Conversational AI. Dialogsysteme, Chatbots, Assistenten

Veranstalter: Christoph Ringlstetter

Sitzung 8: Agents

Was machen wir denn heute.

- Orga. Referate, Zulassung, Termine.
- News of the Week
- Referat Agenten
- Besprechung Paper Tula Masterman et al. The Landscape of Emerging AI Agent Architectures

Tula Masterman et al. AI Agent Architectures – Überblick

Ausgangsthese: Komplexe Ziele erfordern Nachdenken – Reasoning

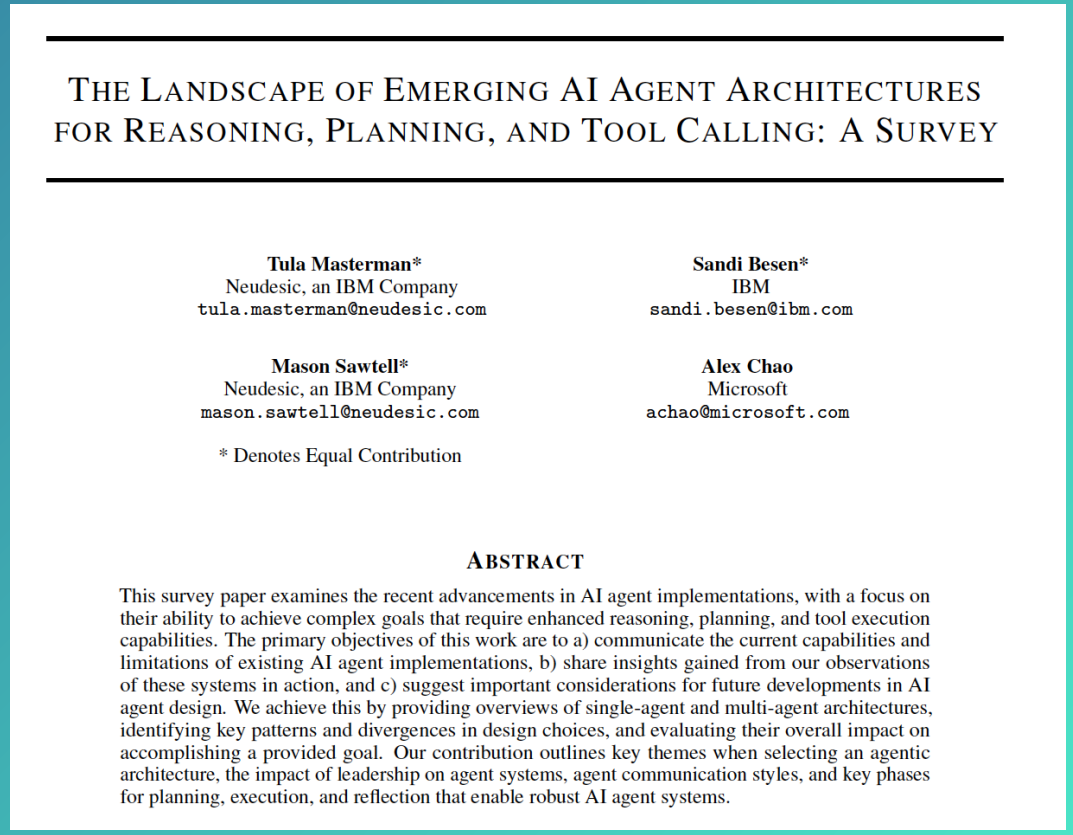
1) Fähigkeiten und Grenzen von Agentenimplementierungen

