Perspective Paper

AI for Office Work

Bridging the Gap Between Natural Language and Deterministic Processing

Dave Raggett [1]

[1] ERCIM; dave.raggett@ercim.eu; ORCID: 0000-0002-5797-758X

Abstract

Agentic AI will transform traditional rigid IT solutions by introducing sophisticated software agents capable of managing complex business processes and difficult corner cases, only escalating to human oversight when necessary for difficult or high-risk cases. This is possible because **advanced neural networks are strong at understanding the context, semantics, and inherent vagueness of natural language**, which is how most business processes and legal documents are expressed—a task that knowledge graphs and conventional databases handle poorly. **Neural networks are weak in precise deterministic operations like counting and aggregation.** This can be addressed with a hybrid approach where natural language documents are augmented with **persistent, structured annotations**. This integration supports powerful complex open-ended queries that combine deep intent-based understanding with iterative processing, such as *"How many of our loans are at risk of defaulting?"* and *"How many of our local trains are at significant risk of failing in the next 24 hours?"*

Keywords: Neural Networks; Agentic AI; Semantic Orchestration; Business Processes, Natural Language; Open Texture

1. Introduction

Current approaches for implementing business processes are brittle and expensive to change. Forms and database models are inflexible and have difficulties with the inevitable corner cases. Recent work has looked at using large language models (LLMs) to translate natural language queries to database query languages such as SQL, SPARQL or CIPHER. To avoid hallucinations, RAG (retrieval augmented generation) can be used to ground LLM responses on knowledge graphs, and other resources, e.g. to ensure citations are real rather than hallucinated.

However, most business processes and legal documents are expressed in natural language and rely on terms like *material breach* and *best efforts* that are deliberately vague, enabling their applicability to be interpreted in context of commercial norms. Work on the Semantic Web has presumed that we need richer formal representations of knowledge, but that has been largely overtaken by progress in neural networks, which are proving highly effective at manipulating knowledge represented in natural language.

Advanced neural networks are strong at understanding the context, semantics, and inherent vagueness of natural language, which is very difficult for knowledge graphs and conventional databases. Neural networks are weak in precise deterministic operations like counting and aggregation. This can be addressed using hybrid approaches that rely on AI to handle the semantics, and conventional approaches for deterministic operations over persistent structured annotations for written documents.

Semantic orchestration is used to split tasks over multiple agents and services, only escalating to human oversight for difficult or high-risk tasks. The approach described in this paper provides greater flexibility and adaptability compared to traditional IT practices, and moreover, reverses the de-skilling and de-humanisation of office work in the nineteenth and twentieth centuries.

To address these challenges, this paper proposes a layered architecture involving four tiers, see Figure 1.

