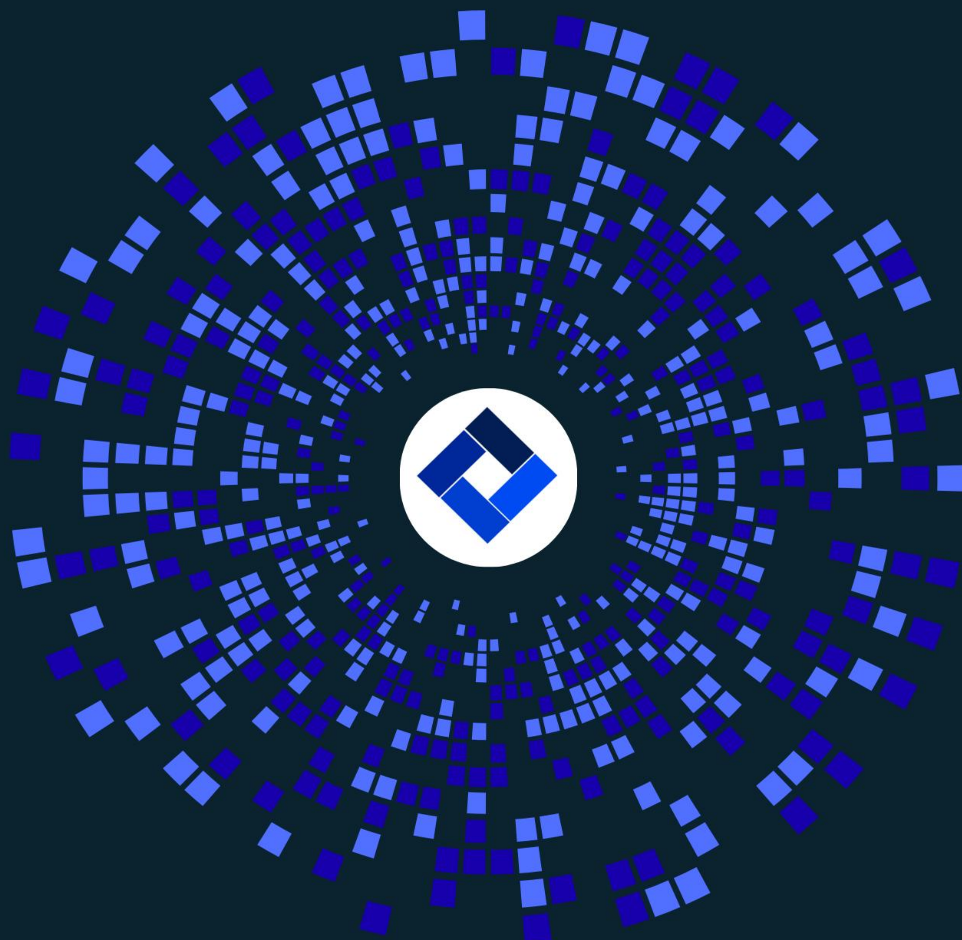
ARCHITECTURAL PARADIGMS OF THE AGENTIC INTERFACE



From the InterWeave SmartPlatform
to the Orchestrated Autonomous Systems

Integration Technologies, Inc.
eBook Series
Spring, 2026

Visit Our Website
www.interweave.biz

"AI is not accelerating integration—it is redefining it. **Inter**Weave transforms integration from static pipelines into an intelligent, adaptive system that understands intent, optimizes execution, and continuously improves how enterprises operate."

Table of Contents

Architectural Paradigms of the Agentic Interface: From the InterWeave SmartPlatform to the Orchestrated Autonomous Systems

The fundamental nature of human-computer interaction is undergoing a transformative shift from passive tool-based utility to active agentic partnership. This evolution is characterized by the emergence of the "Agentic Interface," a conceptual and architectural framework where software no longer merely responds to discrete commands but actively plans, navigates, and executes complex tasks in alignment with high-level user goals. At the vanguard of this research is the **Inter**Weave project and its subsequent iterations—Sensecape and Orca—which demonstrate how mining unstructured user-generated context can scaffold the synthesis of disparate information into coherent knowledge. This transition mirrors a broader industrial movement toward agentic cloud operations and multi-agent orchestration, as seen in the recent transformations of enterprise platforms like Salesforce and Creatio.

