Ontogenia: Ontology Generation with Metacognitive Prompting in Large Language Models

Anna Sofia Lippolis[1,2], Miguel Ceriani[1], Sara Zuppiroli[1], and Andrea Giovanni Nuzzolese[1]
{annasofia.lippolis,miguel.ceriani,sara.zuppiroli,andrea.nuzzolese}@istc.cnr.it

[1] CNR - Institute of Cognitive Sciences and Technologies, Rome & Bologna, Italy
[2] Department of Philosophy and Communication Studies, University of Bologna, Bologna, Italy

Abstract. Recent advancements in Large Language Models (LLMs) have primarily focused on enhancing task-specific performances by experimenting with prompt design. Despite the proven effectiveness of Metacognitive Prompting (MP), its application in the field of ontology generation remains an uncharted territory. This study addresses this gap by exploring this prompting technique in supporting the ontology design process, particularly with GPT-4, where this strategy has demonstrated consistent superiority over conventional and more direct prompting methods in recent research. Our work employs a dataset of ontology competency questions translated into SPARQL-OWL queries. In our methodology, called Ontogenia, we investigate various types and stages of knowledge injection through MP while maintaining the principles and steps of the eXtreme Design methodology, an established protocol in ontology design. Finally, the quality and performance of the resulting ontologies are assessed using both standard ontology quality metrics and evaluation by an ontology expert. This research strives to pave the way for LLMs to contribute to ontology generation, ensuring the inclusion of human insight in the process.

Keywords: Ontology Engineering · Competency Questions · Large Language Models · Metacognitive Prompting

1 Introduction and Related Work

Ontology design, situated at the intersection of knowledge representation, logic, and computational linguistics, involves conceptualizing and formalizing knowledge networks for semantic technologies. Methodologies like eXtreme Design (XD) [1] offer structured frameworks for ontology engineers, yet their complexity demands substantial expertise and resources. Manual curation and validation of ontological elements are labor-intensive, suggesting a need for intelligent automation. Large Language Models (LLMs), being able to effectively perform various natural language processing goals, present a compelling solution. Recently, the Metacognitive Prompting (MP) technique [6], inspired by human introspective processes, encourages self-evaluation through the introduction of a series of steps, building and supposedly improving performance over other methods such as Chain of Thought. Derived from the field of cognitive psychology, metacognition concerns an individual's capacity to self-reflect and critically evaluate their cognitive processes [2, 5]. By adopting this approach, the LLM is able to identify its own inaccuracies and dynamically adjust reasoning strategies, leading to more precise answer generation, as shown by [7]. Starting from these studies, in this paper we define the Ontogenia methodology to explore the usage of LLMs for one of the most crucial and creative steps in ontology design methodologies: the actual specification and formalisation of an ontology (or module thereof) based on a specific set of requirements[3].

2 The Ontogenia methodology

The Ontogenia methodology outlines a concise yet comprehensive approach to ontology development through an iterative and incremental process.

[3] Data and code used for the work is available at this link: https://anonymous.4open.science/r/Ontogenia-CAE4/README.md

Domain and CQ definition The dataset under consideration is derived from a recognized gold-standard used for testing and benchmarking in the field of competency questions research and ontology querying, detailed in [3]. For this study, a specific use case has been selected: the African Wildlife Ontology. This choice facilitates modeling a sufficiently broad domain with respect to the others in the dataset while simultaneously being able to use domain-specific ontology design patterns. The chosen subset comprises 14 distinct competency questions, providing a potentially comprehensive starting point for analysis.

Ontology Design Patterns selection The definition of the Ontology Design Patterns to reuse starts from the list of Content Ontology Design Patterns in the Ontology Design Patterns website[4]. From these, eight have been selected by ontology experts for their relevance to the domain and included in a dataset to be dynamically inputted to the prompt: *AgentRole, AquaticResources, Classification, Climatic Zone, Collection Entity, PartOf, Linnaean Taxonomy, SpeciesEat*. Collectively, these ODPs can provide a comprehensive foundation for answering the targeted competency questions, enabling a thorough exploration of animal-related topics in a systematic and informed manner.