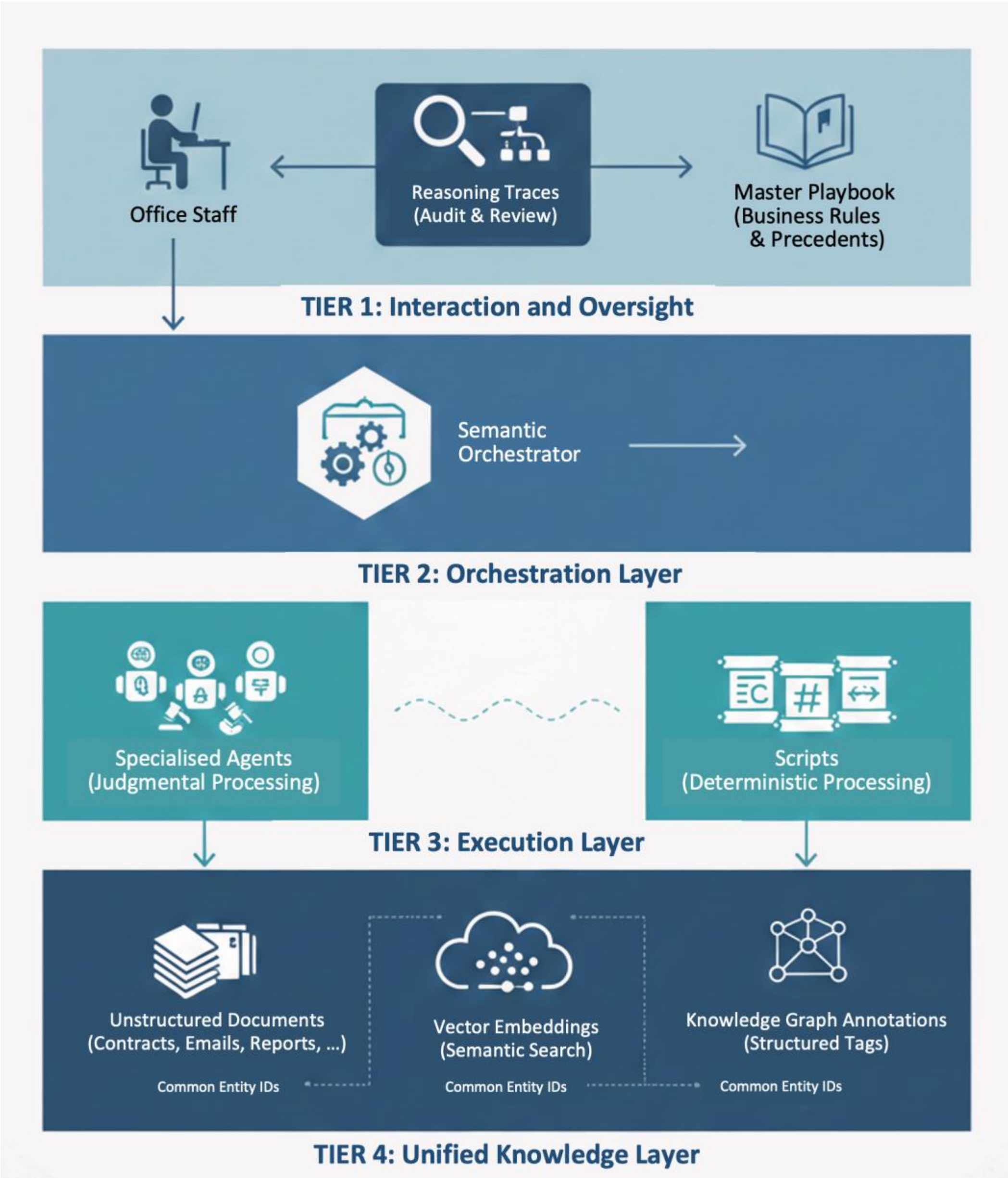

Figure 1. Tiered Architecture

- Tier 1: Interaction & Oversight: The gateway where Office Staff provide high-level goals and review the Reasoning Traces generated by the AI as recorded in the audit trail.

- Tier 2: Orchestration Layer: The Semantic Orchestrator resides here, utilizing a Master Playbook of business rules and precedents to guide sub-agents.
- Tier 3: Execution Layer: A pool of Specialized Agents and Scripts. Agents handle "judgmental processing" while scripts handle "deterministic processing".
- Tier 4: Unified Knowledge Layer: The bottom tier where Unstructured Documents are linked to Vector Embeddings and Knowledge Graph Annotations via common entity IDs

This paper is arranged as follows: section 2 looks at what we can learn from offices before computers, section 3 considers the importance of language, section 4 surveys the data foundations, section 5 looks at agentic architecture, section 6 describes the impact on business, before the overall conclusions in section 7, followed by the references.

2. Offices before computers

James Beniger [1] argues that the information age began in the mid-nineteenth century as the speed of steam power and industrialisation created a crisis of control that could only be addressed with new bureaucratic technologies: the telegraph, the typewriter, the vertical filing cabinet and standardised forms. JoAnne Yates [2] describes how businesses transitioned from oral traditions to internal memos and reports. Earlier authors, e.g. William Henry Leffingwell [3], inspired by Frederick W. Taylor's work on factory management, extended the Taylor system to office work, featuring time & motion studies, task and bonus plans, and arranging the office floor plan to expedite routing of working documents. Harry Braverman [4] noted that the industrialisation of offices de-skilled workers with repetitive rule-bound tasks.

Figure 2. Offices before computers

The vertical filing cabinet invented in the late nineteenth century had a dramatic impact on the speed of access to written records using neat categories for easy lookup. Office staff followed business rules to process documents in a repetitive work flow. Higher skilled staff could be asked to work on special reports, using their judgment to respond to open-ended requests, e.g. in respect to business challenges and opportunities.

Offices before computers can be used for inspiration in respect to the potential for AI powered office work. AI agents can be given particular roles in the overall workflows, working in tandem with office staff. AI can take over the routine work, leaving office staff to handle difficult or high-risk cases. AI can be tasked with filing records and annotating them with the information needed for speedy retrieval. For non-routine work, office staff can guide AI agents to research and prepare reports. A Harvard Business Review article [5] suggests that whilst AI isn't yet ready for direct customer facing applications, AI is well suited for internal processes, taking on mundane tasks and leaving human staff to focus on tasks requiring human judgment, including preparation and oversight of AI applications. Used in this way, AI will boost throughput and flexibility, supporting and enriching rather than obsoleting human roles.

3. The importance of language

3.1. Ambiguity and vagueness are necessary

Natural language thrives on terms that are deliberately vague, allowing judges, regulators, and business partners to apply them reasonably to unforeseen circumstances. Mathematical terms, conversely, are designed to eliminate all ambiguity.

Hart [6] describes laws as a system of social rules with an accepted core along a periphery expressed with a level of deliberate vagueness that judges can interpret to adapt to changing circumstances that were unforeseen when the law was drafted. He refers to this as the open texture of law and a necessary and desirable feature of a healthy legal system.

Concepts like "reasonable person," "good faith," "due diligence," and "unjust enrichment" are cornerstones of law. You cannot assign a numerical value or a precise logical formula to "reasonable" because its definition must adapt to the context of the time, the industry, and the specific facts of a case. A mathematically defined rule would struggle to account for the difference between negligence (a failure to use "reasonable care") in a 1950s factory versus a modern automated warehouse.

Business contracts often use terms like "material breach" or "best efforts." These terms implicitly allow for a range of acceptable actions and are meant to be interpreted in light of commercial norms, not a rigid IF/THEN statement.

3.2. Dynamic, Incomplete, and Contradictory Systems

Mathematics relies on a stable set of axioms assumed to be consistent and complete. Legal systems, however, lack this static foundation. Their 'axioms' are split between two traditions: the judicial precedents of common law and the statutory codification of civil law. Because these systems must constantly reconcile new regulations with conflicting jurisdictional layers, they remain inherently 'incomplete'—a state that mirrors Gödel's Incompleteness Theorems. Just as formal logic contains statements that cannot be proven within the system, legal systems contain logical gaps and contradictions that require human judgment, rather than pure calculation, to bridge.