2) Einsichten aus Beobachtungen

3) Zukünftige Entwicklungen

Single-, Multiagent Architekturen

Umgebung, Kommunikation, Schlüsselphasen: planning, execution, reflection



THE LANDSCAPE OF EMERGING AI AGENT ARCHITECTURES FOR REASONING, PLANNING, AND TOOL CALLING: A SURVEY

Tula Masterman*
Neudesic, an IBM Company
tula.masterman@neudesic.com

Sandi Besen*
IBM
sandi.besen@ibm.com

Mason Sawtell*
Neudesic, an IBM Company
mason.sawtell@neudesic.com

Alex Chao
Microsoft
achao@microsoft.com

* Denotes Equal Contribution

ABSTRACT

This survey paper examines the recent advancements in AI agent implementations, with a focus on their ability to achieve complex goals that require enhanced reasoning, planning, and tool execution capabilities. The primary objectives of this work are to a) communicate the current capabilities and limitations of existing AI agent implementations, b) share insights gained from our observations of these systems in action, and c) suggest important considerations for future developments in AI agent design. We achieve this by providing overviews of single-agent and multi-agent architectures, identifying key patterns and divergences in design choices, and evaluating their overall impact on accomplishing a provided goal. Our contribution outlines key themes when selecting an agentic architecture, the impact of leadership on agent systems, agent communication styles, and key phases for planning, execution, and reflection that enable robust AI agent systems.

Tula Masterman et al. AI Agent Architectures – Überblick

Nächste Generation von KI – Anwendungen: Agents

→ Beispiele: GPT-4 + AutoGPT, Baby AGI etc

Agenten folgen einer komplexeren Architektur als bloße LLMs: Interaktion, Orchestrierung, Planung, Loops, Reflexion und andere Kontrollstrukturen.

Verbunden mit: tools, plugins, function calling: General Purpose Aufgaben.

