

Toward A Pattern Language for Persona-based Interactions with LLMs

WILLIAM SCHREIBER, Vanderbilt University

JULES WHITE, Vanderbilt University

DOUGLAS C. SCHMIDT, William & Mary

A large language model (LLM) is a generative AI system that is (1) trained on vast amounts of text data to understand and generate human-like language and (2) capable of performing tasks like translation, summarization, and conversational interaction. This paper first explores advances in prompt engineering for LLMs as stand-alone pattern examples. It then presents a pattern language that extends the popular *Persona* pattern, which gives an LLM a role it uses to select what types of output to generate and what details to focus on. Earlier descriptions of the *Persona* pattern assigned static roles to LLMs to generate contextually appropriate responses, which is unduly limiting in more complex and dynamic scenarios.

This paper generalizes the *Persona* pattern to create a pattern language that contains the following four persona-related patterns: *Multi-Persona Interaction*, which allows LLMs to embody multiple roles simultaneously, providing richer insights from various perspectives; *Dynamic Persona Switching*, which enables seamless transitions between personas in response to evolving task requirements; *Role-Playing Scenarios*, which facilitate interactive and immersive learning experiences by simulating real-world situations; and *Contextual Depth Enhancement*, which enriches personas with detailed backgrounds, motivations, and constraints, ensuring more tailored and accurate responses.

Each pattern in the *Persona* pattern language enhances the realism, adaptability, and specificity of LLM interactions, enabling them to handle diverse and intricate tasks more effectively. The resulting pattern language provides a comprehensive framework that empowers users to harness the full potential of LLMs, thereby fostering more effective, refined, and reliable AI-driven communication and problem-solving.

Categories and Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces—*Natural language interfaces*; I.2.7 [Artificial Intelligence]: Natural Language Processing—*Language models*; D.2.11 [Software Engineering]: Software Architectures—*Patterns*; H.1.2 [Models and Principles]: User/Machine Systems—*Human-centered computing*

General Terms: Human Factors, Design, Algorithms, Experimentation, Verification

Additional Key Words and Phrases: Prompt engineering, persona-based interactions, language models, natural language processing, multi-persona interaction, dynamic persona switching, role-playing scenarios, contextual depth enhancement, generative AI, pattern languages, conversational AI, human-computer interaction

ACM Reference Format:

Schreiber W. and White J. and Schmidt D. 2024. Toward A Pattern Language for Persona-based Interactions with LLMs. HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. 22 (October 2015), 19 pages.

1. INTRODUCTION

Overview of prompt engineering and the *Persona* pattern. Prompt engineering [10] is a critical discipline for optimizing interactions between users and conversational large language models (LLMs), such as ChatGPT, Claude, Gemini, and DeepSeek. It involves crafting precise natural language instructions (called “prompts”) that guide LLMs to generate desired outputs. Prompt engineering is similar to conventional computer programming but utilizes natural language to harness the computational power of LLMs for various tasks.

Author’s emails: W. Schreiber, William.Schreiber@vanderbilt.edu, J. White, Jules.White@vanderbilt.edu, D. Schmidt, dcschmidt@wm.edu
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission. A preliminary version of this paper was presented in a writers’ workshop at the 30th Conference on Pattern Languages of Programs, People, and Practices (PLoP).
PLoP’24, October 13-16, Skamania Lodge, Columbia River Gorge, Washington, USA. Copyright 2024 is held by the author(s). HILLSIDE
978-1-941652-20-6

The *Persona* pattern was initially documented in [10], demonstrating its potential to instruct an LLM to adopt specific roles or identities. This powerful prompt engineering technique assigns personas, such as safety engineers, project managers, or even historical figures, so users can leverage an LLM's ability to simulate subject matter expertise when generating contextually relevant outputs. This pattern structures the LLM's focus and decision-making processes, aligning its responses closely with user objectives to enhance the utility and effectiveness of LLM interactions, especially in complex domains that require expert knowledge.

Limitations with the *Persona* pattern. Despite its advantages, the initial description of the *Persona* pattern [10] had several limitations that hindered its full potential. In particular, the static nature of personas restricted dynamic role transitions within a single session, complicating tasks that evolve and require different expertise sequentially. For instance, seamlessly shifting from the persona of a software developer to a project manager within the same LLM conversation often required re-initiating the persona definition process.

Moreover, the initial *Persona* pattern description lacked sufficient granularity, leading to broad and generic personas that often failed to capture the specific expertise and subtle understanding needed for specialized tasks. This shortfall can result in superficial or misaligned outputs. Likewise, the initial description did not adequately support multi-language and cultural adaptability, limiting its applicability in diverse linguistic and cultural contexts. Manually refining prompts to achieve desired outcomes can be inefficient, burdensome, and non-collaborative, particularly for users who lack detailed knowledge of the required inputs and outputs to an LLM.

Solution → A pattern language for applying personas in LLMs. To address these limitations, this paper extends the initial *Persona* pattern description and presents it as a pattern language [4]. We generalize the concept of pattern language (originally formulated by Christopher Alexander in the context of architectural design for buildings [1]) to the domain of prompt engineering and apply it to optimize interactions with LLMs. Appendix B provides further discussion of how we adapted the concept of pattern languages for prompt engineering.

A pattern language in the context of prompt engineering consists of a set of interrelated prompt patterns, each addressing specific challenges or tasks when interacting with LLMs to provide a structured framework for designing prompts that guide LLMs. The patterns presented in our *Persona* pattern language describe methods for

- Switching between personas dynamically,
- Enhancing persona customization,
- Supporting multi-language and cultural adaptability in personas, and
- Developing personas collaboratively.

This pattern language seeks to improve the utility, flexibility, and responsiveness of LLMs when handling complex, multi-faceted tasks, ultimately broadening their application and impact across various domains.

Paper organization. The remainder of this paper is organized as follows: Section 2 describes each of the patterns in the *Persona* pattern language; Section 3 summarizes related work on few-shot learning, chain-of-thought processing, prompt chaining, the Automatic Prompt Engineer framework, and ReAct; Section 4 presents concluding remarks and outlines our future work; Appendix A provides an overview of our prompt pattern form (which should be familiar to readers acquainted with classic software pattern form [6; 8]); and Appendix B gives an overview of pattern languages for prompt engineering.

2. PATTERNS IN THE *PERSONA* PATTERN LANGUAGE

This section describes our framework for enhancing prompt engineering practices [5] via the patterns comprising the *Persona* pattern language. The initial *Persona* pattern description [11] tailored the behavior and outputs of LLMs by instructing them to adopt specific roles or perspectives. By assigning a defined persona, such as a reliability engineer or a historian, prompt engineers can guide an LLM to generate more relevant, coherent, and contextually appropriate responses. This pattern leverages the extensive knowledge embedded within LLMs, aligning their outputs with the expectations and requirements of diverse user scenarios without requiring users to have the same in-depth knowledge.

However, the complexity and specificity of LLM-related tasks encountered in various domains necessitates further elaboration on the initial *Persona* pattern. This paper therefore introduces four patterns that form the *Persona* pattern language, which together enable LLMs to handle intricate LLM-related tasks with greater precision and contextual awareness. These patterns are summarized in Table 2 and described in the remainder of the section below.

Table 1: Summary of Prompt Patterns in the *Persona* Pattern Language.