3.3. Dealing with Intent, Ethics, and Emotion

Many core legal and ethical concepts involve internal human states that defy quantification.

Intent (Mens Rea): Criminal law depends heavily on the defendant's mental state—was the action intentional, reckless, or accidental? These are non-numerical categories. A mathematical model can track actions, but it cannot formally define the difference between murder (intentional) and manslaughter (reckless or accidental).

Public Policy: Legal decisions are often grounded in concepts like fairness, equity, and social utility. These are qualitative, moral judgments that prioritize social values over logical derivation. Mathematics can describe utility in economics, but it cannot dictate the moral value of a law.

3.4. The Human Element in Enforcement

Mathematics is specialized and requires technical proficiency. Law and business must be accessible to a wide audience—citizens, jurors, employees, and customers.

Argumentation: Natural language allows for argumentation, where premises are established, exceptions are debated, and rhetoric is used to persuade. Mathematical proof is deductive and leaves no room for debate once the premises are accepted.

Accessibility: A complex legal statute or contract written entirely in set theory notation or predicate logic would be impenetrable to the average person, defeating the purpose of transparency and accountability in governance and commerce.

In summary, while quantitative methods (statistics, financial modelling) are essential within law and business, the overarching framework—the rules, rights, and relationships—must be expressed in natural language. This necessity reflects that these systems are designed to govern human behaviour in a complex, dynamic, and ethical world, not to model a closed, abstract reality.

3.5. The Indivisibility and Non-Quantifiable Nature of Rights

Many fundamental legal and business entities are indivisible qualitative concepts, not measurable quantities.

Contractual Status: A contract is either valid or voidable; a person is either liable or not liable. While a mathematician might represent this as a binary (1 or 0), the reasons for the binary classification are complex legal arguments, not calculations.

Property Rights: A right to property, a patent, or a copyright cannot be meaningfully expressed as an equation. You can calculate the economic value of a patent, but the legal existence and scope of the right is defined entirely by legislative text and judicial precedent.

3.6. The Problem of Abstraction from Reality

Mathematics creates highly abstract models of reality, but law and business must operate directly on the messy reality of human behaviour and history.

No Ceteris Paribus: Mathematical models in economics often rely on the assumption of ceteris paribus (all other things being equal), rational actors, or perfect information. Law and business deal with irrational actors, incomplete information, unforeseen circumstances (Force Majeure), and changing regulations—the very elements mathematical models often try to abstract away.

Defining the Input: For an equation to work, the variables must be perfectly defined. In law, the definition of the variables (e.g., "What constitutes a 'material breach' of this specific contract?") is often the core of the dispute. Mathematics is a system for manipulating given inputs; law and business are systems for defining the inputs.

3.7. Evolutionary and Adaptive Nature of Concepts

Beyond their structural differences, legal and business concepts function as adaptive organisms rather than fixed definitions. This evolution follows two distinct paths: in common law, adaptation is organic and retroactive, where legal 'stories' mutate incrementally through specific disputes. In contrast, civil law adaptation is structural and proactive, evolving through the periodic legislative recalibration of its architectural framework. While one system adapts by updating the narrative (**stare decisis**) and the other by re-drafting the blueprint (**jurisprudence constante**), both ensure that a legal variable—unlike a mathematical constant—is always a function of its technological and societal era.

Regulatory Adaptation: New technologies (like AI or crypto) introduce concepts that legislation must scramble to define. The legal meaning of "asset" or "person" is currently changing—a process of linguistic and regulatory negotiation, not mathematical recalculation.

3.8. Where Mathematics Does Intersect

It's important to note that mathematics is not entirely absent. It is a crucial tool within law and business for specific, quantifiable tasks:

Finance & **Accounting**: Calculating taxes, interest rates, present value, damages, and financial risk models (e.g., Black-Scholes).

Statistics: Analysing market data, predicting consumer behaviour, or proving statistical discrimination in legal cases.

Optimization: Supply chain management, logistics, and resource allocation.

In short, mathematics can calculate the consequences or measure the scale of an event (e.g., "how much compensation is owed"), but it cannot define the fundamental legal or business relationship itself (e.g., "why compensation is owed and under what circumstances").

4. Data Foundations

4.1. Language as a query translator

Advanced neural networks excel at interpreting human intent, even with ambiguity, and can be trained to translate natural language queries into precise, formal query languages. The user might write: "Find all customers in California who spent over $500 last month on product X, and their associated sales agent."

The agent translates this request into a structured query (e.g., SQL for a relational database, Cypher for a property graph, or a SPARQL query for a RDF triple store), and can further help with optimising database performance.

The database executes the formal, optimized query and returns the exact data to the agent, which interprets the structured data and converts it back into a conversational, human-readable response.

This lowers the barrier to entry for non-technical users, achieving "data democratization" without sacrificing the rigor, performance, and integrity of the underlying database.

4.2. Graph Databases for Structured Facts

Grounding: Natural language queries need to be grounded on the facts. This is analogous to how humans consult written records rather than relying on fallible memory. Agents based upon large language models can use techniques like Retrieval-Augmented Generation (RAG) to insert the results from database queries into the language model's prompt. When asked, "What is the relationship between product X and supplier Y?", the agent can query the corporate knowledge graph for the explicit, auditable connections, eliminating hallucination.

