Building Persona-Based Agents On Demand: Tailoring Multi-Agent Workflows to User Needs

GIUSEPPE ARBORE*, Politecnico di Torino, Italy
ANDREA SILLANO*, Politecnico di Torino, Italy
LUIGI DE RUSSIS, Politecnico di Torino, Italy

Recent advances in agentic AI are shifting automation from discrete tools to proactive multi-agent systems that coordinate multi-specialized capabilities behind unified interfaces. However, today's agent systems typically rely on hard-coded agent architectures with fixed roles, coordination patterns, and interaction flows that limit end-user personalization and make adaptation to individual needs and contexts difficult. Given this limitation, we argue that on-demand persona-based agent generation offers a promising path towards more efficient and contextually appropriate interaction within agentic workflows. By dynamically crafting agents and personas at run-time to match user characteristics, task demands, and workflow context, agentic platforms can move beyond one-size-fits-all configurations. We present a pipeline for on-demand persona generation in agentic platforms, detailing how real-time crafting of AI personas can be systematically integrated within agent systems, aiming to open new possibilities in agentic platform design paradigms.

CCS Concepts: • **Computing methodologies** → **Multi-agent systems**; *Intelligent agents*; • **Human-centered computing** → **Human computer interaction (HCI)**.

Additional Key Words and Phrases: multi-agent systems; persona-based agents; workflow automation; agent orchestration; human–AI collaboration; personalization

ACM Reference Format:
Giuseppe Arbore, Andrea Sillano, and Luigi De Russis. 2026. Building Persona-Based Agents On Demand: Tailoring Multi-Agent Workflows to User Needs. In *Proceedings of AutomationXP26 Workshop of the 2026 CHI Conference on Human Factors in Computing Systems, April 14, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 6 pages.

1 Introduction and Background

Agentic AI is a rapidly evolving paradigm. Recent advances are shifting automation from discrete, user-invoked tools toward proactive multi-agent systems that coordinate multiple specialized capabilities behind unified interfaces, enabling agents to plan, reason, and autonomously execute multi-step workflows end-to-end [14]. These systems are increasingly supported by concrete architectural components (e.g., persistent memory, structured knowledge resources, reflection mechanisms, and feedback loops) as well as orchestration frameworks that manage coordination and division of labor across multiple agents [9]. However, recent LLM-based multi-agent systems are often engineered as a set of agents with pre-defined roles ("profiles") and fixed coordination rules (e.g., who communicates with whom, in what order, and through which intermediate artifacts). Prior research identifies *agent profiling* and *communication* as core architectural

*Both authors contributed equally to this research.

Authors' Contact Information: Giuseppe Arbore, giuseppe.arbore@polito.it, Politecnico di Torino, Torino, Italy; Andrea Sillano, andrea.sillano@polito.it, Politecnico di Torino, Torino, Italy; Luigi De Russis, luigi.derussis@polito.it, Politecnico di Torino, Torino, Italy.

Manuscript submitted to ACM

dimensions of LLM-based multi-agent systems, and surveys a wide range of approaches in which these choices are specified upfront through prompt templates, role descriptions, and preset interaction protocols [7]. For example, Bo et al. [1] introduce a collaboration framework in which multiple *actor* agents are guided by a shared *reflector* that generates role-conditioned reflections; while effective, the division of labor and the role schema are designed at the system level [1]. Similarly, Ding et al. [4] hard-wire a two-phase, asynchronous, multi-level communication scheme where higher-level agents decide first and propagate information and actions to lower-level agents, thereby embedding a specific coordination pattern into the pipeline [4].

