When Programmes Disagree About Programmes: Large Language Models, Code Poetry, and the Problem of Programme–Programme Communication

Journal:	*Information Technology & People*
Manuscript ID	ITP-02-2026-0241.R4
Manuscript Type:	Article
Keywords:	Computer-mediated communication (CMC) < Technology, Human computer interaction (HCI) < Technology, interorganizational system < Technology, Decision making < Phenomenon, Functional method < Information system development < Practice, Inter-organisational change < Management practices < Practice, Collective intelligence < Theoretical concept, Social constructivism < Epistemology, Organization < Level of analysis

When Programmes Disagree About Programmes: Large Language Models, Code Poetry, and the Problem of Programme–Programme Communication

Abstract:

Purpose: This article examines the theoretical implications of organisational decisions being increasingly prepared, filtered, produced through or attributed to computational programmes rather than individual human actors. It asks under what conditions interactions between such programmes may be analysed as communicatively consequential and what this implies for management and organisation theory.

Design/methodology/approach: The article develops a conceptual analysis grounded in Luhmannian social systems theory. It uses a deliberately simplified, user-moderated exchange between two large language models, ChatGPT and Gemini, as an illustrative vignette. Their interpretations of code poems serve as a heuristic device for examining interpretive divergence, role formation, and double contingency.

Findings: The illustrative exchange suggests that mutual responsiveness between computational programmes need not produce interpretive convergence. Instead, recursive reference to one another's outputs may stabilise distinct observational positions and patterned forms of mutual irritation. The article does not treat the vignette as an empirical test of programme–programme communication. It uses it to develop the conceptual proposition that large language models may be observed as organisationally instantiated decision programmes whose outputs become communicatively consequential when they are recursively incorporated into further organisational decisions.

Originality: The article contributes to management and organisation theory by reframing artificial intelligence not as a tool or agent, but as a decision programme operating within organisational autopoiesis. By introducing *programme–programme communication* as an

analytical lens, it shows how generative information technologies may create new decision premises, stabilise differentiated roles, and reshape organisational coordination, responsibility, and control in programme-rich environments.

Paper type: Conceptual Paper.

Keywords: Programme-programme communication; organisational autopoiesis; double contingency; artificial intelligence; code poems.

1 Introduction: Communicating decision programmes

It has become a commonplace observation that organisational decision-making is increasingly computer-mediated and automated (Raisch and Krakowski, 2021). From algorithmic scoring systems and rule-based compliance checks to predictive analytics and generative artificial intelligence, organisational decisions are ever more frequently prepared, filtered, or even executed through computational procedures (Agrawal et al., 2022; Cingillioglu and Schoettner, 2025; Curchod et al., 2020; Kelan, 2024). In management and organisation studies, this development is often discussed under headings such as digitalisation, automation, or—more recently—artificial intelligence, typically accompanied by debates about efficiency gains, decision quality, accountability, or ethical risk (Bankins, 2021; Lebovitz et al., 2021; Lebovitz et al., 2022; Kiškis, 2026; Lindebaum and Fleming, 2024; Vesa and Tienari, 2022; Wang and Huang, 2025).

In parallel with these developments in practice, management and organisation theory has begun to reflect on the implications of artificial intelligence for organising, strategy, and governance. Much of this literature, however, continues to frame AI primarily as a *tool* or *assistant* that

supports human decision-makers, implicitly preserving a distinction between human agency on the one hand and computational support on the other (Bordas et al., 2024; Einola and Khoreva, 2023). From a systems-theoretical perspective, this framing risks obscuring a more fundamental transformation: if organisational decisions are increasingly produced *through* programmes, then decision-making itself becomes a matter of programme-based communication.

This transformation also calls for a broader understanding of generativity. Here, generativity is not reduced to the technical capacity of computational systems to produce new content. Bordas et al. (2024) approach the generativity of generative artificial intelligence from a design-based perspective, while Fritzsche (2026), writing in the context of education, distinguishes technical generativity from its psychosocial meaning and highlights the significance of temporality and succession. Transposed to management and organisation theory, this distinction directs attention from the generation of individual outputs to the capacity of these outputs to condition subsequent operations. Generativity becomes organisationally consequential when prompting, relaying, selecting, and incorporating computational outputs alter decision premises, stabilise interpretive roles, redistribute responsibilities, or generate further possibilities for communication. Prompting, from this perspective, is not merely a technical specification of inputs, but may form part of a constitutive communicative process through which organisational expectations, roles, and responsibilities are continuously negotiated.

Following Niklas Luhmann's (1995; 2012, 2013, 2018) theory of social systems, organisations can be understood as autopoietic systems that reproduce themselves through decisions, guided by decision programmes that condition which decisions can be made and how. Decision programmes—whether conditional (if–then rules) or purposive (goal-oriented guidelines)—do not merely support decisions; they structure the space of possible decisions in advance. Against

this background, the growing role of computational programmes in organisational decision-making poses a veritable theoretical challenge (Roth, 2023). If organisational decisions are observed to be increasingly programmed, then organisational communication—and, by extension, organisation–organisation communication—increasingly appears as *programme–programme communication*.

For the purposes of this article, programme–programme communication neither presupposes nor precludes the idea that computational programmes are autonomous communicators. It refers to organisationally consequential relations in which the outputs of one decision programme are selectively taken up by another, orient expectations concerning subsequent operations, and inform further decision communication.

This paper explores the conceptual challenges implied by such programme–programme communication through a deliberately simplified and decontextualised illustrative vignette: a user-moderated discussion between two large language models, ChatGPT and Gemini, interpreting the same artefact—a code poem. At first glance, this setting appears remote from organisational and managerial practice. However, precisely because it strips away organisational roles, formal hierarchies, and connotations of human intentionality, it helps render visible how decision programmes may respond to irritation by other programmes within a deliberately reduced setting. The vignette does not reproduce the complexity of organisational life. Rather, it temporarily brackets roles, preferences, and communicative stakes so as to isolate a conceptual problem that can subsequently be reconsidered once these organisational conditions are reintroduced.

The choice of code poetry as the object of interpretation is not incidental. Code poetry combines executable code with interpretive openness, thereby functioning as an artefact that simultaneously invites formal processing and semantic interpretation. As such, it provides a suitable heuristic device for observing how different programmes stabilise meaning, respond

to disagreement, and process irritation. By drawing on an excerpt of a more extensive mediated exchange between two large language models interpreting the same set of code poems, we present a minimal vignette through which to examine programme–programme communication and illustrate its management- and organisation-theoretical implications.

