, and this section extends them to histories. Cross-participant consequence is satisfied by a game move, an edit, a comment, and a commit, and it is not satisfied by an action that leaves the other's situation unchanged.

Cross-participant consequence is necessary for social organization and is not sufficient. A stateless environment in which each interaction is consequential only at the moment it occurs, and leaves no trace that constrains later interaction, supports only ephemeral exchange. Sustained social organization requires in addition that the history of interaction bear on future interaction. Let h_t denote the history of states and actions up to time t , and let $\pi_u(h_t)$ denote the policy by which participant u selects actions given that history, including the policy determined by the rules in G .

Proposition 1 (Proposed, hidden-dependence form). *If an environment sustains a relation whose continuation depends on prior interactions, then there exist participants u, v and histories h_t, h'_t that agree on the observable configuration but differ in the retained social state, such that Poss_v or π_v differs between them.*

The statement is offered as a theoretical proposal, and the notation does not prove it. Where the relevant history is already fully displayed, as in a public ledger of debts, the dependence is not hidden and the condition is vacuously satisfied, so that only hidden historical dependence requires observationally identical configurations with different futures, and the more general statement of the next section covers the displayed case. The observable configuration here is the part of the state that participants can currently perceive, and the retained social state is introduced formally in Section 33, where it is restated in a more general form and proved, the present statement being the special case for the observable configuration. It formalizes the thought that social structure is history-dependent: the same present configuration means different things to participants depending on what has happened, in the way that a reputation,

a norm, a debt, or an alliance is a feature of a history and not of a state. Environments in which behavior depends only on the observable configuration, with no retained social state and no dependence on past interaction, can be shared and consequential, but they do not exhibit relationship, reputation, or institution. The proposition therefore identifies a distinction between ordinary shared computation and sustained social organization, and the retained histories H of the structural model are the means by which the distinction is realized. The proposition would be tested, not proved, by examining whether the systems that sustain communities all satisfy the history-dependence condition and whether those that do not satisfy it fail to sustain them. The systems examined in this essay are consistent with it, since each retains a record, whether of revisions, commits, mems, or world state, on which participants' later conduct depends, but a study designed to test the proposition would need cases chosen independently of it.

The proposition is close to definitional, and its value lies in a distinction that the notation makes easy to state and that informal discussion tends to blur: the difference between retaining a history and using it. Consider two environments with the same states, the same actions, and the same transition function T , in which participants draw marks on a shared canvas. In the first the canvas is cleared at the end of each round, and in the second the marks persist. In both, a mark is consequential for others at the moment it is made. In the first, however, nothing is retained beyond the observable configuration, so any two histories that arrive at the same configuration are indistinguishable to every participant, and the condition of the proposition fails and no reputation, norm, or alliance can attach to the canvas. In the second the condition can hold. A subtler case separates retention from use. A platform may record every contribution in a log that no participant can read and that no rule consults, so that H is large and yet Poss_v and π_v never vary with the past. The condition fails here as well, because it concerns the effect of history on what participants can and will do and not the existence of a record. The case matters for the analysis above, since a retired annotation surface may leave the stored contributions intact while removing the means by which they bear on later conduct, which on this account is the loss that counts.

The results of this section and of Section 5 can be organized as three conditions on the progression from causal consequence to sustained organization, each stated as a definition or conditional proposition and none as a universal theorem. The first is the necessary structural condition: cross-participant consequence, as defined above, is required for any of the social phenomena under investigation, and it is not sufficient for them. The second concerns practice.

Definition 4 (Social practice). *A pattern of action on shared objects is a social practice when it recurs across occasions or participants and when participants can take its instances as interpretable, so that responses to an instance are conditioned on its being recognized as an instance of the pattern.*

Proposition 2 (Conditional: practice requires uptake). *If cross-participant consequences are*

to constitute a social practice and not merely causal interference, then there must be recurrent patterns of interaction that participants can interpret, and technical access to the shared object is not by itself sufficient.

The proposition follows from the definition and is offered as a clarification and not as an empirical law. The third condition is the one stated in the Historical Mediation proposition of the next section: sustained social organization requires that some consequence of earlier interactions remain causally available to later conduct, whether through records, environmental state, participant memory, or institutions. The three conditions are ordered but not strictly sequential, since a practice may be established by participants who remember it before any record is kept, and shared reference, as discussed in Section 30, operates across all three as a condition for mutual interpretation.