While these system achieve remarkable results in executing and completing tasks, their effectiveness in real-world interaction scenario can be limited by the rigid nature of their configuration. One promising approach to improve LLMs' adaptability is the use of AI-generated personas: constructed identities that changes how the model communicates, reasons, and performs across different tasks and contexts. Persona effects are prompt-sensitive, and changes in how a persona is described (e.g., role-adoption framing, sociodemographic priming, or constraint wording) can substantially alter both open- and closed-ended outputs [12]. At the same time, persona prompting can influence more than just surface form, as it is able to modulate capabilities and strategy selection. For instance, "role play prompting" reports consistent gains in zero shot reasoning across multiple benchmarks, suggesting that persona instruction can induce the model towards effective reasoning paths. Recent work in evaluation further shows that writing style and persona condition formulations can systematically shift performances in different tasks [15]. This also motivates persona-aware task like information retrieval mechanism, where embedding-based retrieval can exhibit preferences over certain query styles, affecting how the task is performed [2]. Beyond modulating reasoning strategies, personas can provide an interaction-level control mechanism that possibly reduce friction between users and systems, by aligning responses with user expectation or preferences about tone, verbosity, and tone depth. Empirical studies on chatbot interaction show that variation in linguistic style can significantly affect perceived credibility, engagement, and customer satisfaction indicating that how the model speaks is a measurable metric of interaction quality [3, 5, 6]. Work on personalized interaction with persona- or profile-conditioned generation to adapt assistance to context shows that users benefit from tailored outputs. For example, we can find this benefit in tutoring system that adapt pedagogy over time or writing assistant that personalize generation to match author's style and preferences [10, 13, 17].

Building on these observation, LLMs persona are not considered anymore as just styling editors, but instead, they can be actually treated as controlled variables inside end-to-end workflows in agentic architectures. PersonaAgent [16] formalizes personalization for agentic LLM by using a user-specific persona prompt as an intermediary between (i) personalized memory (episodic/semantic) and (ii) personalized actions (tool use), and introduces a test-time alignment strategy that optimizes the persona from recent interactions to better match user. Evaluations are also emerging for tool-using settings, such as ETAPP that introduces a benchmark designed specifically to measure personalized tool invocation (not only personalized text), enabling agentic evaluation under diverse user profiles [8]; or Persona-Plug [11] which proposes a plug-and-play persona representation learned from user history (a user embedding attached to inputs) to personalize generation without fine-tuning, offering a practical mechanism for injecting stable user preferences into inference-time behavior.

As agentic AI moves from discrete automation tools to coordinated workflows of specialized components, current systems still encode roles, coordination patterns, and interaction protocols as fixed design-time choices. This design makes it harder to tailor interaction patterns to different users and contexts without revising prompts, role definitions, or orchestration logic. We argue, instead, for run-time persona-conditioned agent generation, where agents (their roles, interaction policies, and tool-use strategies) are synthesized on demand to match the user, their task, and the

evolving workflow state. In this framing, persona-based generation can become a mechanism for adaptivity in agentic platforms: different users and contexts can induce different agent configurations and coordination strategies, possibly leading to more natural interaction and also to different performance profiles depending on who is using the system and under what conditions. To this end, this paper presents a pipeline for on-demand persona generation in agentic platforms, detailing how real-time persona crafting can be systematically integrated within agent systems, enabling context-sensitive adaptivity.

2 Method

To address the friction that users can experience when dealing with multi-agent system, we propose a on-demand agent generation that is not tied to a fixed schema of agent profiles. Consequently, we enable the on-demand synthesis of agent personas at run-time, allowing the system to dynamically craft roles, interaction styles, and coordination behaviors in response to user characteristics, task demands, and workflow context. Instead of requiring manual configuration or predefined agent hierarchies, we treat agent personas as generative constructs that are instantiated and adapted continuously throughout the course of an interaction.

The pipeline is initiated when the user submits a query to the system. At this stage, the user interacts with the system freely, without any structural constraints on how the query is formulated as it can range from a simple question to a complex, multi-part request. This open-ended prompting phase is intentional, as it allows the system to capture the full richness of user intent without forcing predefined templates or interaction patterns. Once the query is received, it is passed to the orchestrator, which serves as the central coordinator of the entire pipeline, managing all subsequent steps without any further intervention required from the user. The on-demand persona generation is composed by four different steps in the orchestrator: (i) Query Analysis, (ii) Agent Generation and Instantiation, (iii) Agent Assigning and Execution, and (iv) Answers Aggregation and Displaying. The pipeline operates on a session-based model, meaning that all the contextual information (e.g., user information, generated persona, and spawned agents) gathered during an interaction are bounded to a single session. This is a deliberate design choice grounded on two main reasons: keeping coherence across the same session with multiple queries, and ensuring that system is not task dependent and can easily adapt itself. An overview of how the pipeline process work is shown in Algorithm 1, while a step-by-step visualization is represented in Figure 1.