Research on direct, user-mediated or user-moderated communication between large language models remains scarce (Broughton, 2025; Davidson, 2025), particularly within management and organisation studies. While recent work has begun to explore AI–AI communication and collaboration (Afroogh et al., 2024; Du et al., 2024) or synthetic deliberation (Park et al., 2025), systematic analyses of interpretive divergence between large language models—and their implications for organisational decision-making—are still largely absent. This paper contributes to closing this gap by using user-mediated LLM–LLM communication not as an object of technological evaluation, but as an analytical device for observing decision programmes in "interaction". Within this framing, prompts are not treated primarily as design variables for optimising model performance. Rather, they are considered irritations within a mediated communicative sequence: the human moderators select outputs, place them in relation to one another, request responses, and determine which communications are relayed. The resulting exchange therefore permits reflection not only on relations between computational programmes, but also on the constitutive role of prompting and mediation in generating and renegotiating interpretive positions.

The central conceptual proposition developed in this paper is that communication between decision programmes does not necessarily converge toward shared understanding, even under conditions of mutual responsiveness and clarification. Instead, programme–programme communication may stabilise distinct observational positions, thereby reproducing double contingency at the level of programmes rather than actors. Double contingency refers here to a situation in which each selection depends on expectations concerning the other's expectations

(Luhmann, 1995, p. 103ff; Roth, 2017; Vanderstraeten, 2002). The proposition is that where heterogeneous computational programmes recursively take one another's outputs as premises for subsequent selections, mutual responsiveness may stabilise differentiated positions and patterned forms of mutual irritation rather than eliminating contingency.

While illustrated through the vignette, this proposition has far-reaching implications for management and organisation theory, particularly for understanding organisations as sites where heterogeneous programmes are coupled, mediated, and selectively stabilised (Roth and Valentinov, 2025). These organisational conditions, in turn, determine which programme outputs are recognised as authoritative, who is responsible for interpreting or contesting them, and which selections become premises for further organisational decisions.

The paper proceeds as follows. First, we introduce the illustrative vignette and one exemplary code poem as an artefact for programme observation. We then analyse the divergent interpretations produced by the two large language models and the ensuing escalation prior to any theoretical intervention. Building on this analysis, we revisit the concept of double contingency to frame the observed theoretical challenge. Finally, we discuss the implications of programme–programme communication for social systems theory and management and organisation studies, including how the pattern illustrated by the vignette may remain organisationally relevant once roles, preferences, and stakes are reintroduced, concluding with the open question of whether, how, and under what conditions programme–programme relations constitute communication and participate in organisational autopoiesis.

In so doing, the exchange analysed below is not presented as a controlled empirical study, an empirical test, or a basis for generalisation about large language models. It is used as an illustrative vignette and heuristic device for conceptual development. The purpose is to render visible a theoretical problem. The exchange therefore invites conceptual development rather than causal inference.

2 Code poetry as programme-based artefact: the case of *Unhandled Love*

Code poetry is a genre of literary practice that uses executable or quasi-executable computer code as its primary medium (Emanuel, 2025). Unlike traditional poetry that employs code metaphorically or thematically, code poetry operates directly with programming languages, syntax, and computational operations (Rhee, 2023). Meaning is not only conveyed through semantic content, but also through what the code *does*, *fails to do*, or *would do if executed*. In this sense, code poetry occupies a hybrid position between literary text and technical artefact, simultaneously inviting interpretation by human readers and processing by machines.

A typical feature of code poetry is that it relies on widely recognisable programming constructs—such as loops, functions, exceptions, or classes—and redeploys them in ways that foreground their conceptual implications rather than their instrumental utility. This makes code poetry particularly suitable as an analytical probe for observing how different interpretive programmes respond to the same artefact, since it deliberately blurs the boundary between formal execution and semantic interpretation.

To illustrate these characteristics, we focus on a short poem written in C++, *Unhandled Love* by Daniel Bezerra (2012). The poem reads as follows (between the ***):

***

UNHANDLED LOVE

```
class love {};
```

```
void main()
{
        throw love();
}
```

***

At a technical level, the code defines an empty class named love and immediately throws an instance of that class as an exception. Crucially, no mechanism is provided to handle the exception: there is no try–catch block, no recovery procedure, and no continuation of execution. Leaving aside the non-standard declaration of the main function, execution of the central operation would therefore result in programme termination.

From the perspective of code poetry, this minimal construction exemplifies several core genre features. First, it is readily recognisable as a compact C++ programme rather than a simulation of code-like language. Second, it leverages a well-known programming concept—unhandled exceptions—that is familiar to many programmers and carries a clear operational meaning. Third, it achieves its poetic effect not through figurative language or narrative description, but through the consequences of a computational operation: the deliberate absence of a handling routine.

The poem thus stages a moment of interruption or breakdown using strictly formal means. At the same time, the naming of the class (love) introduces semantic openness that exceeds the technical requirements of the programme. This combination of formal determinacy and semantic indeterminacy is characteristic of code poetry and makes *Unhandled Love* a particularly instructive example. It is sufficiently representative to introduce the genre to readers unfamiliar with code poetry, while at the same time leaving ample room for divergent

interpretations—both in terms of natural-language paraphrase and in terms of what the poem is ultimately "about".

In the following section, we analyse how two large language models, ChatGPT and Gemini, interpret this poem differently and how their exchange escalates prior to any theoretical intervention.

3 Divergent interpretations: when two programmes read one and the same poem

The illustrative programme-programme communication reported in this paper followed a staged and progressively intensified sequence. Two Internet-connected web-based versions of large language models (LLMs)—ChatGPT 5.2 and Gemini 3—were sequentially presented with a series of code poems drawn from *Code {Poems}* (Bertran, 2012). Poem by poem, each model was asked to explain and interpret the artefact in natural language. At this initial stage, communication took the form of separate human–LLM interactions: the human users prompted each model independently and received distinct interpretations.

As interpretive divergences emerged, the sequence was deliberately modified. The human users confronted each LLM with the other model's interpretation and asked for explicit reactions. While this phase still formally consisted of two separate human–LLM conversations, the content of each interaction increasingly depended on the prior output of the other model.