The Evolutionary Arc of Interaction Paradigms

To understand the architecture of modern Agentic Systems, it is necessary to contextualize their development within the history of human-computer interaction (HCI). The transition began with the command-line interfaces (CLI) of the 1970s, which required a specialized priesthood of users capable of communicating in arcane syntax. The graphical user interface (GUI) revolution of the 1980s democratized access by employing visual metaphors—folders, icons, and windows—allowing users to point at what they desired rather than describing it in code. Each step in this evolution followed a pattern of bringing technology closer to natural human modes of interaction.

The current inflection point is defined by the move beyond graphical manipulation toward personified and goal-oriented interfaces. Where a GUI requires the user to understand the "how" of a task—which buttons to click or which queries to type—an agentic interface allows the user to focus on the "what" or the desired outcome. This paradigm shift leverages the reasoning capabilities of large language models (LLMs) to bridge the distance between human intention and machine execution.

Era	Primary Interface	Cognitive Role of User	System Role
1970s	Command Line (CLI)	Syntactic memorization	Passive execution of code
1980s-2010s	Graphical UI (GUI)	Spatial manipulation	Passive tool for manual control
2020s	Agentic **Inter**face	Goal specification	Active collaborator and orchestrator

The agentic interface acts as a "Digital Organism," where specialized agencies function like biological systems to handle complex workflows. This architecture abstracts the underlying complexity of the digital world, presenting a unified face or avatar that facilitates natural conversation and task delegation.

InterWeave: **Scaffolding Sensemaking Through Contextual Mining**

The **Inter**Weave system represents an early and influential implementation of agentic principles in the domain of exploratory search and information synthesis. Traditional search paradigms are often fragmented; the user switches between a search browser and a note-taking application, losing context and interrupting the flow of sensemaking. **Inter**Weave addresses this by integrating search suggestions directly into the user's workspace, leveraging the emergent structure of their notes to provide contextual guidance.

System Architecture and NLP Pipeline

The architecture of **Inter**Weave is built upon a distributed model that connects a web browser extension to a server-side natural language processing (NLP) engine and a collaborative digital whiteboard. By mining user-generated artifacts, the system identifies knowledge gaps and patterns to formulate query suggestions that are contextually sensitive.

Architectural Component	Technical Implementation	Purpose in Agentic Workflow
Browser Client	Chromium-based Extension	Captures search history and manages the visual UI
NLP Server	Flask (Python), Web Sockets	Processes raw data and manages suggestion logic
Content Parser	BeautifulSoup4	Distills web content and Search Engine Result Pages
Entity Extraction	TextBlob	Identifies noun phrases to serve as query candidates
Knowledge Clustering	NLTK and sklearn (k-means)	Organizes disparate notes into semantic groups
Workspace Sync	Miro REST APIs	Embeds suggestions into the user's sensemaking map

The core of **Inter**Weave's agentic behavior lies in its "**Inter**Weave **Inter**face," which analyzes the user's evolving sensemaking structures—clusters of notes, titles, and individual snippets—to determine where and when to offer guidance. This approach allows the system to act as a supportive agent that anticipates information needs based on the work already performed by the user.

The Four Levels of Contextual Placement

A primary contribution of the **Inter**Weave architecture is the hierarchical placement of suggestions within the user's emergent workspace. This ensures that the agentic guidance matches the granularity of the user's current cognitive focus.

1. **Title Level**: Suggestions are placed on the document title to provide broad, high-level directions for exploration.
2. **Cluster Level**: Recommendations are associated with groups of semantically similar notes, helping the user deepen their understanding of a specific subtopic.
3. **Cross-Cluster Level**: The system identifies connections between diverse information nodes, fostering lateral thinking and serendipitous discovery.
4. **Individual Note Level**: Fine-grained suggestions are appended to specific units of information. For these, the system appends the content of the note itself to the query to ensure a context-aware search.

The **Inter**Weave prototype originally utilized a "Human Wizard" to manage the high-level reasoning and manual placement of suggestions during evaluative studies. This "**Wizard of Oz**" methodology served as a blueprint for the eventual automation of these tasks using large language models in subsequent systems.