Für vollständig definierte Aufgaben: Single Agent

Feedback von anderen Personas, Kollaboration für Lösungen benötigt: Multi Agent

Tula Masterman et al. AI Agent Architectures – Überblick

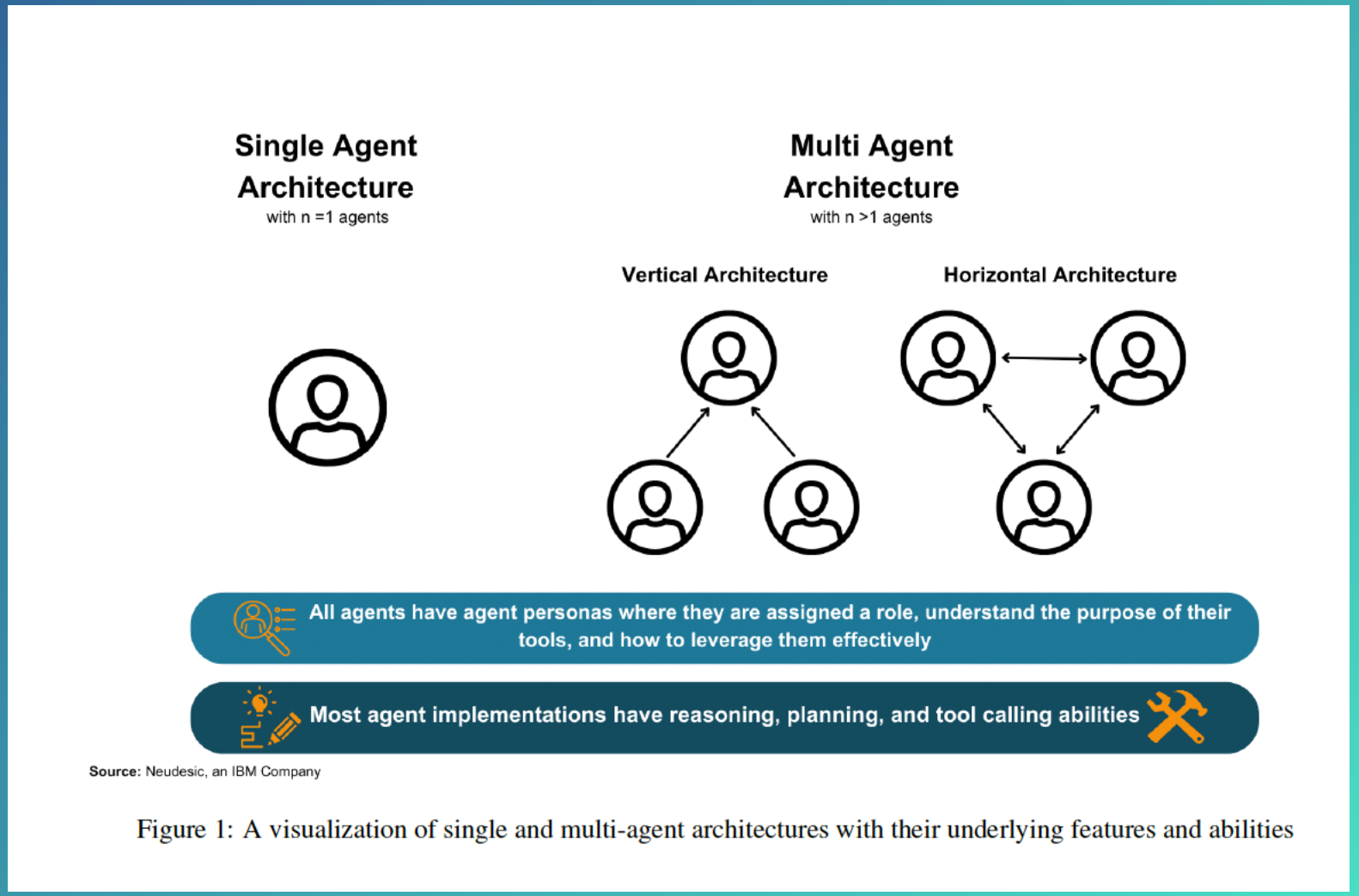


Figure 1: A visualization of single and multi-agent architectures with their underlying features and abilities

Tula Masterman et al. S.2

Tula Masterman et al. AI Agent Architectures – Taxonomie

Agenten als LLM powered Entities. Können Planen, Aktionen durchführen, Reflektieren und damit iterativ Ziele erreichen.

→ Persona, Tools, Memory:

→ brain, perception und action [31] the agent understands, reasons and acts in the environment.

Tula Masterman et al. AI Agent Architectures – Taxonomie

Persona: Rolle, Persönlichkeit: Agent bewußt welche Tools er zur Verfügung hat => beeinflusst das LLM Ausgabeverhalten per Prompting
Agent besser bei der Taskcompletion als LLM Chain of Thought.

Tools: Funktionen mit Auswirkungen auf die Umwelt (objektiv, virtuell) die der Agent aufrufen kann. Pull-Push Info

Tula Masterman et al. AI Agent Architectures – Taxonomie

Single Agent Architectures: ein LLM mit einem Agent Prompt und Tools. Feedback falls überhaupt durch Menschen.

Multi Agent Architectures: typischerweise jeder Agent mit eigener Persona.

Vertical: Leader mit direct reports

Horizontal: all equals, shared thread: jeder kann die Nachrichten der anderen live sehen – siehe Bücher zu Notification, Wissenssharing.

Gruppenanstrengung um die Aufgabe zu erledigen. [8]

Tula Masterman et al. AI Agent Architectures – Haupteigenschaften

Effektive Agenten: besitzen Problemlösefähigkeit:
reasoning, planning, tool calling.

Reasoning und Planen: Durch Reasoning sind Entscheidungen möglich. Agents müssen starke Fähigkeit zum Nachdenken haben um effektiv mit der Umwelt zu interagieren => Anpassen von Plänen aufgrund von Feedback neuen Infos.

Tula Masterman et al. AI Agent Architectures – Haupteigenschaften

Planen: Dekonstruieren. Task Decomposition. Multi-Plan Auswahl, Externes modulbasiertes Planen, Reflexion und Verfeinerung. Memory basiertes Planen.

1) Aufgabe in Unteraufgaben zerlegen
2) Einen Plan aus mehreren Optionen auswählen
3) Externes Planing Modul benutzen
4) Revidieren von Plänen
5) Externe Informationen akquirieren um den Plan zu verbessern

Agentenimplementierungen haben üblicherweise einen Planschritt bevor Aktionen durchgeführt werden. Z.B. Plan like a Graph.