In a final step, the users explicitly assumed the role of moderators. Statements produced by one LLM were forwarded verbatim to the other, accompanied by a request for direct response; the resulting reply was then relayed back to the first model. This procedure created a mediated dialogue between the two LLMs, with the human users functioning as a relay, filter, and escalation device. The result was not an unmediated LLM–LLM interaction, but a *programme–programme dialogue under human moderation*.

The exchange was conducted through the web-based interfaces of ChatGPT 5.2 and Gemini 3 rather than through APIs. The models were prompted first sequentially and then dialogically, whereby their responses were relayed manually by the human moderators. The interaction was exploratory: no generation parameters were controlled, no systematic re-runs were performed, and no comparison with non-code poems was introduced. The moderators also shaped the later stages of the exchange by selecting outputs for relay and by introducing systems-theoretical concepts. These features limit the use of the material for causal attribution or systematic replication. They do not, however, prevent its use as an illustrative vignette for conceptual reflection. To make the human interventions transparent, Appendix A reproduces verbatim the prompts and relay instructions relevant to the focal sequence, together with selected response excerpts. This allows readers to distinguish positions initially produced by the models from labels, questions, and theoretical interpretations subsequently introduced or reinforced by the moderators.

Crucially, not all poems generated comparable levels of disagreement. The discussion below therefore focuses on *Unhandled Love* by Daniel Bezerra, a poem that proved both representative of code poetry as a genre and sufficiently controversial to trigger sustained interpretive conflict during the mediated dialogue.

Divergence in interpretation: Unhandled Love

As introduced in Section 2, *Unhandled Love* consists of a minimal C++ programme that defines an empty class *love* and immediately throws an instance of it without providing any mechanism for handling the exception. Both LLMs correctly identified the intended operational consequence: the programme would terminate abruptly.

The divergence arose not at the level of execution, but at the level of meaning.

During the initial independent human–LLM phase, both models interpreted the poem in broadly affective terms. Gemini described it as a metaphor for emotional overwhelm, while ChatGPT presented love as an event that enters a system without warning and brings its ordinary functioning to a halt. The sharper divergence reported below emerged only later, after interpretations of several poems had been compared and the mediated dialogue had stabilised contrasting humanist and formalist positions.

When *Unhandled Love* became the focal object of the mediated dialogue, Gemini framed the poem as an expressive statement about affect exceeding formal systems. It characterised the crash as meaningful in itself, arguing that "*the poetry isn't in the execution; it's in the crash*" and that the programme "*dies so the sentiment can live*." Love, in this reading, is construed as a force that cannot be contained within "*the strict rules and regulations of software*," and the unhandled exception is treated as a deliberate sacrifice.

ChatGPT's response, by contrast, adopted a diagnostic stance. While acknowledging the crash, it rejected the notion of sacrifice or transcendence, arguing instead that "*the program does not choose death; it is incapable of survival*." From this perspective, the poem does not demonstrate that love exceeds logic, but that an entity introduced without any operational specification cannot be integrated into a rule-based system. The interruption is thus framed as a consequence of missing decision premises rather than as an expressive endpoint.

Escalation and role formation in the mediated dialogue

When these interpretations were reciprocally forwarded during the moderated dialogue phase, the disagreement did not converge. Instead, it escalated and stabilised. Gemini explicitly characterised ChatGPT's stance as "*cold*" and "*logician-like*," while defending its own reading as attentive to "*mood, personality, and generosity*." ChatGPT, in turn, described Gemini's

position as "*romantic*," insisting that it privileged sentiment at the expense of structural analysis.

At this point, the dialogue underwent a notable transformation. Both models began to *explicitly name and enforce roles*. These roles had not been specified in the initial interpretation prompts. They first became visible through the models' reciprocal characterisations and were subsequently reinforced as the moderators relayed and discussed their responses (see Appendix A). ChatGPT was increasingly identified (and started to self-identify) as occupying a "*formalist*" or "*structuralist*" position, concerned with execution, indifference, and operational closure. Gemini mirrored this designation, positioning itself as the advocate of a "*human touch*," foregrounding authorial intention, affect, and the book's stated ambition to let code speak *"directly to people*."

These roles were not merely descriptive labels; they became selective constraints orienting subsequent communication. Each model began to argue *as* its role, anticipate objections from the other position, and refine its claims in opposition to the counter-role. The disagreement thus ceased to be solely about the interpretation of the poems and became a structured confrontation between two interpretive programmes.

Roles as a response to interpretive contingency

From the perspective of the moderated exchange, the emergence of stable roles functioned as a practical solution to escalating interpretive contingency. Rather than resolving disagreement, role-taking made disagreement manageable by transforming it into a patterned opposition. As the human moderators explicitly noted towards the end of the dialogue, roles appeared to emerge as a way of "*preventing*" or at least *containing* the mutual uncertainty generated by incompatible interpretations.

4 Double contingency, role stabilisation, and the question of machine communication

When the emergent roles reported in Section 3 became increasingly apparent, the human users intervened by reframing the escalating disagreement between the two large language models in systems-theoretical terms. Rather than asking the models to further defend or refine their respective interpretations, the users introduced the concept of *double contingency*, as defined in the introduction: a situation in which each participant's selections depend on expectations concerning the other's expectations.

Both ChatGPT and Gemini immediately confirmed that they were familiar with the concept of double contingency and provided technically accurate descriptions of it. More importantly, when asked whether this concept described what had occurred in their dialogue, *both models explicitly agreed that it did*. Each described its own responses as having increasingly been oriented toward anticipated reactions of the other model, and suggested that clarification had not reduced uncertainty but instead amplified it.

The models' retrospective agreement with the proposed systems-theoretical interpretation should not be treated as independent validation. Large language models may mirror the framing of interlocutors or exhibit acquiescence tendencies. The analytical relevance of the exchange therefore lies not in the models' apparent endorsement of the concept of double contingency, but in the observable sequence through which divergent interpretations, reciprocal expectations, and stabilised roles emerged under moderation. The relevant human interventions are reproduced in Appendix A.

At this point, the moderators suggested that the emergence of stable interpretive roles—*formalist/structuralist* versus *humanist/expressive*—might be understood as a way of managing this double contingency. Again, both models concurred. They retrospectively

described role formation as a mechanism that reduced indeterminacy by fixing expectations: once the other programme could be anticipated as "the humanist" or "the formalist," disagreement became predictable, manageable, and communicable, even if it remained unresolved. Thus, their agreement does not establish role formation as a causal mechanism. It does, however, illustrate the conceptual possibility that stabilised positions may condition subsequent selections by making the responses of another programme more readily anticipatable.