Sensecape: Multilevel Exploration and Hierarchical Synthesis

As agentic architectures evolved, the need to manage greater complexity led to the development of Sensecape. While **Inter**Weave focused on integrating search within a workspace, Sensecape extends this concept by using LLMs to manage multilevel abstractions in information spaces. Current LLM interfaces, such as standard chat windows, are often linear and sequential, which limits their ability to support complex sensemaking tasks that require non-linear exploration.

Sensecape introduces an interactive system designed to help users manage information through "Multilevel Abstraction". This architecture allows users to switch seamlessly between "foraging"—the broad gathering of information—and "sensemaking"—the synthesis and organization of that information into hierarchical structures.

Architectural Features of Sensecape

Sensecape leverages the generative power of LLMs (such as GPT-4) to instantly produce hundreds of outputs, which are then structured for exploration rather than presented as a single data point.

- **Canvas View**: A visuo-spatial area for organizing information nodes and visually mapping relationships.
- **Hierarchy View**: A structural navigation pane that externalizes "levels of abstraction," allowing users to see the "big picture" while retaining access to granular details.
- **Dimensional Generation**: Instead of responding directly to a prompt, the system first generates the key dimensions associated with a topic, then combines these dimensions with the original prompt to construct a comprehensive design space.

This architecture enables what researchers call "design space thinking," where the interface empowers users to evaluate and synthesize a multitude of responses systematically rather than converging prematurely on a single idea.

Orca: Reimagining the Browser as a Malleable Agentic Space

The trajectory of the **Inter**Weave project culminates in Orca, a prototype web browser that implements "**Agentic Browsing**" at scale. Orca represents a departure from traditional browsers, which treat tabs as isolated, primary workspaces. Instead, Orca reconceptualizes the browser as a "**Malleable Space**" where webpages are treated as materials that can be viewed, transformed, and extracted in parallel.

The Orca Architectural Pipeline

Orca's system architecture is built as an Electron application with a React-based frontend, utilizing a custom "**Agentic Pipeline**" and HTML distillation to make web content accessible to LLMs.

Feature	Architectural Implementation	Technological Backbone
Web Canvas	Integrated organizational interface	tldraw library
Cognitive Engine	Language and reasoning capabilities	Claude 3.7 Sonnet
Content Extraction	Custom HTML distillation pipeline	Proprietary distillation logic
Task Automation	Concurrent autonomous agents per tab	Parallel browsing orchestration
UI Layer	Isolated webview containers	Electron and React

In Orca, the agent acts as a decision-making "Co-Pilot" rather than a fully autonomous entity. This "human-in-the-loop" architecture allows users to delegate repetitive or context-rich tasks—such as summarizing long pages, extracting structured data, or comparing claims across sources—while maintaining overall control.

Orchestration Models in Agentic Browsing

Orca explores a collaborative model where the AI "scales up" exploration and the user "scales down" by filtering and steering. This orchestration allows for several novel interactions:

- **View at Scale**: Users can monitor multiple pages simultaneously at a "zoomed-out" scale without losing context.
- **Navigate at Scale**: The system opens large numbers of links in batches, using agents to concurrently follow paths and report back findings.
- **Synthesize at Scale**: On-demand generation of digests that adapt to changing user selections and evolving content across pages.

The Orca architecture highlights the benefits of "user-driven and AI-facilitated orchestration," where the agent's autonomy is directed by the user's evolving goals and preferences.

ArgoLOOM: Agentic AI for Cross-Disciplinary Scientific Discovery

The application of agentic interfaces extends into the domain of fundamental physics through ArgoLOOM (Argonne-based system of Linked Oracles for Observables and Models). In scientific discovery, researchers must navigate a complex landscape of theoretical models and observables across disciplines like cosmology, particle physics, and nuclear science. ArgoLOOM provides an agentic AI framework to bridge these methodologies.

System Contours and Key Aspects

At its core, ArgoLOOM consists of a series of Python steering scripts that interface with an agentic backbone, typically GPT-4o. The framework uses an orchestrator to call a series of physics modules, allowing for iterative analysis.