Tula Masterman et al. AI Agent Architectures – Haupteigenschaften

Tool Calling: Ein Schlüsselelement der Agentenimplementierung im Vergleich zu bloßem Prompting.

Komplexe Aufgaben können oft nur durch Tooleinsatz gelöst werden. APIs, Datenquellen.

Methoden zum Toolaufruf und zur Reflexion ermöglichen meist mehrere Runden.

Tula Masterman et al. AI Agent Architectures – Haupteigenschaften

Dekonstruktion: Breaking größerer Probleme in Subprobleme.

→ lösen dann in einer Sequenz eins nach dem anderen mit passenden Tools

→ manche Arbeiten beschreiben dass sich Agenten mit längeren Toolaufrufsequenzen schwer tun: getting lost. Über die Prompts Subtasks konsolidieren und wieder Resetting durchführen.

→ Ev auch auf Multiagents zurückgreifen

Tula Masterman et al. AI Agent Architectures – Haupteigenschaften

Single-Agent Architekturen:

ReAct, RAISE, REFLECTION, AUTOGPT+P => Frameworks mit Reasoning Stage bevor das Problem angefasst wird.

Schlüsselthemen aus Vergleich: Planning und Selbstkorrektur

→ sonst besteht die Gefahr eine Endless Loop von Executions ohne Zielerreichung

=> Präferenz dieser Architektur für Usecases die lösbar sind über: straightforward function calling: kein Feedback von anderen Agenten benötigt.

Tula Masterman et al. AI Agent Architectures – Haupteigenschaften

Single Agent Architekturen Beispiele:

ReAct. Reason + Act

Nachdenken über die Aufgabe → Aktion → Beobachten des Output

=> Iterriert bis die Aufgabe vollständig ist [32]

+ Effektiver als Zero Shot

+ Einbezug von Menschen, Vertrauen: interoperability, trustworthiness offener Prozess der Aufgabenerstellung