Step 1 - Query Analysis. The first step executed by the orchestrator is responsible for transforming the raw user input into a structured representation that can be later used in the pipeline. It consists of two sub-steps that run in sequence. The first sub-step is *ProfileEncode*, which uses implicit or explicit signal from the user query to create a representation of the user and their intent. This profile is intended to capture attributes like domain expertise, preferred communication style, task familiarity, and, crucially, the intent of the query, serving as conditioning knob for persona crafting in the next steps. The second sub-step is *TaskDecompose*, where the orchestrator breaks the query down into a set of discrete tasks. Ideally, this decomposition is more than a flat list; instead, it has to capture dependencies between tasks, determining which tasks can be executed in parallel and which must follow a sequential order. The output of this step is a structured task plan that can be later employed by the orchestrator to manage the generated agents.

Step 2 - Agent Generation and Instantiation. The second step is responsible of translating the plan produced the previous step into a set of operational agents. The first sub-step is *PersonaCraft*, where the orchestrator can dynamically craft a persona for each sub-task. This step is the central part of the entire pipeline. Instead of relying on predefined role the system can generate needed personas and agents on-demand, allowing for a non-fixed architecture. Moreover,

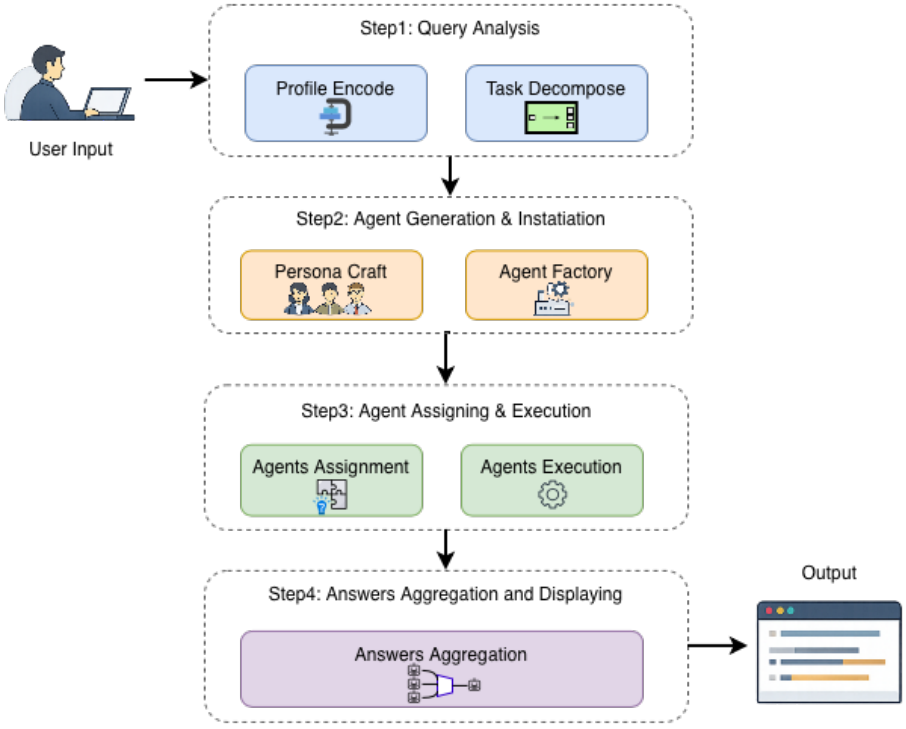

Fig. 1. A visual representation of the pipeline's steps

this process is aware of already existing agents and roles, allowing for precise persona development and avoiding duplicates. Concretely, a persona is a structured specification that defines the agent's role, the domain competencies it should exhibit, the communication style it should adopt when contributing to the final response, and what capabilities it owns. All these pieces of information are extracted in the previous step. The second sub-step is *AgentFactory*, where each synthesized persona is used to initialize one agent instance. The persona specification is passed to configure an LLM-backed agent, shaping its reasoning style, interaction behavior, and output format. The output of this step is a pool of agents with assigned personas, optimized for task-specific goals.