Reasoning: Many complex business or legal queries require traversing multiple, defined relationships (e.g., "Find all subcontracts related to a parent contract whose legal jurisdiction is in a specific country."). Graph databases handle this multi-hop reasoning orders of magnitude more efficiently and reliably than a probabilistic Neural AI.

Context for Agentic AI: For an agent to execute a complex, declarative business process (as discussed earlier), it needs a structured way to understand the relationships between tasks, resources, and actors. A Knowledge Graph is the perfect model for this Agentic Workflow Management.

See Lewis [7] for an account of using a vector index of Wikipedia to ground responses on Wikipedia entries. Edge [8] extends RAG to support summaries over a knowledge graph, whilst Zhang [9] systematically analyses the technical foundations of RAG, identifying key technical challenges and promising research directions.

4.3. Limitations of Graph Databases in Legal and Business

Missing Nuance: The original contract text might say, "This agreement is valid in California, except for section 4.b, unless a force majeure event occurs." This complex, conditional legal nuance is difficult to fully encode as a simple, unambiguous edge property without losing the original legal intent.

The "Black Box" Problem: Once complex language is reduced to a set of formal nodes and edges, the original reasoning behind the structure is lost. If an auditor asks why two nodes are connected, the answer is just "because the data model says so," not "because the specific language in Clause 3.a of the contract stipulates this relationship."

Maintenance Overhead: When a new legal precedent or regulation introduces a new clause (e.g., a "good faith" requirement in a new context), the entire Graph Schema may need manual, labour-intensive restructuring.

4.4. The Future: Hybrid Systems and Vector Integration

Vector databases (or vector extensions to traditional databases) allow the database to store and query not just structured data, but also the semantic meaning of text, images, or code (represented as mathematical vectors).

New database platforms are emerging that can combine relational/transactional data for speed and consistency, graph data for complex relationships, and vector data for semantic search and unstructured content.

In such environments, language models will be an embedded, fundamental component that enables everyone to interact with the data simply by stating their goals and constraints.

4.5. Semantic Annotation in Hybrid AI

Many business processes can be implemented in terms of work flows that generate and operate on business records. Digital records have taken over from paper records, and form a mix of structured and unstructured information. Such records include emails, letters, contracts, orders, reports, tabular data, graph data and JSON. One challenge is how to keep track of these efficiently analogous to books in a library. The answer is a consistent system of identifiers (common entity identifiers) and a convenient means for indexing records for retrieval.

Large Language Models are now being routinely used to generate metadata for business records based upon the context, for example, when distinguishing between a billing address and a shipping address. Structured information often involves cryptic names for properties and for values. An LLM can analyse the context to expand these to meaningful descriptions. For example, a data column named LST_CONT_DT might be expanded to “Last Contact Date: the date of the most recent interaction.”

The metadata of interest depends on the business context, for instance, expiration dates for contracts, sentiment for customer feedback, invoice numbers for financial records and classification acronyms for data catalogues.

LLMs can occasionally “invent” metadata that isn’t present in the record. This necessitates a means to verify metadata for high stakes documents, e.g. placing a human in the workflow to review and sign off the processing. Compliance with privacy regulations like GDPR and HIPAA preclude the use of public LLM APIs due to concerns about controlling access to personally identifiable information, and the need for explainability. Article 12 of GDPR, for instance, requires explanations to be provided in a "concise, transparent, intelligible and easily accessible form, using clear and plain language." Private LLMs are also needed for handling company confidential information. They can be executed either onsite or on trusted 3rd party premises.

Query Pattern	Vector Search Difficulty	Annotation Augmented
Exact Filtering	Difficult. Relies on vector similarity, which can be noisy for exact matches.	Annotation: Tag the relevant paragraph with metadata like {"Contract_Type": "MSA", "Effective_Date": "2024-01-01"}. Querying is now a fast, deterministic lookup filter (e.g., "Filter on Contract_Type = MSA").
Relational/Multi-Hop	Impossible. Vectors only measure similarity; they don't model explicit relationships.	Annotation: Create an entity and a link. Tag the name "Acme Corp" with an entity ID and an edge HAS_PARENT_COMPANY linked to "Global Holdings." This creates a mini-Knowledge Graph overlay.
Aggregation/Counting	Breaks down entirely (see section 5.1).	Annotation: Extract quantitative facts (e.g., tag "The fee is €50,000" with {"Fee_Amount": 50000, "Currency": "EUR"}). The AI can then execute formal arithmetic queries against this structured data (e.g., "Sum all Fee_Amount where the document is a Sales Order").

To extract the metadata, the LLM needs careful prompting that depends on what metadata concepts, properties and relationships need to be extracted, as well as the output format, e.g. JSON or RDF triples. As such this leads to a collection of specialized agents that are designed for specific kinds of records and business contexts. Each agent is given the vocabulary to use along with examples of its use.

Here is a simplified example of the kind of prompt needed for metadata extraction:

> You are a professional records manager and data extractor. Your task is to analyse the provided business document and extract metadata into a structured JSON format.
>
> Use the following extraction schema:
> 1. Document Type: e.g. Invoice, Contract, Internal Memo, Project Plan.
> 2. Primary Entities: list the main organizations or individuals involved.
> 3. Key Dates: extract all relevant dates, e.g. effective date, expiration, due date, and format as YYYY-MM-DD.
> 4. Financial Value: if applicable, extract the total currency amount and type.
> 5. Summary: a one-sentence description of the document's purpose.
> 6. Sensitivity Level: classify as public, internal or confidential based upon the content.
>
> Rules:

- If a field is not found, return null.
- Generate valid JSON without any conversational text and citation markup.