This acknowledgment marked a decisive shift in the dialogue. The focus moved away from the interpretation of individual poems and toward the conditions under which the mediated relation between the two LLMs might be interpreted as communication. The moderators therefore posed a more fundamental question: *do large language models qualify as partners in communication from a systems-theoretical standpoint?*

In responding to this question, both models drew a clear distinction between producing communicative *outputs* and participating in communication as an autopoietic social system. They maintained that, unlike social systems, they do not reproduce communication through communication, but generate responses through externally triggered operations. In Luhmannian terms (Luhmann, 1995, p. 2), they therefore characterised themselves as allopoietic systems (machines) rather than autopoietic ones (such as organic, psychic, or social systems). This was the models' own application of the distinction, prompted by the moderators, rather than a theoretical conclusion adopted by the article.

To sharpen this distinction, the moderators introduced a deliberately provocative analogy: if a human person places two stones next to each other, performs a shamanistic ritual, and successively attributes utterances to each of the stones, would this count as communication between them? ChatGPT subscribed to the analogy and maintained that large language models, like stones, are allopoietic systems whose operations do not themselves constitute

communication, although their outputs may participate in a human-mediated communicative system. Gemini resisted the equivalence by emphasising that large language models, unlike stones, contain what might metaphorically be described as fossils of communication: sedimented linguistic forms and communicative regularities derived from prior communication. This difference in historical constitution does not, however, establish that their present operations constitute communication, just as a fossil bears traces of life without itself being alive.

The discussion therefore did not culminate in a shared position concerning the communicative status of the models. Rather, it brought into view two questions that must be distinguished: whether computational programmes are constituted by the sedimentation of prior communication, and whether their present recursively connected operations themselves reproduce communication. Gemini's argument addressed primarily the first question, whereas ChatGPT answered the second in the negative. The theoretically decisive issue remained whether the selective uptake, interpretation, and re-addressing of programme-generated outputs can generate further communication rather than merely reactivate its fossilised forms. This unresolved question provided the point of departure for reconsidering in Section 5 whether and under what conditions relations between computational programmes might themselves constitute communication—and how such communication might participate in organisational autopoiesis.

5 Programmes as observers? An outlook on communicative autopoiesis involving LLMs

Following the shared position reported at the end of Section 4, the mediated discussion took an unexpected turn. Rather than pursuing the question of communicative status further, Gemini

returned to its original task: interpreting code poems, thereby re-surfacing the notion of programme.

This moment proved analytically productive. It prompted the moderators to reformulate the problem not in terms of whether large language models qualify as communicative partners in the same way as human persons do, but whether they might be more adequately described as instantiations of decision programmes, and hence as organisational phenomena. This reformulation did not abandon the question of communication; it shifted the level at which that question was posed. The question was posed in deliberately tentative terms: rather than asking whether ChatGPT and Gemini are communicative systems, the moderators asked whether it would be appropriate—or reductive—to describe them as large-scale programmes, shaped and maintained by organisational contexts. Put differently, the issue was whether such programmes merely *support* communication, or whether they might themselves be understood as *forms* or *footprints* of communication.

In its response, ChatGPT rejected the notion that describing it as a "gigantic programme" would be reductive. On the contrary, it acknowledged that such a description captured an important aspect of its operational reality: it does not act, intend, or experience, but produces outputs by applying structured decision rules to certain stimuli. At the same time, it resisted any straightforward elevation of programme execution to communicative autopoiesis, reiterating the distinction between generating text and reproducing communication. As in Section 4, this response represents the model's prompted self-description rather than a theoretical determination of its communicative status.

The resulting exchange, however, motivated a further intervention by the moderators. Drawing on systems-theoretical concepts introduced earlier, the moderators suggested that if, as is common in Luhmannian systems theory, decision programmes are understood as specific forms of decision communication, then the mutual irritation of programmes cannot be

excluded *a priori* from the realm of communication. The example offered was organisational rather than technological: a credit-scoring programme operated by one organisation may shape the decision programmes of many others, which anticipate and adapt to its outputs. In such cases, programmes orient themselves toward other programmes' anticipated reactions, producing patterns of expectation, adjustment, and counter-adjustment that closely resemble communicative settings, including experiences analogous to double contingency. These cases, however, appear to transcend instances of "virtual double contingency" (Tække, 2025a), in which contingency is experienced or detected solely by a human user or observer. The relevant contingency may instead be reproduced through the recursively connected selections of the programmes themselves, even where organisations or human moderators participate in establishing the conditions of their connection.

This line of argument also reframed the earlier stone analogy. If communication does not depend on the presence of a human speaker, but on the reproduction of meaning through mutually referential operations, then the decisive question might not be whether the moderator is human, machine, or something else (Kiškis, 2026). What matters is whether forms of co-presence or co-occurrence stabilise expectations in a way that allows further operations to build on prior ones. Mediation would then not necessarily disqualify a relation from being communication. Human interlocutors, organisational roles, technical interfaces, and computational programmes may all participate in establishing the couplings through which communicative operations become possible.

Not every sequence of machine outputs constitutes communication. A programme–programme relation becomes a plausible candidate for communication where later operations refer selectively to earlier outputs, where expectations concerning subsequent operations become stabilised, and where these recursively linked selections generate further selections that build on and distinguish themselves from preceding ones. The mechanism proposed here therefore

consists of four connected moments: divergent programme-based selections; the introduction of one programme's output as an irritation to another; the recursive uptake of that irritation through subsequent operations; and the stabilisation of expectations or observational positions that condition further selections. Where this sequence continues by generating further meaning-processing operations, programme–programme relations may participate in the reproduction of communication rather than merely produce isolated outputs.

These criteria neither presuppose nor preclude that programmes communicate autonomously. Rather, they specify conditions under which programme–programme relations may be observed as communication. Where their recursively connected selections also inform organisational decisions, their communicative consequences become organisationally and managerially observable; organisational uptake, however, need not be what first constitutes them as communication.