33 History Dependence and the Augmented Social State

The requirement that previous interactions influence future ones must be distinguished from the mathematical claim that social systems are necessarily non-Markovian. The latter does not follow from the former. A system may depend on its history while remaining Markovian with respect to a state representation that contains the relevant historical information.

Suppose the immediately observable configuration of an environment is denoted by x_t . This configuration might include the current positions of players, the visible text of a document, or the present contents of a shared workspace. The history of interactions may also have established reputations, obligations, permissions, expectations, and relationships that are not evident from this configuration alone.

Let z_t denote the retained social state containing such information. The augmented state is

$$s_t = (x_t, z_t).$$

The transition dynamics can then be written

$$s_{t+1} = T(s_t, u_t, a_t),$$

where u_t is the participant acting at time t and a_t is the action performed.

The transition is Markovian when the augmented state contains all information relevant to the modeled future dynamics. In probabilistic form,

$$\Pr(s_{t+1} \mid s_t, s_{t-1}, \dots, s_0, a_t) = \Pr(s_{t+1} \mid s_t, a_t),$$

with the acting participant and other relevant inputs understood to be included in the conditioning variables.

This equality does not imply that the past is irrelevant. It implies that the consequences of the relevant past have been incorporated into the present state. A reputation acquired through years of contributions can influence future decisions because it is represented in z_t . An obligation arising from a previous exchange can constrain future actions because the obligation remains part of the current social state. A governance rule established through an earlier dispute can continue to shape behavior because the rule is retained.

The distinction is analogous to the difference between a history and a sufficient summary of that history. The complete sequence of past interactions may be unnecessary for predicting future behavior if the current state contains the information that matters. Conversely, two systems with identical visible configurations may behave differently because their retained social states differ.

Definition 5 (Historically consequential state). *A social state variable z is historically consequential when its value can be modified by prior interactions and when differences in its value can alter subsequent perceptions, admissible actions, transition probabilities, or participant policies.*

The definition requires neither an explicit event log nor an irreducibly non-Markovian process. Historical information may be retained as a complete record, a compressed state, a distributed memory among participants, or an institutional rule.

The strength of a social environment can therefore be examined in terms of what information about previous interactions it preserves, how that information is made available, and how it affects subsequent conduct.

Proposition 3 (Historical mediation of sustained social organization). *For an environment to sustain a social relation whose continuation depends on prior interactions, some consequence of those interactions must remain causally available to subsequent behavior, whether through the computational environment, the participants, or an associated institution.*

Proof. Consider a relation whose continuation depends on a previous interaction. Suppose that no consequence of that interaction remains available to the environment, the participants, or any institution affecting their behavior. Then subsequent behavior cannot be conditioned on that interaction, because no causal pathway connects the earlier event to the later behavior. The continuation of the relation cannot therefore depend on the earlier interaction, contrary to the stipulated property of the relation. $\square$

The proposition is deliberately conditional. It does not assert that every social interaction requires memory or that every repeated interaction constitutes an institution. It identifies a requirement for relations whose continuation is specifically historical.

The proof is correspondingly modest. It follows from the meaning of causal dependence and does not establish a substantive empirical law about the development of communities. Its

usefulness lies in identifying the mechanisms through which history can remain operative.

An important consequence is that the absence of an explicit event log does not entail the absence of historical organization. A multiplayer community may retain norms through the memories and practices of its participants. Conversely, the presence of a complete log does not entail a functioning community. If no participant can access the log and no subsequent process uses it, its contents may be historically extensive without being socially consequential.

34 Event Sourcing, Social Memory, and the Identity of Shared Objects

The distinction between retained history and operative history becomes particularly clear in event-sourced computational systems. In an event-sourced architecture, the authoritative record is a sequence of events, and the current application state is derived from that sequence by a projection or reduction procedure.

Let

$$h_t = (e_1, e_2, \dots, e_t)$$

denote an event history, and let

$$R(h_t) = s_t$$

denote the reduction that produces the current state.

Two distinct histories may yield the same reduced state:

$$R(h_t) = R(h'_t), \quad h_t \neq h'_t.$$

The equality of their current states does not establish the equivalence of their histories. The histories may contain different contributors, different orders of action, different conflicts, or different routes to the same result.