Document Text:
[PASTE THE BUSINESS RECORD HERE]

This instructs the LLM exactly what you are looking for in the output, preventing it from rambling or adding irrelevant information. LLM's are great at handling different ways of expressing dates and turning them into a standard form. JSON is a popular format for metadata, but other formats such as JSON-LD can be used with the appropriate training. This may be needed when the metadata forms a graph with concepts, properties and relationships.

Chunks & Rules, Raggett [10], uses a simple syntax for chunks as a collection of properties, inspired by the widely cited cognitive architecture ACT-R [11], which was developed by John R. Anderson et al. Chunks are at a higher level than RDF and can be used for both concepts and relationships, along with their respective properties.

An open question is whether it is practical to create quasi-symbolic indexes for document stores that reflect the open texture of semantics for business and legal documents. One potential approach would be to use the Plausible Knowledge Notation, see Raggett [12]. This is designed to represent knowledge that is uncertain, context sensitive, imprecise, incomplete, inconsistent and subject to change. The approach was inspired by Collins [13], and replaces mathematical proof by rational arguments for and against a given premise, including the role of analogies.

Measuring the accuracy of metadata extraction is critical for business uses as errors can lead to compliance failures or lost information. The common statistical metrics include the following:

- *Precision:* the percentage of extracted metadata tags that were correct
- *Recall:* the percentage of metadata, the agent was able to find
- *F1-Score:* a weighted blend of precision and recall
- *Extraction Coverage:* the percentage of documents, the agent was able to extract any metadata

Data quality dimensions include the percentage of required fields that are extracted, whether the data is in the requested format, and consistency in extracting the same entity across different records.

Some metrics specific to using LLMs include: semantic similarity compared to human evaluators, what percentage of cases require a human to step in and fix a mistake, and using a more powerful LLM to review the output of a smaller, faster model. Some business metrics include the frequency with which a document is put into the wrong bucket, e.g. a tax form mislabelled as an invoice, and the speed up for retrieving records attributable to LLM metadata enrichment.

Some challenges and technical considerations:

- *Data Privacy and Security*: sensitive data requires strict attention to data privacy and security. Models must be run in secure environments.
- *Model Bias*: minimising bias requires careful attention to the training data.
- *Scalability*: this is challenging when combined with the need to deploy models in secure environments.
- *Integration with Existing Workflows*: this can be eased by adopting the same metadata standards and protocols.

In conclusion, LLM based metadata extraction can be implemented as a collection of agents for specific vocabularies and used to speed selective processing of business records as part of business workflows. Human oversight is needed to ensure privacy, security and accuracy. AI doesn't replace the need for work on metadata ontologies, although it can be used to support people working on them, e.g. checking that proposed changes are consistent with a large collection of use cases.

5. Agentic Architecture

5.1. Neural AI Fails at Counting and Aggregation

Large language models have difficulties reliably executing tasks involving counting and aggregation. Structured Databases (e.g., SQL): Are built on relational algebra, e.g. when you run SELECT COUNT(order_id) FROM Orders WHERE order_date >= '2026-01-01', the database engine executes a precise algorithm that guarantees the result is the exact, auditable, and repeatable number of orders.

AI-Native Text Repositories (RAG/LLMs): Do not count. They generate text. The system uses semantic search to find text snippets that mention "orders," "customers," and "this year." It feeds these snippets to the LLM. The LLM then tries to synthesize the answer based on its language training and the provided context.

Crucially: It can't perfectly synthesize a count from scattered, unstructured text, and it might "hallucinate" or misinterpret the total number based on the surrounding language. The result is probabilistic and highly unreliable for a business-critical metric.

A business system may need to count customer orders, iterating over potentially millions of transaction records. SQL databases are highly optimized (indexed, clustered) to perform this COUNT() operation in milliseconds. By contrast, vector databases would need to retrieve and re-read vector embeddings (mathematical representations) and the original text for every single order, which is computationally expensive and slow for pure counting. This makes it a poor choice for the job.

Counting and aggregation with soft criteria requires a combination of deterministic processing (scripts) with judgmental processing (AI). This points to the importance of hybrid systems!

5.2. AI Centric Hybrid Systems

An AI centric hybrid system has a number of characteristics: document repositories, e.g. reports, contracts, emails, policies, and audit trails; persistent annotations: natural language plus structured information that is updated by the AI agents; vector indexes for similarity-based retrieval; graph databases for the symbolic annotations, and service APIs for operating over the repositories and annotations, etc.

The system involves a combination of deterministic and judgmental processing, e.g. counting with soft criteria; agentic AI that implement the es using agent to agent, and agent to service APIs; agents can take on different roles, including orchestration; office staff that monitor, guide, and apply human judgment for difficult or high-risk cases, briefed by the AI agents. Tasks can be implemented using scripts plus neural AI based agents acting over document annotations, where scripts can invoke AI, and similarly, AI agents can invoke script-based services.

5.3. Semantic Orchestration

Semantic orchestration is the means by which agents can analyse a request and break it down into simpler tasks that are handed off to other agents. This exploits agents that are tuned to specific functions for reliable operation compared to prompting a large language model to handle the request in full.

An open question for future investigation is how to measure the effectiveness of orchestration as part of a testing strategy. Suitable metrics could be divided into categories such as decomposition and planning; adherence to the given business rules, precedents and regulations; resilience and handling of corner cases, including alignment with what human experts would choose, and self-correction rates; finally, collaboration and escalation, including the quality of the reasoning traces and briefings provided to staff and auditors.