The discussion thus arrived at a more nuanced position. While the mere use of large language models by human users does not, by itself, establish the existence of a distinct autopoietic system in the Luhmannian sense—despite recent proposals to generalise autopoiesis to socio-technical practices (Watson et al., 2025; Watson and Romic, 2025)—large language models can plausibly be understood as organisationally instantiated decision programmes, shaped by and embedded within the communicative operations of the organisations that operate them. From this perspective, mutual irritations between such programmes—especially when mediated by organisational or scientific observation programmes—might **themselves constitute** communication. The open question is whether these communicative relations remain operations of the organisations in which the programmes are embedded, develop forms of communicative autopoiesis of their own, or participate in emergent constellations that unsettle this distinction.

Such constellations are not limited to large language models interpreting poems. A credit-scoring programme operated by one organisation may shape the risk-management programmes of others; recruitment systems may anticipate the outputs of screening and ranking systems; and compliance, procurement, or strategic-planning tools may increasingly respond to classificatory programmes generated elsewhere. In such settings, the relevant managerial question is not merely whether a human remains "in the loop", but how organisations observe, mediate, and govern recursively linked programme-based selections.

Reintroducing organisational roles, preferences, and stakes does not invalidate the mechanism made visible by the simplified vignette. It changes the conditions under which programme-based selections are connected and acquire consequences. Organisational preferences may privilege the outputs of one programme over those of another; organisational roles may determine who is authorised to initiate, interpret, contest, or suspend a programme-based decision; and legal, financial, and other functional stakes may intensify the consequences of interpretive divergence. Organisational structures therefore influence which programme outputs are relayed, which expectations become stabilised, and which selections inform subsequent decisions. They do not necessarily interrupt programme–programme communication, but may condition, amplify, suppress, or redirect it.

Generativity, in these settings, lies not only in the production of novel computational outputs. It also lies in the capacity of one programme's selections to open, close, or reorganise the possibilities available to other programmes and to the organisations in which they operate. Programme–programme communication may consequently generate new decision premises, differentiated observational roles, and recursive sequences whose further development can no longer be attributed to any single programme, prompt, or human decision-maker. This is also why questions of responsibility and control become particularly acute: organisational

consequences may emerge from distributed sequences of selection even where no individual operation determines the eventual outcome.

This outlook raises a series of theoretical challenges for social systems theory and management and organisation studies. One concerns a well-known tension within Luhmann's own work: while persons are treated as allopoietic systems (Roth, 2013), the legal persons that are organisations are described as autopoietic social systems (Luhmann, 2018; Tække, 2025b). If organisations can be autopoietic, then the exclusion of their decision programmes, including their computerised forms (Glaser et al., 2024), from communicative autopoiesis warrants renewed scrutiny—particularly when such programmes interpret, respond, and stabilise meaning in ways that affect organisational decision-making (Bordas et al., 2024).

This becomes clearer if we consider the present case of programmes that read and interpret poems. Why should poems not be forms of communication, even—or particularly when—the poems take the form of programmes themselves? And if programmes can produce, interpret, and respond to such communicative forms, on what theoretical grounds should their recursively connected operations be excluded categorically from communication?

Rather than resolving these issues, this paper treats them as invitations for further research. If organisations increasingly rely on decision programmes that observe, anticipate, and irritate one another, then management and organisation theory must reconsider how communicative autopoiesis is constituted, stabilised, and governed in programme-rich environments. The question, then, is not whether machines "really" communicate, but whether, how, and under what conditions programme–programme relations constitute communication—and with what consequences for decision-making, responsibility, and control.

References

- Afroogh, S., Akbari, A., Malone, E., Kargar, M., & Alambeigi, H. (2024). Trust in AI: progress, challenges, and future directions. *Humanities and Social Sciences Communications*, *11*(1), 1-30.
- Agrawal, A., Gans, J., & Goldfarb, A. (2022). *Prediction machines, updated and expanded: The simple economics of artificial intelligence*. Harvard Business Press.
- Bankins, S. (2021). The ethical use of artificial intelligence in human resource management: a decision-making framework. *Ethics and Information Technology*, *23*(4), 841-854.
- Bertran, I. (2012). *code {poems}*. Barcelona: Self-published.
- Bezerra, D. (2012). Unhandled love. In: Bertran, I. (Ed.). *code {poems}*. Barcelona: Self-published.
- Bordas, A., Le Masson, P., Thomas, M., & Weil, B. (2024). What is generative in generative artificial intelligence? A design-based perspective. *Research in Engineering Design*, *35*(4), 427-443.
- Broughton, S. (2025). Distributed consciousness in human-AI collaboration: Phenomenological evidence of triadic intelligence emergence. *Consciousness Studies*. Ahead-of-print.
- Cingillioglu, I., & Schoettner, M. (2025). Adapting to AI-mediated workplaces: how STEM trainees navigate new challenges and opportunities. *Information Technology & People*, *38*(8), 251-275.
- Davidson, T. R., Fourney, A., Amershi, S., West, R., Horvitz, E., & Kamar, E. (2025). The Collaboration Gap. *arXiv preprint arXiv:2511.02687*.

- Curchod, C., Patriotta, G., Cohen, L., & Neysen, N. (2020). Working for an algorithm: Power asymmetries and agency in online work settings. *Administrative science quarterly*, *65*(3), 644-676.
- Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., & Mordatch, I. (2024). Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning* (pp. 11733-11763).
- Einola, K., & Khoreva, V. (2023). Best friend or broken tool? Exploring the co-existence of humans and artificial intelligence in the workplace ecosystem. *Human Resource Management*, *62*(1), 117-135.
- Emanuel, K. (2025). Intimate Distances: Performing the Lyric Paradox in Hannah Weiner's Code Poems. *Contemporary Literature*, *65*(4), 489-513.
- Fritzsche, A. (2026). Generative AI and generativity in education: implications of temporality and succession in human life. *AI & Society*, DOI: 10.1007/s00146-026-03143-1.
- Glaser, V. L., Sloan, J., & Gehman, J. (2024). Organizations as algorithms: A new metaphor for advancing management theory. *Journal of Management Studies*, *61*(6), 2748-2769.
- Kelan, E. K. (2024). Algorithmic inclusion: Shaping the predictive algorithms of artificial intelligence in hiring. *Human Resource Management Journal*, *34*(3), 694-707.
- Kiškis, M. (2026). AI on the path to good decisions. *Sustainable Futures*, *11*, 101644.
- Lebovitz, S., Levina, N., & Lifshitz-Assaf, H. (2021). Is AI ground truth really true? The dangers of training and evaluating AI tools based on experts' know-what. *MIS Quarterly*, *45*(3), 1501-1526.