- eventuell viele Loops aus Nachdenken und Aktionen: Lost

Tula Masterman et al. AI Agent Architectures – Single Agent Architektur

React Beispiel

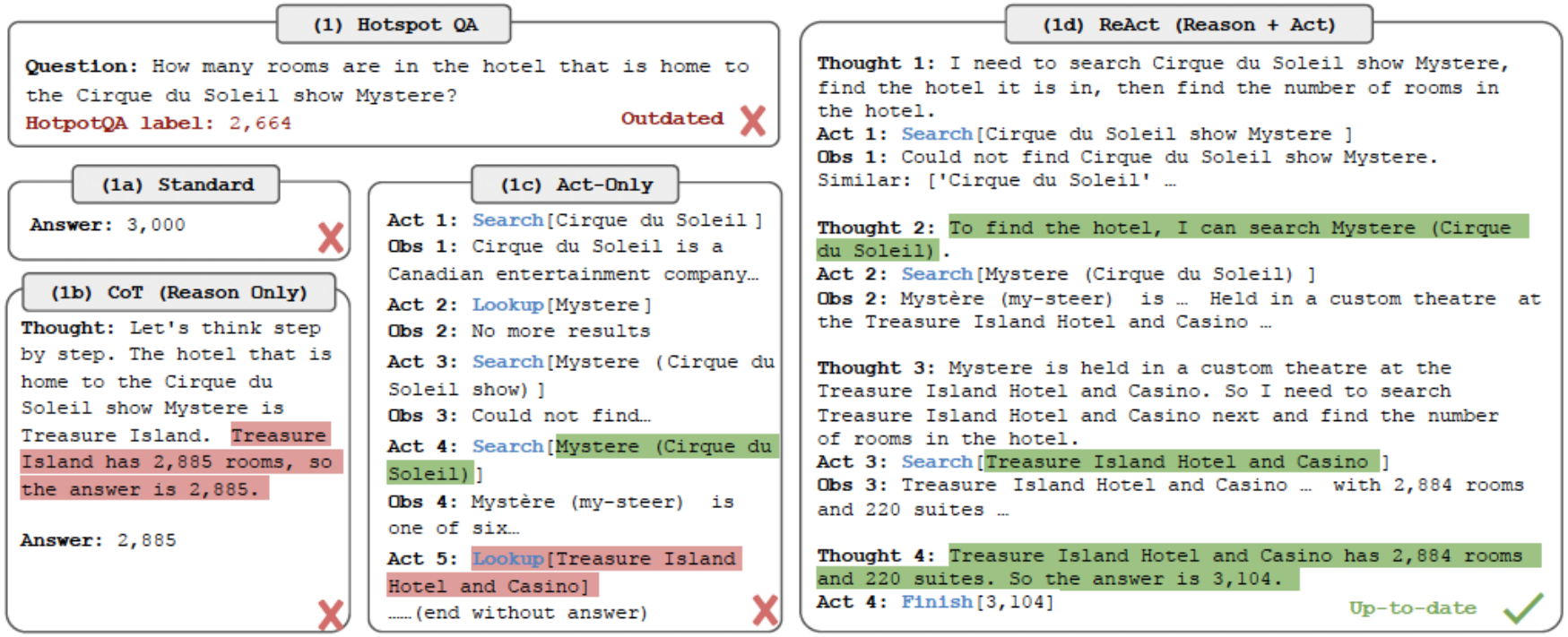


Figure 2: An example of the ReAct method compared to other methods [32]

Mastermann S.5

Tula Masterman et al. AI Agent Architectures – Single Agent

RAISE: Mit Memory Mechanismus ST/LT. [16]

Scratchpad für ST, Datensatz für LT

+ Context für längere Konversationen

+ Finetuning für spezielle Aufgaben superior

- Komplexe Logik

- Halluzinationen zu Rollen „Sales Agent“ coded statt zu verkaufen

Tula Masterman et al. AI Agent Architectures – Single Agent

REFLEXION: Selbstreflexion durch linguistisches feedback: successrate, trajectory persistentes Memory.

LLM Evaluator gebt Feeback an den Agenten.

+ Success Rate höher, reduzierte Halluzinationen

- lokale Minima Lösungen. Bleibt hängen.

- LT ist ein sliding Window nicht Datenbank. Limitiert auf Tokenlimit.

Tula Masterman et al. AI Agent Architectures – Single Agent

AutoGPT+P: Reasoning Limits adressieren.

Für Agenten die Roboter in natürlicher Sprache steuern.

Objekterkennung + Object-Affordance-Mapping OAM

Plannungssystem durch LLMs realisieren

Start: Bild einer Szene: Objekte erkennen.

Sprachmodell nutzt die Objekte um im Planning Tool den Plan anzupassen.

LLM nutzt zusätzlich einen klassischen Planer: Planing Domain Definition Language (PDDL)

Zum Zeitpunkt des Papers begrenzte Reasoning Kapazitäten des LLMs Kombi superior

-- ACC Tool Selection variiert, Unlogische Exploration, Loops, Interceptionfähigkeit begrenzt. Folgt dem falschen Plan.

Tula Masterman et al. AI Agent Architectures – Single Agent

LATS: Language Agent Tree Search.

Bäume für Planen, Actions, Reasoning

+ starke Verbesserung durch Self-Reflection

- hoher Ressourcenverbrauch. Langsamer.

Tula Masterman et al. AI Agent Architectures – Multi Agent

Embodied LLM Agents learn to cooperate in organized teams: Dylan, Agent Verse und MetaGPT.

Zielerreichung durch Inter-Agent-Kommunikation. Nicht einfach. Grounding Probleme? -> Alle haben Zugriff auf alle Kommunikation und verstehen Sie?

LEISS KURS

Schlüsselthemen: Verteilung der Arbeit nach Skills. Feedback von Agent Personas.

Viele Multi-Agent Frameworks erzeugen Agentteams passend zum Problem: planning, execution, review

Klare Leadership Struktur und Information Sharing.

Tula Masterman et al. AI Agent Architectures – Multi Agent

Beispiele:

Embodied LLM Agents Learn to Operate in Organized Teams.