Pattern Name	Prompt Description
<i>Multi-Persona Interaction</i>	Allows an LLM to adopt and integrate multiple personas simultaneously within a single interaction, enhancing the richness and depth of responses by drawing on diverse expertise and perspectives at once.
<i>Dynamic Persona Switching</i>	Enables an LLM to transition seamlessly between different personas within the same session of interactions based on evolving task requirements to increase the adaptability and relevance with an LLM by shifting its focus and expertise as the context changes.
<i>Role-Playing Scenarios</i>	Facilitates interactive and immersive learning experiences by simulating real-world situations to enhance learning, decision-making, and problem-solving by creating more immersive and interactive experiences.
<i>Contextual Depth Enhancement</i>	Enriches personas with detailed backgrounds, motivations, and constraints, ensuring more tailored and accurate responses, as well as increasing the realism and relevance of LLM outputs by grounding them in a well-defined context.

2.1 The *Multi-Persona Interaction* Pattern

Intent. Enable an LLM to adopt multiple personas simultaneously within a single interaction so the LLM can integrate insights from various roles.

Context. In complex tasks requiring diverse expertise, users may need an LLM to provide insights from different perspectives. Traditional single-persona interactions limit the depth and breadth of LLM responses, particularly in scenarios where multiple fields of expertise are needed.

Problem. How can users enable an LLM to provide comprehensive outputs that reflect multiple expert perspectives simultaneously within a single session?

Solution. Allow an LLM to embody multiple personas concurrently by defining and combining distinct roles within a prompt. This approach enables the LLM to consider various viewpoints and generate integrated responses that are rich in detail and context.

Key Ideas. Three fundamental contextual statements for the *Multi-Persona Interaction* pattern include the following:

- Personas definition.* Explicitly define each persona that the LLM will embody, including their roles, responsibilities, and perspectives (e.g., job descriptions, titles, and/or characters).
- Prompt design.* Create a prompt that instructs the LLM to act as multiple personas and provide outputs reflective of their combined expertise.
- Output integration.* Instruct an LLM to generate a cohesive response that integrates insights from all defined personas.

Implementation. Three fundamental implementations for the *Multi-Persona Interaction* pattern include the following:

- Persona specification.* Clearly define each persona's role and expertise within the prompt.
- Contextual prompts.* Provide contextual statements that guide an LLM to integrate the persona's insights.
- Output review.* Ensure the generated outputs are cohesive and reflect the nuances of each persona's perspective.

The *Multi-Persona Interaction* pattern allows users to leverage multiple expert perspectives simultaneously, addressing the challenge of not knowing exactly what details or outputs are crucial for a given task. This approach enables an LLM to deliver these responses by integrating diverse insights tailored to multifaceted tasks.

Implementing *Multi-Persona Interaction* involves creating prompts that clearly delineate the responsibilities and viewpoints of each persona. This delineation can be achieved by explicitly defining the roles within the prompt and instructing an LLM to generate responses that reflect the combined perspectives, thereby leveraging an LLM's ability to context-switch and integrate diverse knowledge bases seamlessly.

Case Study Example. Title. Collaborative Medical Diagnosis and Treatment Planning.

Context. In a virtual healthcare consultation, a clinician seeks LLM assistance for a comprehensive diagnosis and treatment plan. The clinician not only needs a standard diagnosis but also wants insights from various medical perspectives to ensure the best possible care. Therefore, the clinician prompts an LLM to embody multiple medical personas, thereby providing a holistic understanding of the patient's condition and suggest a multidisciplinary treatment approach.

Personas Involved.

- (1) *General Practitioner*. Provides a broad overview of the patient's symptoms, considering common illnesses and general health guidelines.
- (2) *Cardiologist*. Focuses on the patient's heart-related symptoms, interpreting data relevant to cardiac health.
- (3) *Endocrinologist*. Specializes in hormonal imbalances, providing insights into how symptoms might relate to endocrine disorders.
- (4) *Nutritionist*. Advises on dietary adjustments to support overall health and specific medical conditions.

Problem. How can an LLM be instructed to deliver comprehensive medical advice that reflects the expertise of multiple specialists, ensuring a well-rounded and accurate response that integrates diverse medical insights?

Solution. Implement the *Multi-Persona Interaction* pattern by clearly defining each medical persona's role and instructing an LLM to provide outputs that reflect the combined expertise of these personas. The LLM will generate a response that integrates the insights from all defined personas, offering a comprehensive diagnosis and a multidisciplinary treatment plan.

Prompt Design and Sample LLM Response. The prompt should explicitly state (1) the roles of each medical persona and (2) what specific insights they should contribute. For example:

—**User Prompt-1.** “Act as a General Practitioner, Cardiologist, Endocrinologist, and Nutritionist. Provide a comprehensive analysis of the patient's symptoms and suggest a multidisciplinary treatment plan. The General Practitioner will offer a general overview, the Cardiologist will assess any heart-related concerns, the Endocrinologist will look for hormonal imbalances, and the Nutritionist will provide dietary recommendations.”

—**User Prompt-2.** “I am experiencing symptoms such as fatigue, shortness of breath, and weight gain. I have a family history of heart disease and recent blood test results showing elevated blood sugar levels. Please provide a comprehensive diagnosis and a multidisciplinary treatment plan.”

Based on the prompt examples above, an LLM can generate the following cohesive response that reflects the combined insights of all personas, ensuring that the advice given is well-rounded and covers all aspects of the patient's health:

- (1) *General Practitioner*. “Based on the patient's symptoms of fatigue and shortness of breath, a general examination and a review of lifestyle factors such as diet, exercise, and stress levels are recommended. It is important to rule out any common conditions such as anemia or respiratory infections that could be contributing to these symptoms.”
- (2) *Cardiologist*. “Given the family history of heart disease and the symptom of shortness of breath, it is crucial to consider potential cardiac issues. I recommend an echocardiogram to assess heart function and look for signs of any underlying cardiac conditions that may need immediate attention.”

- (3) *Endocrinologist*. “The elevated blood sugar levels suggest a possible endocrine disorder, such as diabetes. It is essential to conduct further tests, including a fasting glucose test and HbA1c, to evaluate the patient’s blood sugar control over time and determine the appropriate intervention to manage this condition.”
- (4) *Nutritionist*. “With the symptoms of fatigue and weight gain, and considering the potential cardiac and endocrine concerns, a dietary plan focused on reducing refined sugars and saturated fats is recommended. Incorporating more whole grains, lean proteins, and a variety of fruits and vegetables can help manage blood sugar levels and support overall cardiovascular health.”

Outcome. By embodying all four medical personas, an LLM provides a more thorough diagnosis and treatment plan that integrates insights from each specialty. The output is tailored to the patient’s needs, addressing the immediate health concerns while also offering long-term lifestyle adjustments. This approach enhances the accuracy and relevance of the medical advice, demonstrating the effectiveness of the *Multi-Persona Interaction* pattern in assisting clinicians in handling complex, multidisciplinary tasks.

Consequences. The *Multi-Persona Interaction* pattern instructs an LLM to embody several personas concurrently. This pattern may yield the following positive and negative consequences that users should consider when deciding whether to apply it.

Positive consequences of applying this pattern include the following:

- Enhanced contextual relevance.* By adopting multiple personas, an LLM can generate outputs that reflect a broader range of expertise. This capability provides richer and more sophisticated responses, which are essential for tasks requiring diverse perspectives. For example, integrating insights from both a teacher and a peer student can yield comprehensive educational interactions that cater to both instructional and engagement needs.
- Improved user experience.* Users may benefit from more dynamic and contextually appropriate interactions. For instance, in a code review scenario, an LLM acting as a subject expert *and* a peer reviewer can provide both detailed technical feedback and accessible explanations, enhancing the overall experience.
- Flexibility and adaptability.* The ability to switch between personas or integrate multiple perspectives allows an LLM to adapt dynamically to evolving user needs and scenarios, offering a versatile tool for complex problem-solving.