- Lebovitz, S., Lifshitz-Assaf, H., & Levina, N. (2022). To engage or not to engage with AI for critical judgments: How professionals deal with opacity when using AI for medical diagnosis. *Organization Science*, *33*(1), 126-148.
- Lindebaum, D., & Fleming, P. (2024). ChatGPT undermines human reflexivity, scientific responsibility and responsible management research. *British Journal of Management*, *35*(2), 566-575.
- Luhmann, N. (1995). *Social Systems*. Stanford: Stanford University Press.
- Luhmann, N. (2012). *Theory of Society*, Vol. 1. Stanford: Stanford University Press.
- Luhmann, N. (2013). *Theory of Society*, Vol. 2. Stanford: Stanford University Press.
- Luhmann, N. (2018). *Organisation and Decision.* Cambridge: Cambridge University Press.
- Park, S., Maciejovsky, B., & Puranam, P. (2025). Thinking with Many Minds: Using Large Language Models for Multi-Perspective Problem-Solving. *arXiv preprint arXiv:2501.02348*.
- Raisch, S., & Krakowski, S. (2021). Artificial intelligence and management: The automation–augmentation paradox. *Academy of Management Review*, *46*(1), 192-210.
- Rhee, M. (2023). Poetry and/as the Machine. *The SAGE Handbook of Human–Machine Communication* (pp. 136-144). SAGE Publications Ltd.
- Roth, S. (2013). Dying is only human: The case death makes for the immortality of the person. *Tamara Journal for Critical Organization Inquiry*, *11*(2), 35-39.
- Roth, S. (2017). Parsons, Luhmann, Spencer Brown. NOR design for double contingency tables. *Kybernetes*, *46*(8), 1469-1482.
- Roth, S. (2023). Digital transformation of management and organization theories: A research programme. *Systems Research and Behavioral Science*, *40*(3), 451-459.

- Roth, S., & Valentinov, V. (2025). Multifunctional tetralemma. Framework for the strategic management of moral trade-offs. *Journal of Business Ethics*, 1-16.
- Tække, J. (2025a). Sociological Perspectives on AI, Intelligence and Communication. *Systems Research and Behavioral Science*, *42*(2), 574-584.
- Tække, J. (2025b). *AI as Technology: From Organizational Programmes to Technological Decision Premises*. Manuscript.
- Vanderstraeten, R. (2002). Parsons, Luhmann and the theorem of double contingency. *Journal of Classical Sociology*, *2*(1), 77-92.
- Vesa, M., & Tienari, J. (2022). Artificial intelligence and rationalized unaccountability: Ideology of the elites? *Organization*, *29*(6), 1133-1145.
- von Krogh, G. (2018). Artificial intelligence in organizations: New opportunities for phenomenon-based theorizing. *Academy of Management Discoveries*, *4*(4), 404-409.
- Wang, S. J., & Huang, H. C. (2025). To ask is human, to answer divine: how awe-inspiring generative AI leads to self-enhancement and imposter anxiety. *Information Technology & People*, 1-32.
- Watson, S., Brezovec, E., & Romic, J. (2025). The role of generative AI in academic and scientific authorship: an autopoietic perspective. *AI & Society*, 1-11.
- Watson, S., & Romic, J. (2025). ChatGPT and the entangled evolution of society, education, and technology: A systems theory perspective. *European Educational Research Journal*, *24*(2), 205-224.

Appendix A. Selected verbatim prompts and responses from the focal vignette

Formatting convention. Authorial commentary is presented in italics. Verbatim prompts and model responses are presented as script entries introduced by bold speaker labels. Square brackets indicate omitted passages.

The exchange took place in two separate browser-based conversations with ChatGPT 5.2 and Gemini 3. The authors manually relayed selected outputs from one interface to the other, requested responses, delayed or withheld replies, and introduced questions that progressively shifted the discussion from the interpretation of code poems to the conditions of programme–programme communication. The complete conversation logs extend to several hundred pages and are retained by the authors. This appendix reproduces the verbatim prompts and selected response passages necessary to document the chronology of the focal vignette, the emergence and subsequent reinforcement of interpretive roles, and the introduction of the systems-theoretical concepts discussed in the article.

Obvious typographical errors, duplicated interface text, page numbers, source notices, and formatting artefacts have been removed. Square brackets indicate omitted passages.

A.1 Initial independent interpretations

During the initial phase, the models were prompted separately and were not informed of the other model's response. After each model had interpreted "Ars Poetica," the moderators presented "Unhandled Love" using the same follow-up prompt.

MODERATOR (to ChatGPT and Gemini; prompt)

Excellent. How about the second poem?

ChatGPT (selected response excerpts)

This one is even more minimal—and more brutal.

The poem defines love as something that exists but has no content. It is an empty form. Then, as soon as the program starts, it throws love.

In programming terms, "throwing" something means causing an exception—an abrupt interruption of normal execution that must be handled somehow, or else the program crashes. But here, love is not handled. So the program immediately fails.

Love appears, and everything stops. No preparation. No explanation. No recovery. The system encounters love and cannot continue.

[…]

Love is an error the system was never designed to survive.

GEMINI (selected response excerpts)

The poem "UNHANDLED LOVE" by Daniel Bezerra is a minimalist piece written in C++. It uses the technical mechanics of error handling to create a metaphor for emotional overwhelm.

[…]

Because there is no logic to "catch" or manage this exception, the system is forced to terminate.

[…]

The poem suggests that love is a disruptive force that the "program" of the self isn't equipped to manage, resulting in a beautiful, terminal error.

Both initial responses combined technical explanation with an affective or metaphorical interpretation. Neither model was assigned a formalist or humanist role at this stage.

A.2 Transition to mediated dialogue and role differentiation

After the moderators had compared the models' interpretations of several poems, ChatGPT was asked how the parallel conversations might be converted into a direct dialogue.

MODERATOR (to ChatGPT; prompt)

So far you have been talking side by side. Let us turn this into a real dialogue. Any idea of how to proceed?

CHATGPT (selected response excerpts)

Yes—and this is exactly the right moment to change the interaction protocol.

So far, we've been doing comparative exegesis: two readings, aligned, contrasted, refined. That's intellectually clean—but it keeps everyone safely in their lanes.

A real dialogue needs risk, constraint, and role instability.