This distinction is central to collaborative systems. Two software repositories may contain identical source files while possessing different commit histories. Two encyclopedia articles may display identical text while having undergone different sequences of dispute and revision. Two learning cards may contain the same answer while having accumulated different annotations and explanations.

At the level of the displayed object, the states may be equivalent. At the level of social history, they are not.

The event-sourced perspective makes it possible to distinguish three forms of identity. *Content identity* concerns the equivalence of the currently displayed material. *Operational identity* concerns the equivalence of the actions currently available and their consequences. *Historical identity* concerns the sequence of events through which the object acquired its present

form.

These forms of identity need not coincide. A platform migration may preserve content while changing operational possibilities. A copied repository may preserve content and commit history while changing governance. A restored annotation feature may preserve the interface while losing the community whose earlier contributions made the feature useful.

The general point is that a shared computational object may have an identity that cannot be adequately described by its present content alone, and that the social significance of a transformation depends not only on the resulting state but also on the permissions, commitments, and histories under which it occurs. The SpherePOP framework, whose event-oriented account of computational objects treats transformations and their histories as structural features and not incidental records, is one architecture in which this can be examined.

The distinction between content identity and historical identity can be sharpened by asking when two histories support the same social future. For each participant v and history h , let $F_v(h)$ denote the perceptions, admissible actions, and institutionally recognized claims available to v following h . Two histories may be called socially equivalent, written $h_1 \sim_{\text{soc}} h_2$, when $F_v(h_1) = F_v(h_2)$ for every participant v . This is a first approximation, and a stronger version would let F_v describe the set of possible future interaction traces and not only immediate opportunities. The relation does not coincide with equality of current state, since $R(h_1) = R(h_2)$ does not imply $h_1 \sim_{\text{soc}} h_2$. Two repositories may hold identical source trees while differing in contributors, licensing history, governance rules, and commitments, and two flashcard collections may hold identical questions and answers while only one preserves the annotations and corrections through which a community developed it. They are equivalent as displayed objects and not as social institutions, which is the sense in which a migration that preserves content may fail to preserve the community.

This also qualifies the common claim that a social platform can preserve a community merely by exporting its data. An export may preserve a set of objects without preserving the operations through which participants interacted with them. It may preserve comments without preserving the visibility rules that made them discoverable. It may preserve identifiers without preserving the recognition that those identifiers had accumulated.

A stronger notion of social portability would therefore require attention to the relation among content, history, operations, and governance.

Such portability need not require reproducing the original platform exactly. It requires preserving enough of the relevant structure that participants can continue to recognize, interpret, and act upon the shared objects and relationships they previously maintained.

35 Sociality Without Profiles, Feeds, or Permanent Identities

The account allows the exploration of an old question about identity. The feed-based model assumes that social organization rests on persistent personal identities with profiles. Contrary examples are plentiful. Anonymous contributors to wikis, temporary participants in multiplayer matches, and distributed collaborators who know one another only through their commits all form meaningful social structures. It is necessary to distinguish personal identity, the identification of a participant with a person, from continuity of contributions and relationships, the persistence of the participant as a locus of history.

The second can exist without the first. A pseudonymous contributor with a long history of careful edits has standing, and the standing attaches to the pseudonym. A game alliance is a relation among accounts, regardless of who operates them. In the terms of the structural account, what is required is that participants be distinguishable and that their actions be attributable to a continuing participant, which is a weaker requirement than that they be identified as persons.

The limits of anonymity should also be stated. Without any continuity, accountability is impossible, since a contribution cannot be attributed to anyone who can answer for it, and the history-dependence condition of the proposition fails for want of a continuing participant. Persistent identity becomes necessary where the stakes of misconduct are high, where the community depends on trust built over time, or where legal or institutional responsibility attaches to the contribution. The design question is therefore the minimum identity continuity that the function requires. Different systems answer differently, and the answer should be a deliberate design decision and not an inheritance from the profile-based model.

36 Institutional Design for Object-Centered Social Systems

The analysis suggests a set of principles for systems that support learning, collaboration, and community without requiring that all activity be organized around an attention feed. The principles are stated here as design hypotheses derived from the cases above, not as established findings.

