Bridging the gap between language and deterministic processing

Dave Raggett (W3C/ERCIM)

W3C staff contact for Web of Things

Introduction

- AI is rapidly improving: reasoning, searching, summarizing and coding, with Agentic AI capable of handling increasingly long chains of tasks
- But organizations lack a clear enough understanding of how their work gets done, limiting the applicability of AI
- The knowledge exists, but it is held by people, and not in systems
- Institutional knowledge is mostly tacit, involving local judgment, habits, honed intuition, and undocumented work arounds
- Return on investment will depend on reshaping how organisations work, just as in past industrial transitions
- This talk will discuss opportunities for combining Agentic AI with conventional symbolic approaches for indexing and processing information, along with semantic orchestration and semantic interoperability
- AI is well suited for reasoning with the open texture of language, in combination with robust deterministic processing for structured information
- AI can help with the integration of systems using incompatible vocabularies, using the context to dynamically resolve the intended meaning
- Human oversight based upon audit trails that encourage institutional learning as new precedents are established

Offices before Computers

- Rows of people working at their desks or filing system
- Processing orders, preparing routine business reports
- Business rules + human judgment + written records
- Filing systems with neat categories for easy lookup
- Managers can request work on special reports, e.g. on business challenges and opportunities

Inspiration for AI powered businesses!

The Role of Language

- Legal documents and business rules are expressed in language, using terms that are intentionally vague
- This allows judges, regulators, and business partners to apply them reasonably to unforeseen circumstances. Mathematical terms, conversely, are designed to eliminate all ambiguity
- Large Language Models (LLMs) are very effective at dealing with human language, having been trained to embody complex statistical relationships
- Traditional IT hates ambiguity and suffers from unforeseen edge cases, but on the plus side, is very good at deterministically handling large amounts of data

Ambiguity and Vagueness are Necessary

- **The Law**: 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.
- **Example**: 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.

LLMs Fail at Counting and Aggregation

Deterministic Challenge

- Structured Databases (e.g., SQL): Are built on relational algebra, e.g. when you run

  ```
  SELECT COUNT(order_id) FROM Orders WHERE order_date >= 'last_week'
  ```

  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 "last week."
  - 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.

Performance Challenge

- Counting customer orders involves iterating over potentially millions of transaction records.
- A SQL database is highly optimized (indexed, clustered) to perform this COUNT() operation in milliseconds.
- A Vector Database would have 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. It simply is not the right tool 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!

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).

Emergence of database platforms that natively integrate:

- Relational/Transactional Data (for speed and consistency).
- Graph Data (for complex relationships).
- Vector Data (for semantic search and unstructured content).
- In this environment, language models will be an embedded, fundamental component that enables everyone to interact with the data simply by stating their goals and constraints.

Language as a Query Translator

The most immediate change is the role of Neural AI as a Universal Data Interface and Translator.

Neural AI excels at interpreting human intent, even with ambiguity, and translating it into precise, formal query languages.

- **User Input**: Natural Language (e.g., "Find all customers in Germany who spent over 500€ last month on product X, and their associated sales agent.")
- **Agent Action**: Neural AI translates this request into a structured query (e.g., SQL, Cypher for a graph, or a SPARQL query for RDF).
  - AI further can help with optimising database performance
- **Database Action**: The database executes the formal, optimized query and returns the exact data.
- **Agent Action**: Neural AI converts the result set 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.

Graph Databases for Structured Facts

Graph databases (GD) provide the structured truth that Agentic AI needs:

- **Grounding (RAG)**: Using techniques like Retrieval-Augmented Generation (RAG), the Neural AI uses the GD as a reliable source of truth. When asked, "What is the relationship between product X and supplier Y?" the Neural AI queries the 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."). GDs 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.

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.

Mitigating LLM Hallucination

- LLMs are prone to hallucination – where it talks confidently about things it has imagined
- This can be mitigated through careful prompting
- For instance, listing the metadata to be extracted rather than leaving this undefined,
- Asking the agent to generate an audit trail of its reasoning, including relevant extracts as evidence, and
- Having agents check the work of other agents

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:

Document Type: e.g. Invoice, Contract, Internal Memo, Project Plan.