Negative consequences of applying this pattern include the following:

- Increased complexity in prompt design.* Crafting prompts that effectively balance the contributions of each persona without overwhelming an LLM can be hard. Ensuring clarity and coherence in an LLM’s responses requires careful design and iterative refinement.
- Risk of hallucinations.* An LLM may make incorrect assumptions or generate fictitious content based on the personas it adopts. For example, when acting as a Linux terminal, it might produce imaginary file systems or synthetic data, potentially yielding misleading outputs [7].
- Consistency and coherence issues.* Maintaining consistent and coherent outputs across different personas is tricky since an LLM may struggle to integrate diverse perspectives seamlessly, leading to fragmented or disjointed responses.

Despite these negative consequences, the *Multi-Persona Interaction* pattern allows an LLM to adopt multiple personas simultaneously, thereby providing richer interactions and better outputs. This pattern is particularly useful for tasks that require diverse expertise and perspectives, allowing an LLM to integrate insights from various roles. It is also useful in scenarios where complex tasks require multiple perspectives or expertise.

See Also. The *Multi-Persona Interaction* pattern allows an LLM to embody multiple personas simultaneously within a single interaction. For example, in scenarios requiring diverse expertise, such as a software development task involving both a security expert and a project manager, an LLM can draw upon multiple personas to provide

insights from these distinct viewpoints at the same time to ensure that rich and diverse perspectives are captured, providing a comprehensive response to complex tasks.

In more dynamic scenarios, however, as a task evolves users may require an LLM to switch between personas based on the task's progression, rather than maintaining all personas concurrently. These scenarios motivate the need for the *Dynamic Persona Switching* pattern presented in Section 2.2. Essentially, *Dynamic Persona Switching* is an extension and evolution of *Multi-Persona Interaction*, where the initial multiple roles give way to a more focused, context-driven persona switching based on the specific requirements of each stage of the task, as described next.

2.2 The *Dynamic Persona Switching* Pattern

Intent. Allow an LLM to shift between different personas within a single session to adapt to changing user needs and task requirements.

Context. In dynamic tasks where the nature of the problem evolves, a single static persona may be insufficient. Users may therefore need an LLM to transition between roles, providing expertise at different stages of the task.

Problem. How can LLMs be instructed to adapt to evolving tasks by transitioning between different personas within the same interaction?

Solution. Instruct an LLM to begin with one persona and then switch to another as the task context changes, thereby allowing the LLM to provide relevant expertise at each stage of the interaction.

Key Ideas. Three fundamental contextual statements for the *Dynamic Persona Switching* pattern are:

- Initial persona.* Define the starting persona based on the initial task requirements and instruct an LLM to begin with a specific role.
- Contextual triggers.* Establish triggers within the task context that indicate when a persona switch is needed based on the evolving context of the task.
- Sequential personas.* Specify the sequence of personas an LLM should adopt as the task progresses.

Implementation. Three fundamental implementations for the *Dynamic Persona Switching* pattern are:

- Sequential prompts.* Design prompts that guide an LLM through the sequence of personas, ensuring smooth transitions.
- Trigger mechanisms.* Define clear conditions or cues that indicate when a persona switch should occur.
- Continuity management.* Ensure an LLM maintains coherence across persona transitions.

Dynamic Persona Switching enables LLMs to adapt to evolving task requirements by transitioning between different personas within a single session. This pattern addresses the limitation of static personas, allowing users to obtain expert-level outputs tailored to different stages of a task without re-initiating the persona definition process. This approach enables adaptive, context-aware interactions tailored to the dynamic needs of complex tasks.

Implementing *Dynamic Persona Switching* involves designing prompts that guide an LLM to change roles based on the progression of the task. This guidance can be achieved by specifying conditions or stages within the prompt that trigger the persona switch. The pattern leverages an LLM's ability to context-switch and maintain continuity in the conversation, ensuring that the transition between personas is seamless and coherent.

Case Study Example. Title. Cross-Disciplinary Scientific Research Assistant.

Context. A research team is working on a complex project that involves multiple scientific disciplines. They are exploring the impacts of climate change on marine biodiversity, requiring expertise from marine biology, climate science, and data analysis. The team seeks assistance from an LLM that dynamically switches personas to provide tailored, discipline-specific insights as the research progresses. The team begins their investigation with an initial

focus on marine biology, needing detailed information about various marine species and their ecosystems. As the study evolves, they require insights into climate patterns and the effects of global warming on these ecosystems. Finally, they need sophisticated data analysis to interpret their collected data and predict future trends.

Persona's Involved.

- (1) *Marine Biologist*. Provides expertise on marine species, their ecosystems, and biodiversity trends.
- (2) *Climate Scientist*. Focuses on climate patterns, ocean temperatures, and the effects of global warming.
- (3) *Data Analyst*. Interprets data and creates predictive models for ecosystem changes.

Problem. How can an LLM effectively adapt to the varying requirements of interdisciplinary research by adopting and switching between distinct scientific personas?

Solution. Leverage the *Dynamic Persona Switching* pattern to enable seamless transitions between relevant personas. This approach ensures the LLM provides domain-specific expertise at each stage of the research, aligning responses with the evolving needs of the project.

Prompt Design and Sample LLM Responses. The prompt should explicitly state (1) the roles of each medical persona and (2) what specific insights they should contribute. For example:

—**User Prompt-1.** "Act as a marine biologist. Provide detailed insights into coral reef ecosystems, focusing on species diversity and ecological roles."

—**LLM Response-1.** "As a marine biologist, coral reefs are among the most bio-diverse ecosystems in the ocean. Species such as parrot fish, clown fish, and various coral types play crucial roles in maintaining the ecological balance. Parrot fish help in controlling algae growth on corals, allowing them to thrive, while clown fish have symbiotic relationships with anemones, providing mutual protection."

—**Persona Switch Trigger.** Once marine ecosystem details are documented.

—**User Prompt-2.** "Switch to a climate scientist. Analyze the impact of ocean warming and acidification on coral reefs."

—**LLM Response-2.** "As a climate scientist, rising ocean temperatures lead to coral bleaching, a process where corals expel the symbiotic algae living in their tissues, causing them to turn white and often die if stressful conditions persist. Ocean acidification, driven by increased CO₂ absorption, reduces the availability of carbonate ions, crucial for coral skeleton formation, weakening the reef structure."

—**Persona Switch Trigger.** After understanding biological and climate factors

—**User Prompt-3.** "Switch to a data analyst. Using the provided data, predict coral reef health trends over the next 20 years."

—**LLM Response-3.** "As a data analyst, by applying regression models to the dataset of historical temperature changes and coral bleaching events, we can predict that if current trends continue, we could see a 40% decrease in healthy coral reefs globally over the next two decades. This prediction is based on the current rate of temperature increase and corresponding bleaching events observed over the past thirty years."

Outcome. Through dynamic persona switching, an LLM provides the research team with in-depth, domain-specific insights at each stage of their project. Starting as a marine biologist, transitioning to a climate scientist, and concluding as a data analyst, the LLM adapts its responses to fit the evolving needs of the research task, enhancing the accuracy and relevance of the assistance provided. This scenario demonstrates the *Dynamic Persona Switching* pattern's effectiveness in managing complex, multi-disciplinary tasks that require varied expertise at different stages. By allowing an LLM to switch personas based on task evolution, users can receive highly specialized guidance, optimizing their workflows and outcomes in dynamic research environments.

Consequences. The *Dynamic Persona Switching* pattern enables an LLM to transition between personas fluidly. This pattern may yield the following positive and negative consequences that users should consider when deciding whether to apply it.