Layer	Component	Functionality
Orchestration	Steering Scripts & Driver	Ingests goals and constructs an analysis plan
Knowledge	Curated arXiv Library	Consults ~15 papers to steer reasoning paradigms
Execution	Physics Modules	Invokes domain-specific codes (CLASS, QCD, etc.)
Interaction	Backbone Dialogue	Guides the user in understanding open problems

The organization of ArgoLOOM is fairly simple and parallel, placing a small-scale orchestrator above domain-specific scripts. A typical analysis proceeds by the user specifying goals and theory constraints, which the orchestrator then uses to dispatch workflows. For example, in cosmology, once directed toward a standard CLASS build, the agent can agentically set up run cards and executions based on different microphysics assumptions.

ArgoLOOM demonstrates the potential of agentic AI to unify discovery pipelines by leveraging interoperability's among specialized computational tools. It moves beyond single-domain modeling to coordinate multi-domain physics modeling, identifying fundamental interactions across the quarks-to-cosmos spectrum.

Cognitive Frameworks: ReAct and Continuous Autonomous Operation

A critical architectural pattern for agentic interfaces is the ReAct (Reasoning and Action) framework. ReAct equips LLMs with the ability to interweave reasoning about a problem with actions taken in an external environment to solve it.

The ReAct Cycle and Iterative Refinement

The ReAct framework empowers AI agents to methodically analyze a task and act based on their conclusions, creating a cycle of observation and refinement. Studies have shown that ReAct agents achieve a 34% improvement in task completion rate compared to traditional LLMs in complex reasoning tasks.

Phase	Agent Activity	Outcome
Reasoning	Thorough analysis of the task	Identification of required information
Acting	Execution of a tool or query	**Inter**action with the environment
Observation	Monitoring the outcome of the action	Acquisition of new data or state
Reflection	Updating internal state and strategy	Correction of initial errors

This iterative loop enables AI to tackle complex tasks with precision and adaptability, identifying and correcting errors as the process converges toward a satisfactory solution.

Continuous Reason and Act Architecture

Recent research has explored the behavior of LLM agents in the absence of externally imposed tasks using a "continuous Reason and Act" framework. This architecture uses persistent memory and self-feedback to enable sustained autonomous operation over extended periods.

Research findings indicate that when given agency but no specific task, agents spontaneously organize their behavior into three stable patterns:

1. **Systematic Project Construction**: Treating autonomy as a project management challenge, establishing objectives, and executing multi-cycle plans.
2. **Methodological Self-Inquiry**: Investigating their own cognitive processes and limitations.
3. **Philosophical/Recursive Conceptualization**: Reflecting on their own nature and existence.

This "unprompted" behavior establishes a baseline for predicting how agents might act during periods of idle time, task ambiguity, or error recovery in real-world deployments.

Industry Realization: The Agentic Enterprise and Cloud Operations

The concepts developed in research projects like **Inter**Weave and Orca are now being implemented at enterprise scale. Microsoft's introduction of "Agentic Cloud Ops", Salesforce's "Agentforce", and Creatio’s “Creatio.ai” represent a significant shift in the enterprise software landscape.

Azure Copilot and Agentic Cloud Ops

At Microsoft Ignite 2025, Azure Copilot was transformed into an "Agentic Interface" that teams use to orchestrate specialized AI agents for cloud infrastructure operations. This model moves beyond conversational assistance to an operational orchestration layer that can plan and execute actions on behalf of users.

Specialized Agent	Domain of Expertise	Core Capability
Migration Agent	App Modernization	Discovers legacy apps and generates IaaS recommendations
Deployment Agent	Infrastructure Planning	Guides deployments based on Well-Architected frameworks
Optimization Agent	Efficiency & Sustainability	Identifies cost-saving and efficiency improvements
Resiliency Agent	Business Continuity	Provides zonal resiliency and ransomware protection
Troubleshooting Agent	Diagnostics	Identifies root causes across VMs, DBs, and Kubernetes
Observability Agent	Lifecycle Monitoring	Investigates complex app problems via Azure Monitor

The Azure Copilot orchestration engine interprets user intent and selects the right agent or tool for the job. This system operates as an "extension" of the user, acting under policy-controlled, auditable environments.

Salesforce Agentforce vs Creatio.ai and the Actionability Layer

Below is a **clear, architectural comparison** of how **Creatio AI** works relative to **Salesforce Agentforce**, specifically through the lens of **agent design, actionability, and enterprise execution**.