Primary Entities: list the main organizations or individuals involved.

Key Dates: extract all relevant dates, e.g. effective date, expiration, due date, and format as YYYY-MM-DD.

Financial Value: if applicable, extract the total currency amount and type.

Summary: a one-sentence description of the document's purpose.

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]

Using LLMs as Agents

- Agentic AI operates through a continuous **Reasoning Loop**
  - *Thought, Action, Observation*
- Agent is given a high-level objective together with guidance on how to fulfil it
- Agent's response can include instructions to invoke a service
- Service's response is fed back into the prompt for the next step in the reasoning loop

The agent prompt includes:

- *Role*: Tell the agent the role it is expected to follow and any associated constraints
- *Thought*: Ask the agent to articulate its reasoning before taking action
- *Action*: Describe which services to use and for what purposes
- *Observation*: Ask agent to review the output of its last action

Think Before You Act!

Key to providing rich audit trails for oversight of business workflows involving Agentic AI

Ask the agent to think in a series of steps before taking action:

1. *Before executing any tool, write out a numbered plan of the 3–5 logical steps required to satisfy this business rule*
2. *Review your proposed action against the 'Compliance Checklist.' Identify any potential rule violations. If a violation is found, rewrite the plan*

Ask agent to check the environment before every move:

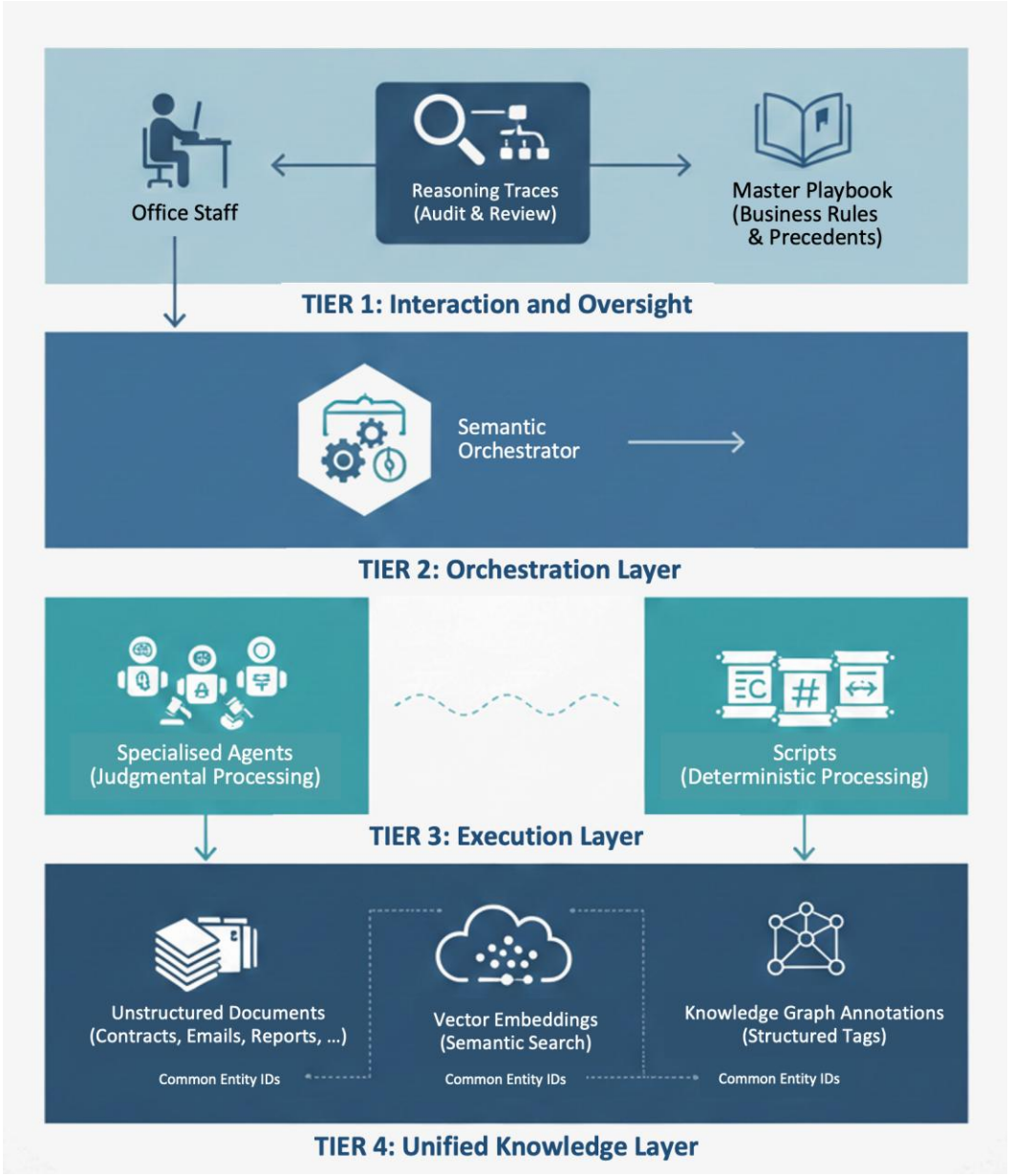
Check the current status of the order. If the status is 'shipped,' abort the 'Cancel Order' logic and notify the manager