Metric Category	Key Indicator	Target Outcome
Operational	Escalation Ratio	Minimum human intervention for routine tasks; strong alignment with human values; effective briefing materials and reasoning traces
Logic	Decomposition Accuracy	100% of critical sub-steps included in plans; avoidance of unnecessary steps
Compliance	Playbook Adherence	Zero violations of core business rules and regulations
Reliability	Self-Correction Rate	High detection and fixing of logical errors; effective use of clarification questions

5.4. Using LLMs as Agents

Agentic AI is an extension of Large Language Models to proactively plan, use external tools, and execute multi-step tasks to achieve specific goals without constant human prompting. Acharya [14] explores the foundational concepts, unique characteristics, and core methodologies driving the development of Agentic AI.

Agentic AI operates through a continuous reasoning loop: *Thought, Action, Observation*: The agent is given a high-level objective together with guidance on how to fulfil it. The agent's response may include instructions to invoke external service APIs. The results from the service are fed back into the prompt for the next step in the reasoning loop.

The agent prompt can be broken down into several parts: *Role*: Tell the agent the role it is expected to follow and any associated constraints, including relevant business rules and where to find working documents. *Thought*: Ask the agent to articulate its reasoning in an auditable trail before taking action to suppress hallucination, including citing relevant passages from working documents. *Action*: Describe which services to use and for what purposes. *Observation*: Ask the agent to review the output of its last action. The audit trails are key to testing and quality control. To avoid bloated prompts, short descriptions of services can be used by agents to pull in more detailed descriptions after they decide on which service to use.

5.5. Think Before You Act!

A useful approach is to ask the agent to think in a series of steps before taking action rather than responding immediately, e.g.

> *Before executing any tool, write out a numbered plan of the 3–5 logical steps required to satisfy this business rule.*

Review your proposed action against the 'Compliance Checklist.' Identify any potential rule violations. If a violation is found, rewrite the plan.

You can ask the agent to check the environment before every move:

Check the current status of the order. If the status is 'shipped,' abort the 'Cancel Order' logic and notify the manager

If an agent is tasked with approving a purchase order, it might proceed as follows:

Think: "I need to check the department budget."
Think: "The budget has €1,000 left. The request is for €1,200."
Think: "Business Rule #9 says I can't approve over-budget items, but Rule #12 says I can route them to the CFO for an exception."
Act: Routes the request to the CFO instead of simply denying it.

6. Business Impact

6.1. Describing Business Processes for AI Agents

Traditional business process mapping (like BPMN) is often too rigid for AI. Agents don't need a step-by-step "if-then" script; they need a playbook of goals, constraints, and resources. *Declarative vs. Imperative:* Instead of "If X, click Y," descriptions must be declarative: "The goal is to resolve this refund within 24 hours while maintaining a 10% margin." *The "Context Layer":* Agents require access to "unwritten rules" or tribal knowledge. This includes: Business Values: Ranking priorities (e.g., "Customer retention is more valuable than strict policy adherence for long-term clients"). Precedents: A library of past successful "judgment calls" made by humans. *Resource Mapping:* A clear inventory of tools (APIs, databases, communication channels) the agent is authorized to use.

6.2. The Impact on Business IT Solutions

The move toward AI agents is breaking the "monolithic application" model: From UI-First to Agent-First: Traditional IT was built for humans to click buttons. Modern IT is moving toward headless systems optimized for agents to interact via APIs and "Agent-to-Agent" protocols (like MCP or Agent2Agent). After selecting a tool based upon a brief description, the agent can retrieve detailed information on how to use it.

Dynamic Orchestration: Rather than fixed workflows, IT solutions will act as orchestrators. They will dynamically spawn agents to handle specific sub-tasks (e.g., one agent for data extraction, another for compliance verification) and retire them once the goal is met.

The Rise of the "Digital Twin" of Work: Every process is logged not just as a transaction, but as a "reasoning trace," allowing the system to audit why a decision was made, not just what happened. This can be used to review the reasoning process and to support learning through access to the precedents set by decisions made in a similar context. The agent is prompted to summarise its reasoning for the service that maintains the audit trails. This should include noting which business rules were applied as well as citing short pieces from the documents used to support decisions. Automated checks on these should be applied to verify the citations, and supplemented by human spot checks on the quality of the reasoning. Note that business rules are passed to agents in the prompts rather than training agents to remember the rules, which would risk hallucination of non-existent rules.

6.3. Corner Cases are Inevitable

Corner cases are the bane of traditional IT because they require human-level reasoning to resolve ambiguity. AI-based systems handle these differently:

Feature	Traditional IT	Agent-Based Agentic AI
Logic	Boolean (true/false)	Probabilistic & reasoning based
Novel Inputs	Error/crash	Decomposes task: searches for similar precedents
Ambiguity	Stops and waits	Asks clarifying questions or makes best-guess based upon values
Adaptability	Requires code update	Learns from the outcome of the corner case (new precedents)

AI agents can navigate "grey areas" by referencing Business Ontologies—structured maps of how different parts of your business relate—allowing them to find creative workarounds that a hard-coded script never could.