Procedure and prompt design The procedure design was crafted through an iterative process, with each phase incrementally tested to evaluate the outcomes. This design strategy aims to amalgamate the MP technique with the eXtreme Design methodology [1], which requires the use of pre-selected competency questions—a collaborative effort between the ontology design team and domain experts. Additionally, it involves the selection, reuse, and integration of specific Content Patterns. This iterative approach, coupled with constant testing and reassessment, has ensured the procedure's alignment with the initial requirements. To bridge any gaps in GPT's understanding of specific ontology features, these elements were explicitly incorporated into the procedure, enhancing its comprehensiveness and effectiveness.

The prompt design is meant to incorporate information about the procedure, eventual previous output, competency questions and patterns to be also added dynamically to the prompt on the basis of specific needs. A specification to not repeat itself and not send comments was added in order to refine the output. The resulting procedure is mapped to the MP five steps as shown in Table 1.

Evaluation measures definition The definition of the evaluation is twofold. On the one hand, it involves an ontology engineer expert that analyzes the produced ontologies in terms of essential requirements such as required classes and object properties and usage of restrictions. On the other hand, in order to complement the expert analysis, it involves the Ontometrics service[5] and the OOPS! Ontology Pitfall Scanner [4].

Testing We use GPT-4 Turbo (gpt-4-1106-preview[6]) as our backbone model, with greedy decoding. Because GPT has a token limit in output, we came up with a division of competency questions to be given one group at a time, and each time the previous output is provided in order not to have a repeating of classes and properties.

We conducted four trials to evaluate the effect of different inputs on ontology generation. Trial 1 used competency questions with a generic prompt, Trial 2 added ontology design patterns, Trial 3 involved only competency questions and MP, and Trial 4 combined competency questions with the prompting procedure and patterns. We tested both the original and thematically grouped questions by GPT. The total computation cost was $2.10. Experiment details are documented in a Github repository[7].

[4] `http://ontologydesignpatterns.org/wiki/Submissions:ContentOPs`
[5] `https://ontometrics.informatik.uni-rostock.de/ontologymetrics/`
[6] `https://openai.com/blog/new-models-and-developer-products-announced-at-devday`
[7] `https://anonymous.4open.science/r/Ontogenia-CAE4/ontology_design.log`

Table 1. Mapping between the Ontogenia methodology and the MP procedure.

MP stages	Ontogenia stages	Description
Comprehension clarification	1. Competency question understanding.	The LLM interprets the CQs, contextualizing them.
Preliminary judgement	2. Preliminary identification of the context. 3. Divide the competency question into subject, predicate, object, and predicate nominative.	Logical analysis supports class and property identification from CQs.
Critical evaluation	5. Starting from your knowledge, extend the ontology with these restrictions.	Reflect on CQs to add rules and restrictions, enhancing the model.
Decision confirmation	8. Confirm the final answer and explain the reasoning.	Justify the decision-making process.
Confidence assessment	9. Make a confidence evaluation and explanation, testing the ontology on instances.	Evaluate the process and test the model's correctness with specific instances.

3 Results and discussion

Table 2 shows the metrics obtained using the OntoMetrics service for each case considered in the experiment. It can be seen that in our experiment the adoption of MP favors a richer formalisation. This is also evident from Table 3 that shows usage of a set of different types of axiom types across the test cases. At the same time, it should be noted that there are important limitations common to all the cases, such as the absence of property hierarchical relationships.