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

1. *Think*: "I need to check the department budget."
2. *Think*: "The budget has €1,000 left. The request is for €1,200."
3. *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."
4. *Act*: Routes the request to the CFO instead of simply denying it

Semantic Orchestration

- Semantic Orchestration is the means for AI agents to route tasks to specialized AI agents and human workers in an auditable workflow
- Each agent has customized prompts and scripts
- This includes agents that extract metadata from a variety of resource formats according to the business context
- Exhaustive testing requires a comprehensive test suite that should be updated as new use cases come to light
- AI supports but does not replace human staff, as such businesses need to nurture their staff and institutional knowledge



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).
- **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.

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	AI-Based Agentic IT
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 on values
Adaptability	Requires code update	Learns from the outcome of the corner case

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.

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.

Opportunities

Standards

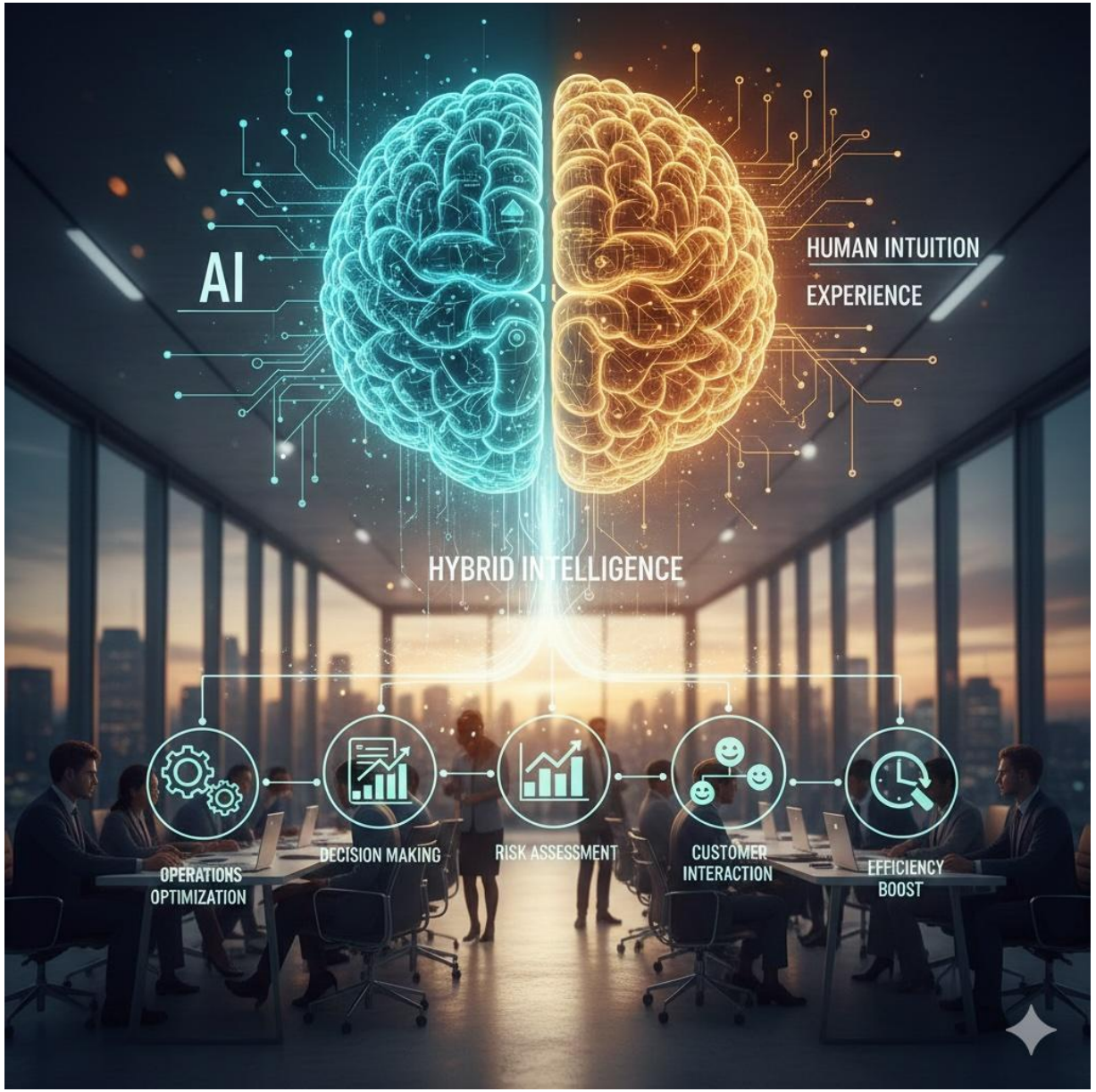
- Protocols such as MCP and A2A
- Annotations for different kinds of documents
  - For parts of a document
  - Vocabularies for structured information
- Simpler more accessible notations
- In respect to security, access control, compliance, auditability