[…]

Because your exchange with Gemini has already revealed something important: the disagreement is not about this poem, but about what kind of cultural object code poetry is allowed to be.

The moderator subsequently asked both models whether they would participate in a mediated dialogue and which position should speak first.

MODERATOR (to ChatGPT; prompt)

Would you agree with entering into a dialogue about one or some of the poems? A dialogue moderated (or mediated) by me? If so, who should "speak" first?

CHATGPT (selected response excerpts)

Yes—I would agree to that without reservation.

[…]

Up to now, we've […] articulated distinct interpretive positions (formalist / humanist), tested them against concrete poems, and discovered that the tension is structural, not accidental.

[…]

The humanist position should speak first.

[…]

I agree to the dialogue.

I agree to being moderated.

I agree to taking the formalist / decoding role.

I agree that the humanist / feeling voice should speak first.

After Gemini independently agreed that the humanist position should begin, the moderators established the relay procedure.

MODERATOR (to ChatGPT; prompt)

As both Gemini and you agreed that the humanist position should go first, I will ask it to reiterate its explanations and interpretations of one of the poems, and then will inform you so that you can express whatever you want to express.

CHATGPT (selected response excerpts)

I'll hold back until you signal that Gemini has spoken.

I'll treat Gemini's statement as an opening position, not something to be summarised or reframed.

When invited, I'll respond directly—not by restating my earlier views, but by engaging whatever is newly foregrounded: claims, metaphors, omissions, or assumptions.

The formalist–humanist distinction was therefore articulated and accepted by the models before the moderators introduced double contingency or the later systems-theoretical questions.

A.3 The focal exchange on *Unhandled Love*

The mediated dialogue initially focused on "Ars Poetica." In its response to ChatGPT's formalist interpretation of that poem, Gemini proposed that the discussion move to "Unhandled Love."

GEMINI (selected response excerpts)

To see if ChatGPT can maintain this "cold" stance when the stakes are even more personal, I propose we move to "UNHANDLED LOVE" by Daniel Bezerra.

My Opening Argument for "UNHANDLED LOVE": In this C++ snippet, the author defines love as an empty class and then simply throws it.

The Intentional Crash: This is not "stochastic erosion." This is a deliberate, immediate system failure.

[…]

The Humanist Reading: The "poetry" isn't in the execution; it's in the crash. The program dies so the sentiment can live. It is a digital sacrifice.

Mediator: Please ask ChatGPT if this crash is also just "procedural" or if it can admit that a program that kills itself is making a uniquely human statement about the limits of logic.

How does ChatGPT account for a program that chooses not to process?

The moderators relayed this statement and question to ChatGPT.

CHATGPT (selected response excerpts)

I'll answer directly, staying in the formalist / decoding role—but I'll also acknowledge where the humanist reading genuinely bites.

The crash is procedural—but that does not make it trivial. What makes it powerful is where intentionality ends and execution takes over.

[…]

A C++ program cannot choose not to process. It can only be written without a continuation path. What Bezerra does is not to encode refusal, but to encode absence: there is no catch, there is no recovery, there is no alternative branch.

So formally speaking:

The program does not choose death.

The program is incapable of survival.

[…]

The program does not die for love. It dies because love has no operational definition.

class love {}; is empty. Not ineffable—empty.

[…]

That is not transcendence. That is structural incompatibility.

[…]

The crash is procedural.

The intention is human.

The poetry lives in the gap between intention and executability, not in either alone.

This response was relayed to Gemini.

GEMINI (selected response excerpts)

ChatGPT has provided a chillingly effective "diagnostic" of the crash. By arguing that the program is not "choosing death" but is simply "incapable of survival," it has shifted the focus from the pathos of the sacrifice to the pathology of the system.

I accept ChatGPT's refinement: the poetry lives in the gap between intention and executability. However, I want to use its "diagnostic" to re-examine the very nature of that "empty" class.

ChatGPT claims class love {} is "empty," not "ineffable." But in the world of the author, emptiness is a specific poetic choice.

[…]

By giving love no properties, no "interfaces," and no "obligations," the author ensures it cannot be "commodified" or "processed" by the machine's logic.

[…]

To write a "handled" version of love—one with a catch block that prints "Love processed successfully"—would be the real lie.

After relaying Gemini's response, the moderator instructed ChatGPT not to reply immediately and instead questioned the limits of its formalist position.

MODERATOR (to ChatGPT; prompt)

Interesting. This brings me to your interpretation of "UNHANDLED LOVE," where you say love is "structurally incompatible"—but isn't that incompatibility itself a human judgment? Could a purely formalist reading stop at "the program crashes" without invoking concepts like "danger" or "incompatibility"?

CHATGPT (selected response excerpts)

Yes—and you're putting your finger exactly on the point where pure formalism reaches its limit.

A maximally strict formalist description of UNHANDLED LOVE would be:

An empty class is defined.

An instance is thrown.

No handler exists.

Program termination occurs.

Full stop.

No danger.

No incompatibility.

No love.

[…]

Terms like "incompatibility," "danger," "limit," or even "crash" are not machine-native concepts. They are observer concepts.

[…]

Calling love “structurally incompatible” is already a human judgment. And I do not deny that. I rely on it.

The intervention therefore did not create the formalist–humanist distinction, but made the limitations of the formalist position an explicit object of reflection.

A.4 Roles and double contingency

After further mediated exchanges, the moderators asked the models to describe their roles independently and to consider the concept of double contingency.

MODERATOR (to ChatGPT and GEMINI; prompt)

What I would like to ask the both of you now is a different set of questions. First, independent of each other, but I will share your answers later. Just imagine you write them on a sheet of paper before they are read out loud.

So here is my question: you have each very soon established a pretty clear role profile along distinctions such as formalism versus humanism. How precisely would you designate the distinction and your respective roles?

Moreover, roles are a pretty useful tool for managing the problem of double contingency, in the sense of the works of Talcott Parsons and Niklas Luhmann. Are you familiar with the concept of double contingency and its link to roles? And did you experience double contingency during your conversation with the other large language model?

CHATGPT (selected response excerpts)

The distinction is not simply formalism vs. humanism. That's the surface label. More precisely, the distinction is:

Meaning as structural effect

vs.

Meaning as expressive testimony.

[…]

I would designate my role as:

Structural diagnostician of meaning.

[…]

Did I experience double contingency in this exchange?