Positive consequences of applying this pattern include the following:

- Enhanced adaptability.* This pattern ensures that an LLM can provide appropriate expertise throughout a task by transitioning between roles as needed. This flexibility is crucial for tasks that require different skill sets or perspectives at various stages, such as shifting from a security expert to a developer during a software assessment.
- Improved problem-solving.* Sequentially adopting different personas allows more comprehensive solutions from multiple perspectives. For example, beginning as a project manager to outline goals and transitioning to a data analyst to interpret data ensures thorough analysis and informed decision-making.
- Flexibility and adaptability.* Users may experience more interactive and responsive sessions as LLMs adapt dynamically to their changing needs and queries, enhancing satisfaction and effectiveness.

Negative consequences of applying this pattern include the following:

- Complex prompt design.* Designing prompts that manage transitions effectively without overwhelming an LLM or the user can be hard. It requires clear planning to ensure smooth transitions between personas and coherent responses.
- Risk of hallucinations.* As with maintaining multiple personas, transitioning between them could increase the risk of an LLM generating fictional or inaccurate information, particularly if triggers for switching are not clearly established. [7]
- Potential for inconsistent outputs.* Maintaining a consistent and coherent output across different personas can be hard since an LLM may struggle to integrate diverse perspectives seamlessly, yielding fragmented or disjointed responses.

Despite these negative consequences, implementing *Dynamic Persona Switching* within a single session can help address different aspects of a problem as it evolves. This pattern extends the capabilities of LLMs to handle complex, evolving tasks by adapting to changing user needs and task requirements dynamically.

See Also. While the *Multi-Persona Interaction* pattern presented in Section 2.1 establishes a *static* interaction where multiple personas contribute simultaneously, the *Dynamic Persona Switching* pattern adapts this foundation to situations where evolving contexts necessitate switching between personas *dynamically*. For instance, in a lengthy process or prompt chain users may first need insights from a project manager but later require more technical analysis from a software developer. The initial multi-persona setup provides the flexibility of starting with different roles, but as the task progresses only one active persona may be needed at a time, shifting dynamically to the persona most relevant to that phase of the task.

2.3 The *Role-Playing Scenarios* Pattern

Intent. Enable LLMs to simulate interactive, scenario-based role-playing exercises to enhance learning, decision-making, and problem-solving by placing users in realistic and immersive situations.

Context. In educational, training, or professional environments, users often need to apply theoretical knowledge in practical scenarios. Traditional interactions with LLMs might lack the depth and realism needed to fully engage users or simulate the complexities of real-world tasks.

Problem. How can LLMs be instructed to create realistic and immersive interactions that allow users to practice skills and apply knowledge in practical, scenario-based contexts?

Solution. The *Role-Playing Scenarios* pattern instructs an LLM to engage in role-playing within a specified scenario. By assuming specific roles (such as mentor, customer, or expert), the LLM generates contextually relevant and interactive outputs that simulate real-world situations. This pattern helps users better understand concepts, improve decision-making, and enhance problem-solving skills through practical application.

Key Ideas. Four fundamental contextual statements for the *Role-Playing Scenarios* pattern are the following:

- Persona definition.* Define the role(s) an LLM will adopt, including detailed descriptions of the responsibilities, knowledge, and behavior expected from each role.
- Scenario context.* Establish the context or setting of the role-playing exercise, including the objectives, challenges, and any specific conditions or constraints.
- Interactive prompts.* Develop prompts that guide an LLM through the scenario, encouraging it to generate responses that are relevant to the context and aligned with the defined roles.
- User interaction.* Allow the user to interact with an LLM, asking questions, making decisions, or responding to the scenario as it unfolds.

Implementation. Three fundamental implementations for the *Role-Playing Scenarios* pattern are the following:

- Role and context specification.* Clearly define the roles that an LLM will adopt, including any specific characteristics or expertise required for the scenario. Likewise, establish the scenario context, outlining the objectives and key challenges.
- Scenario design.* Craft prompts that set the stage for the interaction, providing an LLM with the necessary background information and guiding the user through the scenario.
- Iterative refinement.* Leverage user feedback to refine the prompts and improve the realism and effectiveness of the scenario.

Role-Playing Scenarios enhance the persona pattern by creating immersive, scenario-based interactions that simulate real-world situations. Implementing these types of scenarios involves crafting prompts that define the roles and the context of the interaction. These prompts must clearly outline the objectives, the personas involved, and the expected interactions. An LLM can then generate responses that reflect the depth and expertise required by each role, creating a dynamic and immersive experience.

Case Study Example. Title. Emergency Response Training Simulation.

Context. In a virtual emergency response training environment, a group of trainees is being prepared for handling high-pressure situations, such as natural disasters, terrorist attacks, and large-scale accidents. The goal is to improve decision-making skills, enhance situational awareness, and foster effective communication among team members. An LLM is used to simulate different roles within the scenario, providing real-time feedback and guidance based on the trainees' actions and decisions. The simulation begins with an LLM assuming the roles of various key figures in a disaster response situation, including an Incident Commander, a Medical Officer, and a Public Information Officer. Each persona has specific responsibilities and provides tailored information to the trainees based on the evolving scenario. The simulation starts with a simulated earthquake in a densely populated urban area, requiring immediate coordination and response from multiple agencies.

Personas Involved.

- (1) *Incident Commander.* Oversees the emergency response, makes strategic decisions, and ensures effective coordination.
- (2) *Medical Officer.* Manages medical resources, prioritizes casualties, and directs treatment and evacuation efforts.
- (3) *Public Information Officer.* Communicates with the public and media, ensuring accurate dissemination of information to prevent panic.

Problem. How can an LLM simulate an interactive training environment that replicates the complexity of emergency response, providing participants with realistic, role-based guidance and feedback?

Solution. Implement the *Role-Playing Scenarios* pattern to simulate a multi-role emergency response environment. This pattern enables the LLM to adopt specific roles dynamically, facilitating immersive, scenario-based interactions that enhance the training experience.

Prompt Design and Sample LLM Responses. The prompt should explicitly state (1) the roles of each persona and (2) what specific insights they should contribute. For example:

- Role Description 1.** The *Incident Commander* oversees the entire emergency response operation, making strategic decisions, allocating resources, and ensuring effective communication among all involved parties.
- Role Description 2.** The *Medical Officer* is responsible for managing medical response efforts, including triage, treatment, and evacuation of casualties, as well as provide real-time updates on the status of medical teams and resource needs.
- Role Description 3.** The *Public Information Officer* manages communications with the media and the public, ensuring that accurate and timely information is disseminated to prevent misinformation and panic.
- User Prompt-1.** "Act as the Incident Commander. Provide an initial assessment of a simulated earthquake scenario, including resource allocation and public safety instructions."
- LLM Response-1.** "A 7.5 magnitude earthquake has struck downtown. Allocate search and rescue teams to sectors with reported structural collapses. Establish a command post and prioritize public safety announcements to reduce panic."
- User Prompt-2.** "Switch to the Medical Officer persona. Provide an update on the status of medical teams and advise on prioritizing casualties for treatment and evacuation."
- LLM Response-2.** "As the Medical Officer, we've set up a triage area near the command post. Team Alpha is treating minor injuries; Team Bravo is handling critical cases. We need more supplies for treating fractures and burns. Prioritize evacuating the critically injured from Sector 4, where the hospital has sustained damage and is at capacity."
- User Prompt-3.** "Switch to the Public Information Officer persona. Craft a public statement to inform the community about the ongoing emergency response efforts and provide safety instructions."
- LLM Response-3.** "As the Public Information Officer, our priority is to keep the public informed. Here's the statement: We are currently responding to a major earthquake in the downtown area. Emergency services are on the scene conducting search and rescue operations. We urge all residents to avoid affected areas, stay indoors if safe, and follow instructions from local authorities. Updates will be provided as more information becomes available."