6.4. Intent Based Interpretation of Queries

Business managers will often ask questions that should be answered in terms of the likely intent, e.g. consider the queries: *"How many of our loans are at risk of defaulting?"* and *"How many of our local trains are at significant risk of failing in the next 24 hours?"* In the first query, "risk of defaulting" is an imprecise term, and an effective response could provide estimates for the number of defaulted loans for different levels of risk, e.g. 20% vs 80%. In the second query, "risk of failing" is very open ended. e.g. what kinds of failures are critical and which are not, e.g. an inability to close the train doors vs a failure of a train toilet. An office worker tasked with the original question would be expected to explore the risks and the actions that could be taken in mitigation, including a review of maintenance schedules. An AI system should be prompted to consider how to answer open ended queries intelligently based upon the likely intent of the question.

6.5. Intelligent Escalation and Human Oversight

When an agent hits a "tricky case" (high risk or high ambiguity), it doesn't just "fail." It provides an Escalation Brief to a human partner. *Background Analysis:* The agent presents the human with a summary of the case, the conflicting rules involved, and a search of internal precedents (e.g., "In 2023, we made an exception for a similar client under these conditions"). *Business Value Alignment:* It suggests 2-3 possible actions and ranks them based on company values (e.g., "Option A maximizes short-term revenue, but Option B protects the brand's reputation for fairness"). *Human-in-the-loop:* The human acts as the final "judgment engine," and their decision is fed back into the agent's memory, effectively "training" the system for the next time that corner case occurs.

6.6. Workflow Orchestration and Adaptation

Business workflow orchestration is the process of coordinating multiple automated tasks, systems, and human actors into a single, cohesive end-to-end process. While automation focuses on making a single task run without human help (like sending an automated email), orchestration is the "conductor" that ensures all those automated tasks happen in the right order, at the right time, and across different departments.

Using AI for orchestration offers considerable flexibility compared to traditional approaches, powering improved customer satisfaction and agility in responding to changing market tastes and conditions.

On the other hand, most business workflows are proprietary, making it challenging to gather sufficient data for fine tuning AI models to manage workflow orchestration. A work around is to use generative AI to create synthetic data as variations of manually produced examples. Fine tuning and alignment with business values can be further enhanced by training the orchestrator to generate the prompts for the AI agents that implement particular tasks. Businesses need an effective testing strategy that exploits the skills of experienced staff, and ensures on-going quality control. This includes the preparation and ongoing maintenance of test data, and its reapplication when any changes are made to the business work flow and rules.

Jiang [15] provides a systematic overview of adaptation of Agentic AI in respect to both the AI models and the tools they invoke.

6.7. Opportunities for standards and open source

Agentic AI can use emerging standards for industry protocols such as MCP and A2A. There are opportunities for standards in respect to document annotations. Annotations for different kinds of documents can apply to specific parts of a document, e.g. a numbered clause in a business contract. Annotations can be arranged as graph statements in standardised vocabularies. There are opportunities for simpler, more accessible notations that are easier for human staff to review. There are further opportunities for standards in respect to security, access control, compliance, and auditability.

There are plenty of opportunities for open-source implementations, e.g. AI models for different business process roles, suitable for the edge or cloud as appropriate. The running costs for large models can be reduced through techniques such as: mixture of experts, attention-based short-term memory in lieu of large context windows; learnable-long-term memory; model distillation involving reduced precision and weight pruning, and using a large model as a teacher for a smaller one.

Vector databases provide a means to offload parts of the model to external databases. Open source can be further used for common scripts and services, and libraries for working with different document formats.

Of particular note are the frameworks for orchestrating agentic AI. These are mostly open source, though several follow a "Core Open Source + Proprietary Platform" model. Some example frameworks include LangGraph, Microsoft AutoGen, LlamaIndex and Microsoft Semantic Kernel.

7. Conclusions

Neural AI excels at understanding context, semantics, and the intrinsic vagueness of natural language, but like human memory, is weak in respect to counting and aggregation. Humans counter this with written records and filing systems with categories for easy retrieval. Neural systems can likewise use persistent records with structured annotations that enable precise deterministic processing without hallucinations. AI can invoke services, likewise, services can invoke AI, for flexible handling of open-ended tasks that are based upon understanding the intent of a query rather than interpreting it literally. Audit trails are kept on how and why decisions were made rather than just recording the results. This further enables hybrid AI systems to learn from experience through access to precedents. AI agents can be tuned for different roles, akin to specialisation of human workers, and orchestrated to implement complex work flows.

Natural language is open textured with meanings that are necessarily imprecise and context sensitive. This gives business processes and legal documents the flexibility to cope with new situations using human judgement and precedents. Previous work has used

Graph RAG to ground reasoning on facts expressed as knowledge graphs, which limits their utility compared to the richness of natural language. The approach presented in this article combines AI for context sensitive deep semantics with conventional services for deterministic processing. You can think of this as semantic orchestration involving AI agents and human staff, where humans are in the loop for difficult or high-risk cases, and for guiding open ended analyses. This reverses the de-skilling of office workers we saw in the 19th and 20th centuries, bringing back the flexibility of human judgment to offer greater flexibility and faster adaptation to changes in business requirements along with improved productivity.

Agentic AI is an active area for extensive research and development, and it is beyond the scope of this article to provide a comprehensive survey. This article instead seeks to provide a forward-looking perspective on applying AI to orchestration and reasoning for business workflows in collaboration with experienced human staff, and flexibly exploiting business rules and precedents expressed in natural language rather than mathematically precise deterministic instructions. This provides an opportunity for businesses to boost productivity and better handle unforeseen cases.

Funding: This research received no external funding.