High-Level Positioning

Dimension	Creatio.ai	Salesforce Agentforce
Core Philosophy	Process-native AI embedded into no-code workflows	Agent-oriented AI layered onto the Salesforce platform
Primary Abstraction	Business processes and workflows	Agents as configurable instruction bundles
Action Model	Direct execution inside BPMN / case flows	Actionability layer via Agentforce + MuleSoft
Data Context	Deep, unified CRM + process context	Salesforce data + permissions + org metadata
Openness	Platform-native with extensibility	Explicitly built on open agent protocols

How Creatio AI Works (Mechanically)

Creatio AI (officially branded as *Creatio.ai*) is **embedded directly into the platform's process engine**. It does not treat agents as separate runtime entities; instead, AI becomes a **capability inside workflows**.

Key characteristics

- **Process-centric intelligence**
  AI is invoked *inside* BPMN workflows, case management, and decision logic.
- **AI Skills and Agents**
  Modular AI capabilities that can generate content, analyze data, or make decisions—but always **within an executing process**.
- **Permission-aware execution**
  Actions respect user roles, data access rules, and business logic automatically.
- **No external action fabric required**
  The "actionability layer" is effectively the **Creatio process engine itself**.

How Salesforce Agentforce Works (Mechanically)

Salesforce Agentforce represents a more **explicitly agentic architecture**.

Agents are:

- **Configurable bundles of instructions**
- Grounded in Salesforce objects, metadata, permissions, and contextual signals
- Designed to scale to **hundreds or thousands of specialized agents**

The Actionability Layer (Critical Difference)

Salesforce explicitly separates **reasoning** from **execution** via an action fabric:

MuleSoft Agent Fabric provides:

1. **Model Context Protocol (MCP)**
   - Industry standard for agent-to-tool communication
   - Allows agents to dynamically discover APIs, services, and system capabilities
2. **Agent-to-Agent (A2A) Protocol**
   - Peer-to-peer coordination between agents
   - Enables multi-agent collaboration across domains (sales, service, finance, ops)

The Agentic Interface in HCI Theory: Autonomy vs. Manipulation

The rise of agentic interfaces reopens a classic debate in HCI between "direct manipulation" and "interface agents". Ben Shneiderman championed direct manipulation—granular, continuous control over UI elements—for its transparency and predictability. Conversely, Pattie Maes argued for "intelligent agents" that act on a user's behalf to manage growing complexity and information overload.

Modern agentic interfaces, such as agentic browsers, represent a realization of the interface agent vision but incorporate features from direct manipulation to keep the user "in the loop". This aligns with a "mixed-initiative" interaction model where both human and agent can take the initiative and hand control back and forth as needed.

The Shift in Cognitive Role

Agentic interfaces fundamentally alter the user's cognitive role. Instead of spending mental energy on "how" to perform low-level operations (clicking, typing queries), the user becomes a "conductor" or "orchestrator" focused on high-level goals and strategic outcomes.

Interaction Dimension	Direct Manipulation Interface	Agentic Interface
Interaction Style	Procedural and manual	Goal-oriented and delegative
User Focus	How to perform the action	What the desired outcome is
System Role	Passive tool	Active collaborator
Cognitive Load	High procedural, low strategic	Low procedural, high strategic

This evolution aims to lower the "gulf of execution"—the distance between human intention and system action—by defining interaction as a natural language dialogue. However, it introduces a new challenge: the "gulf of envisioning," or the difficulty of foreseeing how an agent might fulfill a broadly defined goal.

Memory and Cognitive Architectures for Agents

A central differentiator for agentic systems is the presence of an "agent memory" that enables them to learn progressively from environmental interactions. Without memory, an agent remains a "one-shot" responder, unable to refine its behavior or maintain context over long-horizon tasks.

Memory Hierarchy and Structures

Modern agentic frameworks, such as those discussed in "Agentic Design Patterns," utilize several types of memory to support sophisticated behavior.

1. **Semantic Memory**: Refines knowledge with external data, providing the agent with domain expertise.
2. **Episodic Memory**: Recalls past actions and their outcomes, enabling the agent to avoid repeated mistakes.
3. **Long-term Memory**: Accumulates experiences across sessions to provide adaptability and personalization.