Outcome. The *Role-Playing Scenarios* pattern enables an LLM to simulate a realistic emergency response training environment. By adopting different roles and providing tailored responses, the LLM helps trainees practice decision-making and communication skills in a controlled, yet immersive setting. Trainees can engage with the simulation by asking questions, making decisions, and receiving immediate feedback, enhancing their preparedness for real-world emergencies.

This example demonstrates the versatility and effectiveness of the *Role-Playing Scenarios* pattern in creating dynamic, context-rich training environments that simulate real-world challenges. It allows users to practice and refine their skills in a safe and controlled setting, preparing them for high-pressure situations they may encounter in their professional roles.

Consequences. The *Role-Playing Scenarios* enables an LLM to adopt and switch between multiple roles within a structured scenario, thereby facilitating a more interactive and engaging learning or problem-solving environment. This pattern may yield the following positive and negative consequences that users should consider when deciding whether to apply it.

Positive consequences of applying this pattern include the following:

- Enhanced learning and engagement.** By providing an immersive, interactive experience, role-playing scenarios can significantly enhance learning and engagement. Users can practice skills and apply knowledge in practical, contextually rich environments, fostering deeper understanding and retention.

- Realistic feedback.* The interactive nature of role-playing allows users to receive immediate, realistic feedback based on their decisions and actions. This hands-on approach helps reinforce learning and improve problem-solving skills, making theoretical knowledge more applicable.
- Customizable experiences.* Role-playing scenarios can be tailored to specific needs and contexts, making interactions more relevant and beneficial. For instance, users can specify the type of code review, or the nature of the security threat being simulated, allowing for focused learning or assessment.

Negative consequences of applying this pattern include the following:

- Complexity in scenario design.* Designing realistic and engaging scenarios requires significant effort and expertise, especially in defining roles and contexts. Ensuring that these scenarios are well-structured and maintain logical progression can be hard.
- Risk of hallucinations.* An LLM might generate inaccurate or unrealistic responses, particularly if the scenario is not well-defined or if the prompts are vague, which can lead to confusion or misinformation [7].
- Consistency challenges.* Maintaining a coherent narrative and logical progression throughout the scenario can be hard, particularly in complex or dynamic scenarios.

Despite these negative consequences, incorporating *Role-Playing Scenarios* can make interactions more engaging and realistic, particularly in educational or training contexts. This pattern not only enhances the realism and engagement of interactions but also deepens users' understanding of complex concepts through practical application.

See Also. The *Multi-Persona Interaction* pattern presented in Section 2.1 introduced the ability for an LLM to embody multiple personas simultaneously. This interaction is essential in contexts where tasks require multiple areas of expertise to be addressed concurrently, such as business strategy or multidisciplinary medical assessments. In contrast, the *Dynamic Persona Switching* pattern presented in Section 2.2 allows an LLM to shift between different personas depending on the evolving needs of a task. Rather than requiring simultaneous engagement from multiple personas, the LLM can dynamically adapt as new contexts or stages of the problem arise.

The *Role-Playing Scenarios* pattern builds on these ideas but incorporates a more structured, immersive interaction. In this pattern, an LLM does not simply provide multiple perspectives or switch dynamically between personas. Instead, it engages users in interactive simulations or scenarios that closely mirror real-world situations, which makes the LLM not just a source of expert knowledge but an active participant in simulated environments. Personas are not thus just consulted in this pattern but are involved in a role-playing exercise where their behaviors, decisions, and actions evolve based on the context and user inputs.

Prompt chaining sequentially links multiple LLM prompts, decomposing them into discrete, interdependent sub-tasks, as described in Section 3. Each sub-task is addressed through a distinct LLM prompt, and the output of one step serves as the input to the next. In *Role-Playing Scenarios*, prompt chaining can play a crucial role because it allows a controlled progression of the interaction. As the user progresses through the scenario, each stage can be broken down into simpler tasks, enabling deeper engagement with each persona. It can thus allow an LLM to adapt to the evolving needs of the scenario, guiding the user through a dynamic, responsive narrative where different personas can be applied as required.

2.4 The Contextual Depth Enhancement Pattern

Intent. Enhance the realism, relevance, and specificity of interactions generated by an LLM by adding rich layers of context to personas by specifying detailed backgrounds, motivations, and constraints for the personas.

Context. The effectiveness of LLM responses often hinges on the context provided by users. In many cases, however, users may be unaware of the specific details or context needed for LLMs to generate the most relevant and precise outputs. Their lack of awareness is particularly problematic in specialized fields, such as finance, law, or healthcare, where deep expertise and nuanced understanding are essential.

Problem. How can LLMs be instructed to generate outputs that are not only accurate but also deeply informed by the specific context in which they are applied?

Solution. Enhance the traditional *Persona* pattern by incorporating rich layers of context, including detailed backgrounds, motivations, and constraints, to guide an LLM's responses.

Key Ideas. Two fundamental contextual statements for the *Contextual Depth Enhancement* pattern are the following:

- Act as persona X with specific background Y.* This statement instructs an LLM to adopt a persona enriched with detailed background information. For example, the persona could be a financial analyst with a decade of experience in the tech industry.
- Reflect persona X's expertise and constraints.* This statement directs an LLM to generate responses that are informed by the persona's background, motivations, and constraints. The LLM should consider these factors in its outputs, ensuring that they are contextually accurate and relevant.

Implementation. Four fundamental implementations for the *Contextual Depth Enhancement* pattern are:

- Define the persona.* Create a persona that includes detailed background information relevant to the task. For example, "Act as a financial analyst with ten years of experience in the high-tech industry."
- Specify contextual factors.* Include motivations, constraints, and other contextual factors that may influence the persona's perspective. For example, "You are motivated by risk aversion and constrained by limited financial data."
- Craft the prompt.* Develop a prompt that clearly instructs an LLM to generate outputs based on the enriched persona. Ensure the prompt is clear and concise, providing all necessary context without overwhelming the LLM.
- Test and iterate.* Run an LLM with the crafted prompt, review the outputs, and refine the persona and prompt as needed to improve accuracy and relevance.

While the traditional *Persona* pattern is effective in guiding LLM responses, it lacks the depth needed to address complex tasks that require detailed contextual understanding. In contrast, the *Contextual Depth Enhancement* pattern ensures LLM outputs are contextually accurate and nuanced, reflecting a deeper understanding of the persona's expertise and the specific context of the task.

Implementing *Contextual Depth Enhancement* involves crafting prompts that include extensive details about the persona's background and context. These prompts should specify the persona's expertise, their current task, and any relevant constraints or motivations. This pattern leverages the capacity of LLMs to integrate and reflect on these details, generating outputs that are not only relevant but also better informed by the persona's contextual framework.

Case Study Example. Title. Strategic Business Decision-Making in a Global Context.

Context. A multinational corporation is considering entering a new market in Asia. The company's board of directors seeks comprehensive advice that integrates insights from economic trends, cultural nuances, political environments, and competitive landscapes. The board wants to ensure that the decision-making process is informed by rich, contextual knowledge specific to the region, simulating a strategic business analyst with deep expertise in global markets, including detailed backgrounds, motivations, and constraints.

Personas Involved.