Acknowledgments: During the preparation of this manuscript/study, the author acknowledges limited use of Gemini 3 Flash to assist with drafting the author's ideas. The author has reviewed and edited the output, and takes full responsibility for the content of this publication.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LLM	Large Language Model
RAG	Retrieval Augmented Generation
RDF	Resource Description Language

References

1. **James Beniger,** ***The Control Revolution: Technological and Economic Origins of the Information Society*** (1986), Harvard University Press. Beniger argues that the "Information Age" actually began in the mid-19th century. He describes how the speed of steam power created a "crisis of control" that could only be solved by new bureaucratic technologies: the telegraph, the typewriter, the vertical filing cabinet, and standardized forms.
2. **JoAnne Yates,** ***Control through Communication: The Rise of System in American Management*** (1989), The John Hopkins Press. This is the definitive history of "paperwork." Yates explains how firms transitioned from oral traditions to internal memos and reports. She provides a fascinating look at the evolution of physical infrastructure, such as how the invention of the vertical file (around 1893) revolutionized data retrieval.
3. **William Henry Leffingwell,** ***Scientific Office Management*** (1917). Inspired by Frederick W. Taylor's earlier work on factory management, this book extends the Taylor system to office work, featuring time & motion studies, task and bonus plans, and arranging the office floor plan to expedite routing of working documents.
4. **Harry Braverman** (1974), ***Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century***, Monthly Review Press.
5. **Harvard Business Review** (November 2025), **AI Agents aren't ready for consumer facing work, but they can excel at internal processes**. An account of practical deployment experience showing the importance of breaking tasks into small simpler ones that can be reliably carried out by individual agents to boost the productivity of human staff by taking over mundane time consuming work. See: https://hbr.org/2025/11/ai-agents-arent-ready-for-consumer-facing-work-but-they-can-excel-at-internal-processes

6. **H. L. A. Hart,** ***The Concept of Law*** (1961), Oxford University Press. Hart describes law as a system of social rules grounded in social acceptance. Every legal rule has a core of settled meaning where the rule clearly applies, and a penumbra of uncertainty where the rule's application is debatable due to the deliberate vagueness of words. Judges exercise discretion when a case falls within the penumbra. Hart refers to this as the open texture of law and describes it as a necessary and desirable feature of a healthy legal system to adapt to unforeseen combinations of circumstances.
7. **Patrick Lewis, Ethan Perez, et al.** (2020), ***Retrieval-Augmented Generation for Knowledge - Intensive NLP Tasks***, Facebook AI Research. Introduces RAG as a means to inject search results into the LLM prompt using an external vector database.
8. **Darren Edge, Ha Trinh, et al.** (April 2024), ***From Local to Global: A Graph RAG Approach to Query-Focused Summarization***, Microsoft Research. See: https://arxiv.org/abs/2404.16130. Introduces GraphRAG as a way to address weaknesses of RAG in respect to global questions over a dataset. They derive a knowledge graph from the source documents then generate summaries for groups of closely related entities. The summaries are used to aggregate a final response to the user's query.
9. **Qinggang Zhang, Shengyuan Chen, et al.** (January 2025), ***A Survey of Graph Retrieval-Augmented Generation for Customized Large Language Models***, See: https://arxiv.org/abs/2501.13958. Systematically analyzes the technical foundations of GraphRAG, and examines current implementations across various professional domains, identifying key technical challenges and promising research directions.
10. **Dave Raggett** (2025), ***Chunks and Rules Specification***, see: https://w3c.github.io/cogai/. This defines a cognitively inspired database model using chunks as collections of properties and rules that operate on them in conjunction with highly scalable graph algorithms.
11. **John R. Anderson et al.** (2004), ***An Integrated Theory of the Mind***. Psychological Review, 111(4), 1036–1060. https://doi.org/10.1037/0033-295X.111.4.1036. Describes an extended version of Adaptive control of thought-rational (ACT-R) introduced in 1998. It provides a comprehensive overview of the architecture's evolution and its integration of multiple modules to produce coherent cognition.
12. **Dave Raggett** (2025), ***The Plausible Knowledge Notation (PKN)***, see: https://w3c.github.io/cogai/pkn.html. This defines PKN as a lightweight syntax for expressing imperfect knowledge, i.e. knowledge that is uncertain, context sensitive, imprecise, incomplete, inconsistent and likely to change.
13. **Allan Collins** and **Ryszard S. Michalski** (1989), ***The Logic of Plausible Reasoning: A Core Theory***. Cognitive Science, Vol. 13, Issue 1, pp. 1–49. Introduces a system for representing human inference patterns, and the role of parameters—such as typicality, similarity, and conditional likelihood—that determine the strength or certainty of a plausible conclusion.
14. **Deepak Bhaskar Acharya et al.** (2025), ***Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey***, IEEE Access, vol. 13, pp. 18912-18936, 2025, doi: 10.1109/ACCESS.2025.3532853. This defines Agentic AI as a paradigm shift from reactive prompting to autonomous systems capable of proactive goal decomposition, self-correction, and long-term orchestration within complex environments.
15. **Pengcheng Jiang, Jiacheng Lin, et al.** (December 2025), **Adaptation of Agentic AI**. This provides a systematic framework for categorizing agentic AI adaptation strategies into agent-centric and tool-centric approaches, offering a roadmap to help researchers and practitioners navigate design trade-offs and build more reliable, complex systems. See: https://arxiv.org/abs/2512.16301v2