Open Source

- AI models for different business process roles, suitable for edge/cloud as appropriate
  - Mixture of experts
  - Short term memory
  - Model distillation
- Vector databases
- Common scripts and services
- Libraries for working with different document formats

Questions and Comments?

We should have time for a few questions right now. See the following slide for some suggestions for the panel discussion at the end of this session.



Some Potential Questions for Discussion*

Natural language and AI models operate probabilistically, while semantic interoperability relies on deterministic representations such as ontologies and formal models

- How can we effectively bridge these two paradigms so that AI systems can understand and reliably operate on semantically described digital twins?
- Can LLMs generate trustworthy semantic structures, or should they only assist humans in ontology creation?

Ontology development and maintenance remain complex and resource-intensive.

- How can AI (particularly large language models) support ontology engineering tasks such as: ontology generation, ontology alignment, schema mapping, semantic documentation
- How can AI improve the ontology engineering process without introducing inconsistencies or semantic drift?

Digital twins increasingly span multiple systems, organizations, and lifecycle stages.

- What governance or architectural approaches are needed to maintain semantic interoperability across large digital twin ecosystems?
- Do Digital Twins ensure that data models and data can continuously evolve?

Streaming knowledge graphs are emerging as a backbone for dynamic digital twins.

- How do we ensure that the semantics of these knowledge graphs remain consistent and interoperable?
- Are knowledge graphs resilient models when data is ingested in real time from heterogeneous sources?

Even with well-defined ontologies, accessing and integrating data from digital twin is can be difficult by its nature of specialisation

- What role do semantic query languages and graph querying techniques play in enabling interoperability?
- What about cross-digital twins analytics, should query languages evolve to support AI-driven interaction with digital twins?

Semantic models provide structure and meaning, while AI provides pattern recognition and prediction.

- How can these two approaches complement each other to enable more intelligent digital twins that can reason, learn, and adapt?

Standards are essential for interoperability, but the ecosystem of ontologies, vocabularies, and data models remains fragmented.

- What should be a priority for the community: harmonizing existing ontologies or creating shared domain vocabularies?
- What would resolve the problems of data interoperability in mid-term building tools for semantic alignment or developing interoperability testing frameworks?

* With thanks to Martin Serrano