Social and competitive graphs should be separable. The Memrise case shows that coupling the relation by which participants attend to one another with the relation by which they are ranked distorts both, and a system that wants both functions should let users manage them independently. Users should have control over annotations, including the ability to attach, edit, and remove their own contributions and to control how those contributions are displayed to others, since the removal of the annotation surface was the most consequential of the foreclosures described above. Shared objects should be portable, so that the community that has formed around them can leave with them if the operator changes the rules, a principle that applies to

cards, courses, annotations, and histories alike. Moderation should be designed as part of the object and not as an afterthought, with rules that are legible, procedures that are appealable, and participation by the affected community, in line with Ostrom’s (1990) account of durable commons governance and with Gillespie’s (2018) analysis of platform moderation. Provenance should be retained for every object and annotation, particularly where generative systems are involved. Interoperability should allow annotations and discussions to travel among systems that implement common standards, so that the social layer is not captive to any one of them. And the preservation of community histories should be treated as an obligation, since the history is the substance of the community’s social structure, as the proposition of the previous section suggests.

The measures by which such systems are judged should differ from engagement metrics. Time on site, session counts, and the volume of interaction measure the extent to which a system captures attention, which is the quantity that feed architectures are built to maximize. Object-centered systems should be judged by measures of successful collaboration, such as the rate at which errors are corrected and the persistence of corrections; by measures of learning, such as retention and transfer; and by measures of sustained participation, such as the continuity of contribution and the renewal of the community. These measures are harder to obtain, and they are the ones that correspond to the purposes for which such systems are built. The distinction is a form of the point that the measurement validity of a ranking depends on the population it ranks: a metric that rewards what the system was not for will shape the system toward it.

37 Conclusion: Social Media as a Property of Shared Worlds

The argument has returned to its starting point. PLATO, built to teach, produced discussion, correspondence, games, and communities, and it did so more than a decade before the commercial internet and several decades before the feed. The social functions did not wait for a social platform. They were made possible by the structure of a shared computational environment with many authors, persistent objects, and observable actions, and they were built by people who recognized what that structure allowed. The same structure, in different forms, accounts for the sociality of Wikipedia, GitHub, YouTube, multiplayer worlds, and the annotation surface of a flashcard.

The central conclusion is that social media did not begin with modern social networking platforms, nor is it exhausted by their familiar interfaces. It emerges wherever computational environments allow participants to encounter one another through shared objects, observable actions, and consequential histories. The feed is one architecture. The encyclopedia, the repository, the flashcard, the multiplayer world, and the auditory memory palace represent others, and they are not lesser or earlier forms of the same thing but different solutions to the

problem of making contributions mutually accessible.

The practical import of this conclusion lies in the governance of such systems. The Memrise sequence shows that the social functions of a platform can be removed by decisions that look, from the operator's side, like product simplification, and that restoring a feature need not restore the relation between that feature and the community that used it. Generative systems will make the production of structured shared objects far cheaper than it has been, and the decisions that determine whether those objects become the basis of communities that develop knowledge or merely a mass of private artifacts will be decisions about publication, annotation, provenance, and governance. The essay therefore ends by asking which forms of computational organization permit communities to develop knowledge, relationships, and institutions without reducing every interaction to a competition for attention. The cases examined here suggest that the answer lies in the design of the shared object and the rules that govern it, and not in the improvement of the feed.

The argument can be summarized in a single formulation. A computational environment becomes socially generative when the actions of its participants can enter one another's situations through shared objects or events. It becomes a medium of sustained social organization when those interactions acquire recognizable meanings, retain consequential histories, and operate under institutions that govern their continuation. The sociality of computation therefore resides neither in the interface nor in the communication channel alone, but in the organized persistence of mutually consequential activity.

Note on Sources and Verification

Several claims in this essay depend on sources that should be checked before publication. The account of PLATO follows Dear (2017) and encyclopedic summaries of the secondary literature (Woolley 1994; Rankin 2018; Latzko-Toth 2017; Van Meer 2003), and should be checked against those works and the CERL primary documentation, in particular for the dates and authorship of Notes, Talkomatic, and Term-Talk, for the Watergate discussion group episode and the role of the funding agency, for the figures on game hours and on the 2004 revival, and for any episode concerning Vietnam War protests, which is treated above as a question and not a finding. The Memrise dates are taken from the operator's announcements listed below and should be confirmed against archived copies. The limit of 500 followed accounts is confirmed on the community site by a dated screenshot (27 October 2026) in the author's possession, and the use of followed accounts as the leaderboard comparison population is confirmed by direct use of the interface and by the dated screenshots, though dated support documentation would strengthen it. The reports of Memrise's 2026 reintroduction of user-created word lists and of a revised form of mems, including the claim that the original mems were not restored, rest on the operator's announcements and on user reports and secondary sources that should

be checked against dated primary pages. The characterization of the breadth of Memrise’s community catalogue should be confirmed against archived listings. Whether the affected Memrise communities dispersed or migrated requires evidence from user forums and archived discussions, and claims about the scale of the player economies of EVE Online and Roblox should be sourced to operator reports or the specialist literature.