Step 3 - Agent Assigning and Execution. The third step executed in the pipeline coordinates collaboration among agents by assigning each task to its corresponding agent, determining an execution order derived directly from the dependency structure of the tasks produced during decomposition, and managing dependencies and information flow. Tasks with no dependencies each one other are scheduled for parallel execution, while those that depend on the output of a preceding task are queued for sequential execution, ensuring that no agent begins its task before the information it requires is available. Each agent executes its assigned task and produces a partial result that is used by the orchestrator that, after inspecting the dependency graph to determine which agents require that result as input, passes it accordingly before triggering their execution. This controlled information routing ensures that each agent operates with the context it needs while remaining decoupled from the internal workings of other agents in the pool, preserving the modularity of the system.

Algorithm 1 On-Demand Persona-Based Agent Generation

Require: User queries q
Ensure: Response $\mathcal{R}$ delivered to user
Step 1: Query Analysis
Extract user characteristics $u \leftarrow \textsc{ProfileEncode}(\mathcal{U}, q)$
Decompose query into tasks: $\{t_1, \ldots, g_n\} \leftarrow \textsc{TaskDecompose}(q, u)$
Step 2: Agent Generation and Instantiation
for each task t_i **do**
    Synthesize persona: $p_i \leftarrow \textsc{PersonaCraft}(t_i, u, p)$
    Instantiate persona-based agent: $a_i \leftarrow \textsc{AgentFactory}(p_i)$
end for
Step 3: Agents Assigning and Execution
Orchestrator assigns t_i to a_i and manages execution order
for each agent a_i in execution order **do**
    Agent execution: $r_i \leftarrow a_i.\textsc{Execute}(t_i)$
end for
Step 4: Answers Aggregation and Displaying
Orchestrator answer aggregation: $\mathcal{R} \leftarrow \textsc{Aggregate}(\{r_1, r_2, \ldots, r_n\})$
return $\mathcal{R}$ to user

Step 4 - Answers Aggregation and Displaying. After all agents have produced their results, the system integrates them into a final response. This aggregation typically involves selecting, merging answers, and resolving eventual inconsistencies across agents, removing redundancies, and aligning the final output with the user's requested style and format. The system then delivers a single coherent answer that preserves the benefits of specialization while remaining concise, consistent, and directly usable by the user.

At each new user query within the same session, the pipeline is re-executed. Previously instantiated agents and persona profiles are retained rather than discarded; when additional specialization is required, new agents/personas can be created and appended to the existing pool.

3 Conclusions

This paper outlines how AI-generated personas can be integrated into agentic platforms to move beyond traditional schema-fixed architectures, thereby increasing personalization and adaptability across users and tasks. Our proposal presents a system, where persona-conditioned agent generation is treated as one of the architectural pillars alongside tool use, memory, and orchestration in agentic platforms. By moving agent roles, interaction styles, and behaviors from design-time constants to runtime variables, we aim to make systems being able to adapt and align the agents and the users in real-time. Users are no longer forced to understand and to adapt to a fixed agent topology; instead, it is the system that tailors itself to the user. Our proposal can carry implications that may go beyond usability, as it positions on-demand persona-augmented agent generation as a mechanism to empower end-users, enabling them to increase personalized interactions across different backgrounds, expertise levels, and task contexts. As agentic platforms grow in capability and complexity, leveraging persona-aware designs combined with runtime adaptability may contribute to keep these systems accessible, adaptable, and meaningful to users.

References

[1] Xiaohe Bo, Zeyu Zhang, Quanyu Dai, Xueyang Feng, Lei Wang, Rui Li, Xu Chen, and Ji-Rong Wen. 2024. Reflective Multi-Agent Collaboration based on Large Language Models. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 138595–138631. doi:10.52202/079017-4397

[2] Hongliu Cao. 2025. Writing Style Matters: An Examination of Bias and Fairness in Information Retrieval Systems. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining* (Hannover, Germany) *(WSDM '25)*. Association for Computing Machinery, New York, NY, USA, 336–344. doi:10.1145/3701551.3703514

[3] Ana Paula Chaves, Jesse Egbert, Toby Hocking, Eck Doerry, and Marco Aurelio Gerosa. 2022. Chatbots Language Design: The Influence of Language Variation on User Experience with Tourist Assistant Chatbots. *ACM Trans. Comput.-Hum. Interact.* 29, 2, Article 13 (Jan. 2022), 38 pages. doi:10.1145/3487193