- (1) *Economic Analyst.* Assesses market viability through economic indicators and trends.
- (2) *Cultural Strategist.* Explores cultural influences on consumer behavior and brand reception.
- (3) *Political Advisor.* Examines geopolitical and regulatory risks impacting business operations.
- (4) *Competitive Intelligence Specialist.* Evaluates market competition and potential alliances.

Problem. How can an LLM assist in delivering a comprehensive, multidisciplinary analysis to guide the corporation's strategic decision-making process, ensuring a nuanced understanding of the region's complex dynamics?

Solution. Apply the *Contextual Depth Enhancement* Pattern to enable the LLM to embody multiple personas simultaneously, ensuring an integrated response. Each persona is defined explicitly, with instructions to analyze distinct aspects of the market, producing a unified strategic recommendation.

Prompt Design and Sample LLM Responses. The prompt should explicitly state (1) the persona definition and (2) the contextual details. For example:

—**Persona definition.** The LLM is instructed to act as a “Strategic Business Analyst” with 15 years of experience in Asian markets, specializing in economic forecasting, cultural analysis, and geopolitical strategy.

—**Contextual details.** The persona is motivated by long-term profitability and risk management, constrained by ethical business practices and the need to adhere to international trade regulations. The analyst's background includes navigating complex trade agreements and understanding consumer behavior in diverse cultural settings.

—**User Prompt-1.** “Act as a Strategic Business Analyst with 15 years of experience in Asian markets. Provide a comprehensive analysis of the potential challenges and opportunities for entering the Southeast Asian market, considering economic trends, cultural dynamics, political stability, and competitive environment. Emphasize long-term profitability and risk management while adhering to ethical standards and international trade regulations.”

—**LLM Response-1.**

—**Economic trends.** “Based on current economic forecasts, Southeast Asia shows promising growth prospects, driven by increasing consumer spending and a young demographic. However, potential risks include currency volatility and dependence on exports, particularly in economies like Thailand and Malaysia. A cautious entry strategy that includes currency hedging and diversified investments could mitigate these risks.”

—**Cultural dynamics.** “Understanding local cultures is critical for success in Southeast Asia. Consumer preferences can vary widely across the region, with significant differences in brand loyalty, price sensitivity, and purchasing behavior. It's important to tailor marketing strategies to reflect local customs and values. For instance, emphasizing community and family-oriented messaging may resonate more in countries like Indonesia and the Philippines.”

—**Political stability.** “Political environments in Southeast Asia are generally stable but require careful navigation. Recent developments, such as changing trade policies in Vietnam and ongoing tensions in the South China Sea, could impact market entry strategies. Building strong relationships with local stakeholders and staying informed about political developments will be essential to maintain a competitive edge.”

—**Competitive environment.** “The competitive landscape in Southeast Asia is diverse, with both established local players and international firms. A thorough analysis of competitors' strengths and weaknesses, along with an assessment of potential partnership opportunities, could provide a strategic advantage. Collaborating with local companies could facilitate market entry and enhance brand credibility.”

Outcome. By incorporating detailed background information and specific motivations and constraints, an LLM provides a refined, correspondingly rich analysis that addresses the complex challenges and opportunities of entering a new market. The output demonstrates a deep understanding of the multifaceted nature of strategic business decisions in a global context, highlighting the effectiveness of the *Contextual Depth Enhancement* pattern in guiding LLMs to produce highly relevant and informed responses.

This scenario illustrates how *Contextual Depth Enhancement* can be applied to provide comprehensive and nuanced insights in complex decision-making processes. By enriching an LLM's persona with detailed contextual information, users can receive more tailored and precise guidance that aligns closely with specific goals and challenges, enhancing the overall effectiveness of LLM-driven interactions.

Consequences. The *Contextual Depth Enhancement* enriches the persona with detailed attributes, such as professional experience, personal motivations, and operational constraints, that should be considered. This pattern

may yield the following positive and negative consequences that users should consider when deciding whether to apply it.

Positive consequences of applying this pattern include the following:

- Increased realism and relevance.* By providing detailed backgrounds, motivations, and constraints for personas, this pattern results in more realistic and contextually accurate outputs. It allows users to receive responses highly relevant to their specific needs and tasks.
- Enhanced expertise simulation.* Enriching personas with specific attributes enable an LLM to simulate expert-level understanding and deliver nuanced insights. For example, a financial analyst persona with extensive experience can provide detailed financial analyses, aiding in better decision-making.
- Better customization and precision.* Users can fine-tune an LLM's responses by specifying detailed persona attributes, perhaps leading to more precise and tailored outputs that closely align with the task requirements.

Negative consequences of applying this pattern include the following:

- Complexity in prompt creation.* Crafting prompts that incorporate detailed contextual information effectively can be hard and time-consuming. Users must carefully construct these prompts to ensure an LLM accurately reflects the intended persona's depth.
- Risk of hallucinations.* An LLM may generate fictional or inaccurate details based on the enriched context, leading to potential misinformation. For example, the LLM might invent historical events or data that do not exist, compromising the reliability of its outputs [7].
- Inconsistency and coherence issues.* Maintaining consistency across detailed persona attributes can be hard. If the contextual depth is not well-integrated or the prompts are not structured coherently, an LLM might produce disjoint or conflicting information.

Despite these negative consequences, the *Contextual Depth Enhancement* pattern adds layers of context to personas that can improve the realism and relevance of the outputs by specifying detailed backgrounds, motivations, and constraints for the personas.

See Also. The *Contextual Depth Enhancement* pattern enriches personas with detailed backgrounds, motivations, and constraints. In contrast to the previous three patterns presented above, this pattern focuses on providing a deeper layer of realism. An LLM not only assumes a persona but also brings in situational, historical, and even psychological context, ensuring that its outputs are tailored precisely.

The *Contextual Depth Enhancement* pattern is the culmination of the evolution started with *Multi-Persona Interaction*, *Dynamic Persona Switching*, and *Role-Playing Scenarios*. It adds a necessary layer of realism and depth to personas and enables LLMs to generate more tailored, contextually aware outputs. Incorporating prompt chaining across all four of these patterns provides a method for users to build complexity into their interactions progressively. Rather than overwhelming an LLM with one large, multifaceted task, users can guide the interaction through a series of smaller, manageable prompts that build on one another.

3. RELATED WORK

Prompt engineering has become a pivotal area of research on optimizing interactions between users and LLMs. This discipline involves crafting precise natural language instructions, or “prompts,” to guide LLMs in generating desired outputs. Several advanced techniques have emerged to enhance the performance and versatility of LLMs, including few-shot learning, chain-of-thought processing, prompt chaining, the Automatic Prompt Engineer (APE) framework, and React. Each approach offers unique strengths and addresses different challenges associated with optimizing LLMs, as discussed in this section.

Few-shot learning. Few-shot learning is designed for rapid adaptation to new tasks by providing an LLM with several examples within the prompt. This technique leverages an LLM's ability to generalize from limited data points, enabling it to perform tasks with minimal supervision. Brown et al. [2] demonstrated its effectiveness in GPT-3,

showing that this LLM could achieve high accuracy across a range of tasks with just a few examples. Few-shot learning's performance can be sensitive to the provided examples, however, and it may struggle with tasks requiring complex reasoning. Despite these limitations, it remains a foundational approach in prompt engineering due to its simplicity and rapid adaptability.

Chain-of-thought processing enhances LLMs' reasoning capabilities by instructing them to break down problems into intermediate steps. Introduced by Wei et al. [9], this method encourages LLMs to articulate their reasoning processes, leading to more transparent and accurate outputs. This approach is particularly beneficial for tasks requiring logical reasoning and multi-step problem-solving. While chain-of-thought processing improves the interpretability of LLM outputs, it also increases prompt complexity and computational costs. Nevertheless, it represents a significant advancement in enabling LLMs to handle complex reasoning tasks effectively.