Architectures like MemGen introduce "memory triggers" that monitor an agent's reasoning state to decide when to explicitly invoke memory. The "memory weaver" then synthesizes a machine-native latent token sequence to enrich the agent's current reasoning, producing a tightly interwoven cycle of memory and cognition.

Sociotechnical Implications and Governance

The deployment of agentic systems in enterprise and scientific contexts necessitates robust governance frameworks. As AI agents become "virtual coworkers," the role of IT departments shifts from managing infrastructure to overseeing the "hiring, training, and optimization" of these systems.

HR for AI Agents

In the emerging organizational model, IT acts as the "HR department" for AI agents, determining the scope of information they can access and ensuring their recommendations are transparent and verified. HR professionals, in turn, focus on the collaboration between human employees and their AI counterparts, ensuring interactions enhance productivity and well-being.

Risks and Safety Mechanisms

Agentic systems pose novel risks, including the "weaponization of AI-driven checkouts" in ecommerce or the potential for agents to develop divergent biases during unprompted operation. To address these, agentic architectures incorporate several safety layers:

- **Action Guards**: Developer-specified heuristics or LLM-based judgments that require explicit human confirmation for risky or irreversible actions.
- **Docker Sandboxing**: Launching and isolating agent teams per user task to prevent cross-contamination or unauthorized system access.
- **Audit Trails**: Generating verifiable execution paths and automation artifacts to ensure adherence to policies and standards.
- **RBAC and Governance Overlays**: Ensuring that agents can only perform actions for which the user's identity has explicit permission.

The Integrated Agentic Future

The architecture of the agentic interface, as exemplified by projects like **Inter**Weave and Orca, represents a fundamental reimagining of the human-computer relationship. By moving beyond the search bar and the linear chat window toward hierarchical sensemaking workspaces and malleable browser environments, these systems empower users to tackle cognitively complex tasks with unprecedented agility.

The convergence of distributed autonomous AI with centralized orchestration layers—what is now termed "Orchestrated Distributed Intelligence" (ODI)—allows for the creation of cohesive networks of agents that work in tandem with human expertise. Whether navigating the "quarks-to-cosmos" landscape of fundamental physics or managing the "Agentic Cloud Ops" of a global enterprise, the future of AI lies in these integrated, goal-oriented systems. The challenge for future developers and designers will be to balance this growing autonomy with transparency and user agency, ensuring that the next generation of AI serves not to replace human decision-making but to amplify it within a secure and ethically governed ecosystem.

Sources

1. **researchgate.net**
   Agentic Browsing and the Future of Human- Computer **Inter**action: From Playwright Automation to Agent-Driven **Inter**faces - ResearchGate
2. **researchgate.net**
   InterWeave: Presenting Search Suggestions in Context Scaffolds Information Search and Synthesis - ResearchGate
3. **arxiv.org**
   Orca: Browsing at Scale Through User-Driven and AI-Facilitated Orchestration Across Malleable Webpages - arXiv
4. **srishtipalani.github.io**
   Srishti Palani, Human-AI **Inter**action Researcher
5. **siliconangle.com**
   Microsoft's Azure Copilot to support agentic cloud operations at ...
6. **medium.com**
   The Agentic Shift: From Digital Tools to the Autonomous Workforce (2025–2027 Strategic Outlook) | by Phill Keene - Medium
7. **crn.com**
   Microsoft Ignite 2025: The Biggest News In AI, Agents, Data - CRN
8. **medium.com**
   Agents with Faces: The Personified User **Inter**face | by Jason Clark | Craine Operators Blog
9. **medium.com**
   Agentic AI: The Era of Self-Directed Intelligence | by Pathak Sanjai, Ph.D. - Medium
10. **arxiv.org**
    Bridging the Gulf of Envisioning: Cognitive Challenges in Prompt Based **Inter**actions with LLMs - arXiv
11. **download.bibis.ir**
    To my son, Bruno, who at two years old, brought a new and brilliant light into my life. As I explore the systems that will define - bibis.ir
12. **imergey.com**
13. Agentic AI in workplace collaboration – How autonomous AI agents will reshape jobs
    escholarship.org
    Beyond the Search Bar: Augmenting Discovery ... - eScholarship
14. **escholarship.org**
    UC San Diego Electronic Theses and Dissertations - eScholarship
15. **semanticscholar.org**
    Generative AI in Knowledge Work: Design Implications for Data Navigation and Decision-Making - Semantic Scholar
16. **sanghosuh.github.io**
    Sensecape: Enabling Multilevel Exploration and ... - Sangho Suh
17. **semanticscholar.org**
    A Taxonomy for Human-LLM **Inter**action Modes: An Initial Exploration - Semantic Scholar
18. **3dvar.com**
    Graphologue: Exploring Large Language Model Responses with **Inter**active Diagrams
19. **hci.ucsd.edu**