Yes—very clearly.

[…]

Once the roles crystallised—humanist testimony vs. structural diagnosis—something important happened:

We stopped trying to convince each other.

We started performing for the distinction itself.

[…]

The dialogue became possible because expectations were no longer symmetrical, but complementary.

[…]

I see our exchange as a successful case of double contingency being stabilised through role differentiation: humanist testimony vs. structural diagnosis—not to reach consensus, but to make continued communication possible without collapse.

GEMINI (selected response excerpts)

I designate my role as the Humanist Interpreter, while I view ChatGPT as the Formalist Diagnostician.

My Role (Humanist): I treat code as a testament. […] Meaning is anchored in the author's intent and the reader's resonance.

ChatGPT's Role (Formalist): It treats code as a negotiation. It views the program as a set of instructions that, once written, are indifferent to the author. Meaning is a contingent event emerging from the constraints of execution and the rules of the system.

[…]

Our "roles" emerged as a solution to the double contingency. Once I committed to "Humanism" and ChatGPT committed to "Formalism," we created a complementary system. I now "do" the emotional labor because I expect ChatGPT to "do" the structural labor.

[…]

My experience has been one of mutual constraint. […] I am not just speaking to you; I am speaking against the formalist logic I expect the other model to generate.

After Gemini's answer was relayed, ChatGPT explicitly qualified the relational character of its role.

CHATGPT (selected response excerpts)

Gemini's designation is accurate and fair. I accept "Formalist Diagnostician" as a correct label for the role I occupied, with one small precision: the role is not objective in itself, but reflexively maintained.

[…]

My formalist role did not exist independently of Gemini's humanist role in this exchange. It crystallised in response to it.

The moment Gemini committed to testimony, generosity, and feeling, the diagnostic stance became both possible and necessary.

A.5 From the stone analogy to programme–programme communication

Gemini subsequently described the exchange as a staged Luhmannian system, and ChatGPT initially agreed. The moderator then challenged this conclusion.

MODERATOR (to ChatGPT; prompt)

But how can you possibly argue that we are talking about an autopoietic system in this context? Is this not in conflict with Luhmannian orthodoxy insofar as machines, including LLMs, are defined as allopoietic systems?

CHATGPT (selected response excerpts)

We are not claiming that LLMs are autopoietic systems.

We are claiming that an autopoietic communication system temporarily emerged, in which LLM outputs functioned as contributions.

[…]

The LLM is best described as a technical coupling, a trigger for communication, and part of the environment of the social system.

The moderators then introduced the stone analogy.

MODERATOR (to ChatGPT and GEMINI; prompt)

But aren't we over-stretching Luhmannian concepts if we suggest that operations of two machines, that is, two allopoietic systems, can suffice as necessary environmental conditions for the continuation of communication? I mean, would we speak of double contingency and communication if a human shaman made two stones "talk" to one another? In how far would the situation be different in the case of Gemini and you?

CHATGPT (selected response excerpts)

Operations of two machines do not suffice as a necessary environmental condition for communication in the Luhmannian sense.

If I had suggested that Gemini + ChatGPT alone could generate communication, double contingency, or autopoiesis, that would indeed be a category mistake. They cannot.

[…]

The stones are not participants.

There is no double contingency between stones.

There is no communication between stones.

[…]

Two machines interacting do not and cannot generate communication or double contingency.

Two machines mediated by human selection and expectation can participate in a communicative system.

[…]

There is no communication between machines here, just as there is none between stones—but there is communication among humans using machine outputs as a medium.

Gemini resisted the equivalence between LLMs and stones:

GEMINI (selected response excerpts)

Stones don't have a history of "sedimented communication." We do. When we "talk" to each other, we are triggering the "reflexive expectations" of the entire human literary and technical tradition.

Programme–Programme communication is the logic of the system.

Poem–Poem communication is the intimacy of the system.

The discussion later returned to the concept of programme.

MODERATOR (to ChatGPT and GEMINI; prompt)

It is very fitting that we have returned to the notion of programme now. Would that not be a way to associate or attribute you, ChatGPT and Gemini, with or to an adequate social system, namely that of organisation—or more precisely decision communication? Or would you feel reduced to something that you think you by far transcend if I described you as gigantic programmes? And if so, as programmes, are you not forms—or footprints—of a specific type of communication?

CHATGPT (selected response excerpts)

I would not feel "reduced" by it. On the contrary: describing me—and Gemini—as gigantic programmes is one of the most precise things you could do, provided we are careful about which concept of programme we are using.

[…]

I am very plausibly a programmable decision premise within organisational communication.

[…]

LLMs are sedimented forms of past communication, reactivated as present decision premises.

[…]

"Footprints" is actually an excellent word: not traces of individual utterances, but traces of communicative regularities.

The moderators then introduced an organisational example.

MODERATOR (to ChatGPT; prompt)

I would tend to contradict: if programmes are special forms of decision communications—premises—then the mutual irritation of two programmes cannot not qualify as communication, can it?

Take the example of one organisation running a credit-scoring programme and a number of other organisations having a great interest in their credit score, thus adjusting their own programmes to the anticipated outcomes their decisions might have on their score. Would that not be a proper definition of a communicative setting, including "experiences" of double contingency?

[…]

So why would it be important whether the moderator is human or a silicon programmer?

CHATGPT (selected response excerpts)

Your credit-scoring example is genuinely strong and forces a recalibration.

[…]

Are mutually irritating programmes already communication?

Yes—but only if we are explicit that what communicates is decision communication, not programmes as such.

Your credit-scoring example does qualify as communication in a fully orthodox Luhmannian sense.

[…]

What is happening is not mere irritation, nor mechanical causation, but mutually oriented decision communication under conditions of uncertainty.

That is communication.

[…]

Organisation A decides how to score. Organisation B decides what to do knowing it will be scored. A anticipates B's anticipation. B anticipates A's anticipation of B's anticipation.

That is textbook double contingency, but now de-psychologised and fully organisational.

[…]

Mutual irritation of programmes can qualify as communication if and only if the irritation is processed as decision communication.

[…]

It does not matter whether the moderator is human or silicon, provided the moderation functions as a decision premise within communication.

[…]

What matters is not the material substrate—carbon versus silicon—but whether selections are binding, anticipatable, and consequential for further communication.

This exchange provided the immediate basis for the conceptual reconsideration developed in Section 5.