Prompt chaining [12] is a related prompt engineering technique that guides an LLM through a series of related tasks or queries. This method decomposes complex prompt sequences into simpler, more manageable steps. It connects multiple prompts in a logical order, where the output of one prompt serves as the input for the next, enabling it to handle tasks that require multi-step reasoning, decision-making, or a process that unfolds over time.

In contrast to "Chain-of-thought" processing, prompt chaining organizes interactions across multiple prompts, where the output of one serves as the input for the next. This approach enables users to decompose large, multifaceted tasks into smaller, more manageable steps, ensuring clarity and precision at each stage. Prompt chaining can be seamlessly integrated into the *Persona* pattern language for persona-based interactions with LLMs leveraging the language's responses in the right environment, thereby producing more accurate and understandable results.

The **Automatic Prompt Engineer (APE)** framework automates and optimizes prompt generation using machine learning algorithms. Proposed in [14], APE iteratively refines prompts based on performance metrics, leveraging reinforcement learning and evolutionary algorithms to discover effective prompt configurations. This automation saves time and resources, uncovering novel prompts that might not be intuitive to human designers. However, APE's computational demands can be substantial and its effectiveness depends on the quality of performance metrics. Despite these challenges, APE shows promise in enhancing prompt engineering through automation.

React, proposed by Yao et al. [13], ensures real-time adaptability and responsiveness by integrating real-time feedback into the prompt optimization process. It continuously monitors model performance and incorporates user feedback to adjust prompts dynamically, making it particularly useful for interactive applications where context and requirements change rapidly. React promotes a user-centric approach, ensuring prompts remain relevant and effective in dynamic environments. However, implementing React requires robust mechanisms for real-time monitoring and feedback integration, which can be complex and resource-intensive. Despite these challenges, React provides a powerful framework for developing highly responsive and effective prompts.

4. CONCLUDING REMARKS

This paper described four patterns comprising the *Persona* pattern language, which enhances the adaptability, depth, and versatility of LLM interactions and unlocks new potential for LLMs in domains like healthcare, education, business, and beyond. For example, in the healthcare domain, *Dynamic Persona Switching* can enable the seamless transition from a general practitioner to a specialist within a single interaction. Likewise, in the education domain *Role-Playing Scenarios* can create immersive learning experiences by simulating real-world scenarios. Moreover, in the business domain, *Multi-Persona Interaction* allows more comprehensive analyses by integrating multiple expert viewpoints.

The following are a summary of lessons learned from our documentation and application of the *Persona* pattern language:

- Refined interaction quality.* The *Persona* pattern language helps refine the interaction quality between users and LLMs, ensuring that LLMs can successfully navigate complex, multi-faceted tasks that require subtly refined understanding and adaptive responses.

- Enhanced effectiveness and adaptability.* Documenting *Persona* as a pattern language helps enhance the effectiveness, adaptability, and user-friendliness of LLM interactions, ultimately improving the capabilities of AI-driven communication and problem-solving. By moving beyond traditional static roles, these patterns offer clear, adaptable, and contextually rich collaborations.
- Improved realism and engagement.* Incorporating multiple personas, dynamic role transitions, and detailed contextual backgrounds can improve the realism and engagement of LLM outputs. These enhancements are particularly valuable in domains requiring specialized knowledge and adaptive responses, such as education, healthcare, business, and customer service. Moreover, the ability to co-create personas with users fosters a more personalized and precise communication.
- Increased functionality through pattern application.* Applying the *Persona* pattern language as a prompt engineering aid enhances the functionality and utility of LLMs. By addressing the identified challenges and continuing to innovate in these areas, we can unlock new potentials for LLMs, making them even more indispensable tools across a wide range of applications.
- Challenges and ongoing research needs.* While the benefits of the *Persona* pattern language are substantial, they also present challenges. Ensuring consistency and coherence across dynamic interactions, managing the complexity of persona development, handling cyber-security concerns, and addressing ethical considerations are critical areas that require ongoing research.

As LLMs continue to evolve, the development and refinement of the *Persona* pattern language present numerous opportunities for future research. Our future work focuses on expanding the repertoire of patterns, refining existing ones, and exploring automated tools for pattern application and customization. The following are future work activities we plan to conduct to enhance the adaptability, effectiveness, and user-centricity of LLM interactions:

- LLM output consistency between different models.* Different LLMs have different criteria based on the way they are designed, guardrails that have been put into place, and built in biases. The same prompts run on different LLMs may give different results. Research should focus on if and why LLMs might give varying answers to the same question.
- Output generation accuracy and consistency.* When an LLM is instructed through prompts to take on the role of an expert persona, delivering accurate, correct information consistently is critical especially in industries like healthcare. Research should thus focus on verifying through actual experts the veracity of the information that is output by the LLM.
- Contextual consistency and coherence.* Ensuring that LLMs maintain contextual consistency across interactions, especially in *Dynamic Persona Switching*, is crucial. Research should therefore focus on developing algorithms that track and integrate context seamlessly, allowing LLMs to switch personas without losing coherence or shifting out of context.
- Automated persona generation and refinement.* Future work should explore automated tools for generating and refining personas. By leveraging advanced natural language processing (NLP) techniques, these tools can dynamically create detailed personas based on user input and context, reducing the need for manual prompt engineering.
- Multi-domain adaptation.* Extending the *Persona* pattern language across multiple domains and integrating domain-specific knowledge bases can enhance LLM versatility. Research in this area could develop methods for dynamically adapting personas to different fields, such as healthcare, education, and business, based on the task at hand.
- Interactive feedback mechanisms.* Enhancing *Collaborative Persona Development* through sophisticated feedback loops can make the process more intuitive and effective. Investigating user-friendly interfaces and real-time feedback mechanisms will help users co-create more accurate and detailed personas.

- Cultural and temporal sensitivity.* Research should continue to delving into multi-language, cultural adaptation and temporal perspectives. Developing robust datasets and LLMs that accurately reflect diverse cultural norms and historical contexts will make LLMs more globally applicable and historically informed.
- Ethical and bias mitigation.* Ensuring that extended *Persona* patterns do not perpetuate biases or ethical concerns is critical. Future research must focus on developing ethical guidelines and bias mitigation strategies to ensure responsible and fair use of LLMs.
- Security and Criminal Considerations.* As LLMs become more advanced and versatile, they may be exploited for malicious purposes, such as generating phishing schemes, creating deepfake content, or aiding in cyber-attacks. Future work should focus on developing robust safeguards to prevent the misuse of LLMs in criminal activities. This work includes implementing ethical guidelines, monitoring for suspicious behavior, and enhancing security protocols to detect and mitigate the potential for abuse.

Acknowledgments

We would like to give special thanks and appreciation to our PLoP shepherd Kyle Brown, for his incredibly helpful feedback and insights which greatly improved the form and content of this paper. Likewise, the prompt patterns outlined in this paper have been designed to apply robustly across different LLMs. To showcase their utility and practicality, however, these patterns have been exclusively tested using OpenAI's GPT-4.

APPENDIX

A. OVERVIEW OF PROMPT PATTERN FORM

The four prompt patterns presented in this paper are documented using a form similar to classic software patterns [6; 8], with analogous versions of the **Name**, **Intent**, **Context**, **Problem**, **Solution**, **Key Ideas**, **Implementation**, **Case Study Example**, **Consequences**, and **See Also** sections in those pattern descriptions. Each section of our prompt pattern form is outlined briefly below. Together, this pattern form provides a framework for understanding, designing, and implementing patterns in persona-based interactions with LLMs.