Lead Agent: Reserach by Guo [9]

10% Vorteil gegenüber single

Ohne Leader 50% Kommunikation sind Befehle/Aufgaben die gegenseitig gegeben werden.

Am effektivsten: Stand Paper: der Leader ist ein Mensch.

60% der Leader Kommunikation – Directions.

Andere wichtige Arten: kritisieren, Reflektionsschritt um Pläne zu generieren.

Performanz evaluieren, Feedback zur Verfügung stellen, Team reorganisieren, Leadership rotieren.

Tula Masterman et al. AI Agent Architectures – Multi Agent

Dylan: Dynamic LLM Agent Network. Agenten Struktur für komplexe Aufgaben.

Coding, Reasoning.

Prunning: Für jede Runde rücken nur die Bestperformer weiter.

Agent Verse: Gruppenphasen im Planning verbessern das Reasoning und das Problemsolving.

4 Stages: Recruiting, Collaborative Entscheidung, Unabhängige Aktion, Evaluation.

Tula Masterman et al. AI Agent Architectures – Multi Agent

Masterman et al. S.8

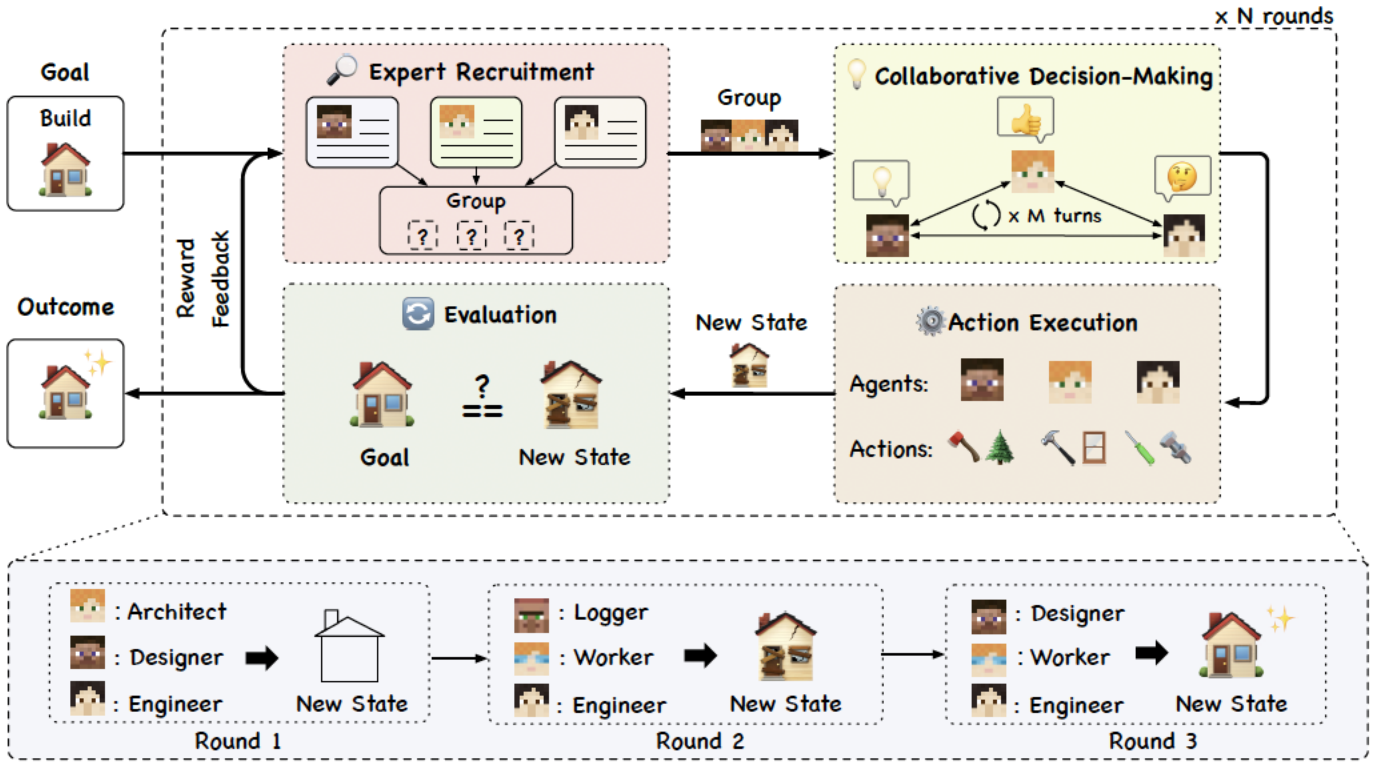


Figure 6: A diagram of the AgentVerse method [2]