References

- [1] Avner, R. A., and Tenczar, P. (1969). *The TUTOR Manual*. CERL Report X-4. Computer-based Education Research Laboratory, University of Illinois.
- [2] Boellstorff, T. (2008). *Coming of Age in Second Life: An Anthropologist Explores the Virtually Human*. Princeton University Press.
- [3] Boyd, D. M., and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1), 210–230.
- [4] Castronova, E. (2005). *Synthetic Worlds: The Business and Culture of Online Games*. University of Chicago Press.
- [5] Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. (2012). Social coding in GitHub: Transparency and collaboration in an open software repository. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 1277–1286.
- [6] Dear, B. (2017). *The Friendly Orange Glow: The Untold Story of the PLATO System and the Dawn of Cyberculture*. Pantheon Books.
- [7] Flyxion (2023). *What the Birds Are Tweeting* (April 2023) and the auditory operating system proposal (2023). Public domain, GitHub (standardgalactic).
- [8] Gillespie, T. (2018). *Custodians of the Internet: Platforms, Content Moderation, and the Hidden Decisions That Shape Social Media*. Yale University Press.
- [9] Grassé, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. *Insectes Sociaux*, 6, 41–80.
- [10] Heylighen, F. (2016). Stigmergy as a universal coordination mechanism I: Definition and components. *Cognitive Systems Research*, 38, 4–13.
- [11] Hutchins, E. (1995). *Cognition in the Wild*. MIT Press.
- [12] Jenkins, H. (2006). *Convergence Culture: Where Old and New Media Collide*. New York University Press.

- [13] Kaplan, A. M., and Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of social media. *Business Horizons*, 53(1), 59–68.
- [14] Latzko-Toth, G. (2017). Out from the PLATO cave: Uncovering the pre-Internet history of social computing. *Internet Histories*. doi:10.1080/24701475.2017.1307544.
- [15] Memrise. Announcements concerning the retirement of community mems (announced November 2021, carried out from 5 September 2022), the closure of the community forum (December 8, 2023), the move of community courses out of the main application (March 31, 2024) to a separate site, and the 2026 return of user-created word lists and revised mems. Memrise support and community pages (memrise.com), to be cited from archived copies.
- [16] Nardi, B. (2010). *My Life as a Night Elf Priest: An Anthropological Account of World of Warcraft*. University of Michigan Press.
- [17] Ostrom, E. (1990). *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press.
- [18] Rankin, J. L. (2018). PLATO’s Republic (or, the other ARPANET). In *A People’s History of Computing in the United States*, 193–227. Harvard University Press.
- [19] Smith, S. G., and Sherwood, B. A. (1976). Educational uses of the PLATO computer system. *Science*, 192(4237), 344–352.
- [20] Star, S. L., and Griesemer, J. R. (1989). Institutional ecology, translations and boundary objects: Amateurs and professionals in Berkeley’s Museum of Vertebrate Zoology, 1907–39. *Social Studies of Science*, 19(3), 387–420.
- [21] Suchman, L. (1987). *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press.
- [22] Taylor, T. L. (2006). *Play Between Worlds: Exploring Online Game Culture*. MIT Press.
- [23] Tsay, J., Dabbish, L., and Herbsleb, J. (2014). Influence of social and technical factors for evaluating contribution in GitHub. In *Proceedings of the International Conference on Software Engineering*, 356–366.
- [24] van Dijck, J. (2013). *The Culture of Connectivity: A Critical History of Social Media*. Oxford University Press.
- [25] Van Meer, E. (2003). PLATO: From computer-based education to corporate social responsibility. *Iterations: An Interdisciplinary Journal of Software History*, 2(1), 1–22.

- [26] Viégas, F. B., Wattenberg, M., and Dave, K. (2004). Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 575–582.
- [27] Woolley, D. R. (1994). PLATO: The emergence of online community. *Matrix News*, January 1994.