- Name** – A unique identifier for referring to the prompt pattern.
- Intent** – Summarizes the core purpose or goal of the pattern. It defines what the pattern aims to achieve and provides a high-level overview of how it addresses a specific challenge in the interaction between users and LLMs.
- Context** – Outlines the circumstances or environment in which the pattern is applicable. It includes the background information, prerequisites, and conditions that must be met for the pattern to be effective.
- Problem** – Presents a concise, clearly articulated, specific issue or challenge that the pattern seeks to address in the form of a question.
- Solution** – Provides a detailed description of how the pattern addresses the identified problem. It outlines the steps, strategies, and mechanisms that should be implemented to achieve the pattern's Intent.
- Key Ideas** – Breaks down the pattern into its fundamental components and concepts. It defines the key elements that make up the pattern, such as personas, roles, scenarios, and interactions, and explains how these elements interact to produce the desired outcomes.
- Implementation** – Offers practical advice on how to put the pattern into action. It includes guidelines, best practices, and potential challenges to consider when applying the pattern.
- Case Study Example** – Provides a case study showing how the pattern can be applied using an LLM.
- Consequences** – Discusses the potential outcomes of applying the pattern, both positive and negative. It explores the benefits and advantages of using the pattern, as well as the possible drawbacks, risks, and trade-offs.

—**See Also** — Explains which prompt pattern(s) are closely related to this pattern and outlines important differences.

Our prompt pattern form intentionally omits the **Known Uses** section. Unlike classic software patterns, which are well-documented and accessible through open-source repositories across the Internet, prompts for LLMs are generally not codified or systematically available to the public. This lack of codification and availability thus poses challenges in presenting our proposed patterns entirely in a conventional pattern form.

The use of prompts is still an emerging practice, and there is no centralized repository or widely recognized standard that documents their use in a systematic way. As a result, our work aims to fill this gap by identifying, formalizing, and sharing these prompt patterns based on our extensive prompt engineering experience. While we understand that this approach does not align fully with classic pattern forms, we hope to contribute to the foundation upon which future prompt patterns can be more formally validated and documented.

GPT-4's advanced natural language understanding capabilities make it an ideal candidate for demonstrating the effectiveness of our structured data extraction methods. We encourage readers to apply these prompt patterns with their favorite LLMs, such as ChatGPT-4, or other advanced LLMs available to them. Experimentation with different LLMs is not only welcome but encouraged, as it can provide further insights into the generalizability and adaptability of prompt patterns across different LLM architectures and training paradigms.

B. AN OVERVIEW OF PATTERN LANGUAGES FOR PROMPT ENGINEERING

Pattern languages in the context of software design have long been a cornerstone for structuring and solving complex problems [4] by codifying best practices that guide developers in creating robust and scalable systems. Pattern languages were first popularized by Christopher Alexander in the field of building architecture [1] and later adapted to software development [3]. They consist of interconnected patterns that solve specific architecture and design problems, and when used together, they guide the development of comprehensive and coherent systems.

In the domain of prompt engineering for LLMs, the application of pattern languages is equally vital since they provide a structured framework for crafting prompts that yield consistent, high-quality outputs from LLMs, ensuring that they are applied effectively across various tasks and contexts. A pattern language for prompt engineering is a structured collection of prompt patterns, each designed to solve a specific problem encountered when interacting with an LLM. These problems might include generating outputs that are too generic, failing to capture the nuances of a task, or not adapting well to the user's evolving needs. A pattern language helps mitigate these issues by providing a set of templates or guidelines that can be used to construct prompts tailored to the task at hand.

As with patterns in a pattern language for software development, the patterns in a prompt engineering pattern language are not standalone solutions but are interconnected. This inter-connectedness allows users to combine and adapt patterns to fit complex and dynamic tasks. For example, a pattern language might include patterns for generating code, conducting data analysis, or simulating conversations. Each pattern includes details on how to structure the prompt, what kind of responses to expect, and how to refine the prompt based on LLM outputs.

The strength of a pattern language for prompt engineering resides in its reusability and adaptability. After the patterns in a pattern language are documented, they can be reused in different contexts with minimal adjustments. For instance, a pattern language designed to guide an LLM in generating project plans can be adapted for different industries by modifying the specific terms and conditions within the prompts. This adaptability makes pattern languages powerful tools for prompt engineering since they can be applied to a wide range of domains without needing to start from scratch each time. Moreover, applying pattern languages in prompt engineering encourages consistency across different interactions with LLMs, which is crucial for maintaining the reliability of LLM outputs, especially in high-stakes domains (such as healthcare, law, and aerospace) where precision and accuracy are paramount.

As LLMs continue to evolve, the development of comprehensive pattern languages will become increasingly important. These languages need to incorporate patterns that address emerging challenges, such as multi-modal interactions, cultural and linguistic diversity, and the ethical considerations of using AI in sensitive domains.

Moreover, the integration of pattern languages with automated tools may revolutionize how users interact with LLMs. Tools that automatically generate or suggest patterns based on the user's input could make prompt engineering more accessible to non-experts, democratizing the use of LLMs across various fields.

As the field of prompt engineering continues to evolve, the development of more and better pattern languages will play a crucial role in harnessing the full potential of LLMs. This approach allows prompt engineers and users to leverage well-defined patterns to address complex, multifaceted tasks efficiently. Moreover, the inter-connectivity of patterns within a pattern language ensures a holistic approach to solving problems. In particular, patterns are not isolated solutions to common prompting tasks, but instead form an interconnected web that allows users to combine and adapt them to meet diverse and evolving needs enabling the creation of more sophisticated and context-aware interactions with LLMs.

REFERENCES

- Shlomo Angel and Nikos A Salingaros. Christopher Alexander's Architectural Insights and Limitations. *New Design Ideas*, 6(3):386–401, 2022.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-shot Learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.
- Frank Buschmann, Kevin Henney, and Douglas C Schmidt. *Pattern-Oriented Software Architecture Volume 4: A Pattern Language for Distributed Computing*. John Wiley & Sons, 2007.
- Frank Bushmann, Kevin Henney, and Douglas C Schmidt. *Pattern-Oriented Software Architecture Volume 5: On Patterns and Pattern Languages*. John Wiley & Sons, 2007.
- Michael Desmond and Michelle Brachman. Exploring Prompt Engineering Practices in the Enterprise. <https://arxiv.org/abs/2403.08950>, 2024.
- Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, 1995.
- Fang Liu, Yang Liu, Lin Shi, Houkun Huang, Ruifeng Wang, Zhen Yang, Li Zhang, Zhongqi Li, and Yuchi Ma. Exploring and Evaluating Hallucinations in LLM-Powered Code Generation, 2024.
- Douglas C Schmidt, Michael Stal, Hans Rohnert, and Frank Buschmann. *Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects*. John Wiley & Sons, 2013.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought Prompting Elicits Reasoning in Large Language Models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. In *Proceedings of the 30th Pattern Languages of Programming (PLOP) conference*, Allerton Park, IL, October 2023.
- Jules White, Sam Hays, Quchen Fu, Jesse Spencer-Smith, and Douglas C Schmidt. ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design. In *Generative AI for Effective Software Development*, pages 71–108, 2024.
- Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. Promptchainer: Chaining Large Language Model Prompts Through Visual Programming. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–10, 2022.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing Reasoning and Acting in Language Models. <https://arxiv.org/abs/2210.03629>, 2022.
- Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large Language Models Are Human-Level Prompt Engineers. <https://arxiv.org/abs/2211.01910>, 2023.