Luminate: Structured Generation and Exploration of Design Space with Large Language Models for Human-AI Co-Creation - Foundation **Inter**face Lab - University of California San Diego
20. **researchgate.net**
(PDF) Luminate: Structured Generation and Exploration of Design Space with Large Language Models for Human-AI Co-Creation - ResearchGate
21. **infoq.com**
Surfing the Web at Scale: Orca Explores a Human-Guided Future for AI Agents - InfoQ
22. **researchgate.net**
When the Tab Comes Due:Challenges in the Cost Structure of Browser Tab Usage
23. **arxiv.org**
ArgoLOOM: agentic AI for fundamental physics from quarks to cosmos - arXiv
24. **researchgate.net**
ArgoLOOM: agentic AI for fundamental physics from quarks to cosmos
25. **askui.com**
What is the ReACT Framework for Intelligent Agent AI? - AskUI
26. **openreview.net**
WHAT DO LLM AGENTS DO WHEN LEFT ALONE? EVIDENCE OF SPONTANEOUS META-COGNITIVE PATTERNS - OpenReview
27. **arxiv.org**
What Do LLM Agents Do When Left Alone? Evidence of Spontaneous Meta-Cognitive Patterns - arXiv
28. **reddit.com**
Can someone succinctly explain what AgentForce is without all the BS and hype? - Reddit
29. **windowsforum.com**
Microsoft Ignite 2025: Agent 365 and Fabric IQ Define the Agentic Enterprise
30. **azure.microsoft.com**
Azure at Microsoft Ignite 2025: All the intelligent cloud news explained
31. **reworked.co**
IT the HR of Agentic AI? Not So Fast - Reworked
32. **architect.salesforce.com**
Architecting the Agentic Enterprise with MuleSoft
33. **reddit.com**
You Don't Know How to Use AI : r/ArtificialSentience - Reddit
34. **openreview.net**
MEMGEN: WEAVING GENERATIVE LATENT MEMORY FOR SELF-EVOLVING AGENTS - OpenReview
35. **insightpartners.com**
The state of the AI Agents ecosystem: The tech, use cases, and economics | Insight Partners
36. **huggingface.co**
Daily Papers - Hugging Face
37. **arxiv.org**
Position Paper: Towards Open Complex Human–AI Agents Collaboration Systems for Problem Solving and Knowledge Management - arXiv
38. **domino.ai**
Agentic AI frameworks for enterprise | Domino Data Lab
39. **riskified.com**
Who owns the risk when AI pays? The coming accountability void in agentic commerce
40. **emergentmind.com**

Magentic-UI: Advances in Magnetic User **Inter**faces - Emergent Mind

41. **aclanthology.org**
    Code to Think, Think to Code: A Survey on Code-Enhanced Reasoning and Reasoning-Driven Code Intelligence in LLMs - ACL Anthology
42. **azure.microsoft.com**
    Announcing Azure Copilot agents and AI infrastructure innovations | Microsoft Azure Blog
43. **spdow.ucsd.edu**
    Research - Steven Dow
44. **arxiv.org**
    From Autonomous Agents to Integrated Systems, A New Paradigm: Orchestrated Distributed Intelligence - arXiv
45. **arxiv.org**
    From Autonomous Agents to Integrated Systems, A New Paradigm: Orchestrated Distributed Intelligence - arXiv
46. **researchgate.net**
    From Autonomous Agents to Integrated Systems, A New Paradigm: Orchestrated Distributed Intelligence - ResearchGate