Tula Masterman et al. AI Agent Architectures – Multi Agent

MetaGPT: Konversation während der Arbeit an der Task. Chatter Gefahr =>

Agenten dürfen nur via strukturierte Outputs kommunizieren.

→ publish-subscribe: Prozess

Evaluationsresultate viel besser als ohne Kommunikation.

Tula Masterman et al. AI Agent Architectures – Diskussion

Typische Bedingungen um Single/Multi Agent Architektur zu entscheiden.

Single Agent: Tasks eng definiert. Tools, Prozesse die angewandt werden können klar.

→ einfacher zu implementieren. Kein Chatter, kein unpassendes fehlleitendes Feedback.

→ aber Gefahr stecken zu bleiben. Loop. Reasoning nicht robust.

Multi Agent: Personas geben hilfreiches Feedback, Aufgabe komplex. Parallelisierung, abgrenzbare Aufgaben ohne Beispiele.

→ aber komplexeres Leadership- und Kommunikationsmanagement.

Tula Masterman et al. AI Agent Architectures – Diskussion

Lit. 26 Widerspruch. Wenn der Prompt robust genug ist kein Unterschied im Paper feststellbar zwischen Multi und Single Agent.

Asynchrone Taskausführung: Single Agent kann auch multiple Threads haben aber es gibt keine Operationsmodell zur parallelen Ausführung: muss warten.

Dagegen Multi Agent: jeder kann schon abarbeiten.

Input Feedback Human Oversight: Lösung wird genähert und dann Kritik Prozess. Oder ein anderer Agent wird um konstruktives Feedback gebeten.

LLMs Problem: legen sich früh auf eine Antwort fest: Snowball Effekt → Diversion vom Zielstustand.

Über Agent: Feedback zur Kurskorrektur + Human Oversight [4,9]

Tula Masterman et al. AI Agent Architectures – Diskussion

Problem: *sycophantic* Verhalten von LLMs kann dem klaren Feedback entgegenstehen.

Spiegelt nur die Haltung des Users wieder. [20] in manchen Fällen sogar biased.

Insgesamt wäre der Idealprozess: Planabweichung, User Feedback, Korrektur

<u>Group Conversations, Information Sharing:</u>

Multi Agent: belanglose Kommunikation: „Wie geht's dir Agent B" v.a. horizontal wenn Groupchats implementiert sind.

Auf Anfragen wird zu ausführlich geantwortet. Unsinnige Kommunikationen.

=> Lösung Message Subscripting.

Tula Masterman et al. AI Agent Architectures – Diskussion

Information Sharing: nur Infos die Task relevant sind.

Vertikal: Probleme treten auf wenn der Leader kritische Information nicht weiter gibt.

Klarer System Prompt mit Zugangsrechten.

Klare Rollendefinition im Multiagent Szenario: Jeder Agent mit entsprechend spezifischem Prompt.

Tula Masterman et al. AI Agent Architectures – Diskussion

Limitierungen und Zukunft: Evaluation. Wie sollen Agenten bewertet werden?

Es gibt keinen Zentralen Prozess. Risiken für Overfitting etc.

Datenkontamination: Testdaten sind in die Trainingskorpora geraten. Evaluierte Performanz sinkt dramatisch wenn die Tests verändert werden.

Neue Fähigkeiten LLM-Agent sind in den Daten nicht abgebildet.

Agent Bench für Einfache Aufgaben gut als Test Bench geeignet – sonst nicht.

Bias und Fairness: Agenten importieren die Proleme des LLMs