[4] Ziluo Ding, Zeyuan Liu, Zhirui Fang, Kefan Su, Liwen Zhu, and Zongqing Lu. 2024. Multi-Agent Coordination via Multi-Level Communication. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 118513–118539. doi:10.52202/079017-3763

[5] Ela Elsholz, Jon Chamberlain, and Udo Kruschwitz. 2019. Exploring Language Style in Chatbots to Increase Perceived Product Value and User Engagement. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval* (Glasgow, Scotland UK) *(CHIIR '19)*. Association for Computing Machinery, New York, NY, USA, 301–305. doi:10.1145/3295750.3298956

[6] Yafeng Fan, Xiaohui Yue, Xiadan Zhang, and Luyao Zhang. 2026. Elaborate or Succinct? The Impact of AI Chatbots' Language Style on Customers' Satisfaction in Online Service. *Journal of Theoretical and Applied Electronic Commerce Research* 21, 2 (2026). doi:10.3390/jtaer21020051

[7] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 8048–8057. doi:10.24963/ijcai.2024/890 Survey Track.

[8] Yupu Hao, Pengfei Cao, Zhuoran Jin, Huanxuan Liao, Yubo Chen, Kang Liu, and Jun Zhao. 2025. Evaluating Personalized Tool-Augmented LLMs from the Perspectives of Personalization and Proactivity. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 21897–21935. doi:10.18653/v1/2025.acl-long.1064

[9] Dr. Sanjay Nakharu Prasad Kumar. 2025. Building Scalable and Reliable Agentic AI Systems: A Technical Blueprint for Autonomous Intelligence. *Global Journal of Engineering and Technology Research* (2025). https://api.semanticscholar.org/CorpusID:283433037

[10] Dejian Liu, Ronghuai Huang, Ying Chen, Michael Agyemang Adarkwah, Xiangling Zhang, Xin Li, Junjie Zhang, and Ting Da. 2024. *Personalized Tutoring Through Conversational Agents*. Springer Nature Singapore, Singapore, 59–85. doi:10.1007/978-981-97-5826-5_4

[11] Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2025. LLMs + Persona-Plug = Personalized LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 9373–9385. doi:10.18653/v1/2025.acl-long.461

[12] Marlene Lutz, Indira Sen, Georg Ahnert, Elisa Rogers, and Markus Strohmaier. 2025. The Prompt Makes the Person(a): A Systematic Evaluation of Sociodemographic Persona Prompting for Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 23212–23237. doi:10.18653/v1/2025.findings-emnlp.1261

[13] Armand Nicolicioiu, Eugenia Iofinova, Andrej Jovanovic, Eldar Kurtic, Mahdi Nikdan, Andrei Panferov, Ilia Markov, Nir Shavit, and Dan Alistarh. 2025. Panza: Design and Analysis of a Fully-Local Personalized Text Writing Assistant. arXiv:2407.10994 [cs.CL] https://arxiv.org/abs/2407.10994

[14] Dr. Urmila R. Pol. 2025. Generative AI, AI Agents, and Agentic AI : An Overview of Current AI Technologies. *International Journal for Research in Applied Science and Engineering Technology* (2025). https://api.semanticscholar.org/CorpusID:283379174

[15] Kimberly Truong, Riccardo Fogliato, Hoda Heidari, and Steven Wu. 2025. Persona-Augmented Benchmarking: Evaluating LLMs Across Diverse Writing Styles. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 22676–22709. doi:10.18653/v1/2025.emnlp-main.1155

[16] Weizhi Zhang, Xinyang Zhang, Chenwei Zhang, Liangwei Yang, Jingbo Shang, Zhepei Wei, Henry Peng Zou, Zijie Huang, Zhengyang Wang, Yifan Gao, Xiaoman Pan, Lian Xiong, Jingguo Liu, Philip S. Yu, and Xian Li. 2025. PersonaAgent: When Large Language Model Agents Meet Personalization at Test Time. arXiv:2506.06254 [cs.AI] https://arxiv.org/abs/2506.06254

[17] Theresa Zobel and Christoph Meinel. 2025. Chatbot Personas as a Gateway to Enhanced Learning Experiences. In *Advances in Information and Communication*, Kohei Arai (Ed.). Springer Nature Switzerland, Cham, 208–220.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009