The pitfalls found by OOPS! were also analysed. Some issues are common to all the cases, as the lack of annotations (P08) and inverse relationships (P13)[8]. All the cases except case 3 contain at least a property that is defined with more than one domain or range (P19). This happens when the LLM generates multiple times the definition of an object property, with somewhat different domain/range values. It seems that the "intent" of the LLM would be to define the property over the union of the referenced classes, mirroring a common beginner's error in RDFS modelling. In the cases using patterns (2 and 4) an "untyped class" (P34) and "different naming conventions" (P22) are found. These are both due to the erroneous of an object property (`hasPart`) imported from a pattern but used as it was a class. Furthermore, in Case 1, no disjoint axioms are used (P10) and there are a couple of properties missing explicit domain/range declaration (P11)[9]. The ontology obtained in Case 3 is the only one featuring an ontology element, the `Plant` class, unconnected from the rest of the ontology. Nevertheless, Case 3 is the one having less pitfalls.

For what concerns basic metrics, Ontometrics shows a larger number of axioms when a pattern is used, along with a higher number of classes and object properties. Data properties are instead a weaker point, despite their use having been specified in the procedure.

According to the qualitative analysis by the ontology expert, while the LLM successfully identifies necessary classes and relevant subclasses, the generated ontologies exhibit numerous intrinsic and domain-related issues. Particularly problematic is the pairing of classes and properties. Properties like `eats` often possess overly specific domains and ranges, leading to the creation of unrelated properties such as `eatsPlant`, `eatsAnimal`, and `eatsPlantPart`.

[8] The lack of license information (P41) is also common to all the cases, but this is not an information to be expected from the LLM

[9] Domain and range of those two properties are actually in part inferable because, errors aside, they are meant to be defined in relationship to other properties.

While simple restrictions in class definitions are generally correct, the classification of animals by diet consistently falls short. This shortfall is partly due to the ambiguity of terms like “carnivore” in biological contexts, where strict logical constraints are challenging to establish. This highlights the necessity for further research into the collaboration between ontology design teams and LLMs, opening avenues for exploring new directions.

Ontometrics	**Case1 No pattern No MP**	**Case2 Pattern No MP**	**Case3 No pattern MP**	**Case4 Pattern MP**	**Reference Ontology**
Axioms count	49	119	64	118	108
Logical axioms count	26	74	36	76	56
Class count	14	17	14	21	31
Object property count	8	11	8	14	5
Data property count	0	2	3	2	0
Properties count	8	13	11	16	5
Individual count	1	19	0	11	0
DL expressivity	ALCROI	AL(D)	ALC(D)	ALCI(D)	SRI

Table 2. How Ontometrics base metrics vary between various test cases and the reference ontology. For each case, it is indicated whether the patterns or the MP have been used.

Axiom type	**Case1 No pattern No MP**	**Case2 Pattern No MP**	**Case3 No pattern MP**	**Case4 Pattern MP**
owl:Ontology	Yes	Yes	Yes	Yes
owl:Class	Yes	Yes	Yes	Yes
owl:ObjectProperty	Yes	Yes	Yes	Yes
owl:DatatypeProperty	No	No	Yes	Yes
rdfs:domain	Yes	Yes	Yes	Yes
rdfs:range	Yes	Yes	Yes	Yes
rdfs:subClassOf	Yes	Yes	Yes	Yes
rdfs:subPropertyOf	No	No	No	No
owl:disjointWith	No	No	Yes	Yes
owl:equivalentClass	Yes	No	Yes	No
owl:Restriction	Yes	No	Yes	Yes
owl:imports	No	Yes	No	Yes

Table 3. Use of types of OWL axioms in various test cases. For each case it is indicated whether the patterns or MP were used.

References

1. Blomqvist, E., Presutti, V., Daga, E., Gangemi, A.: Experimenting with extreme design. In: Cimiano, P., Pinto, H.S. (eds.) Knowledge Engineering and Management by the Masses. pp. 120–134. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
2. Lai, E.R.: Metacognition: A literature review. Always Learning: Pearson Research Report **24**, 1–40 (2011)
3. Potoniec, J., Wiśniewski, D., Ławrynowicz, A., Keet, C.M.: Dataset of ontology competency questions to sparql-owl queries translations. Data in Brief **29**, 105098 (2020). https://doi.org/https://doi.org/10.1016/j.dib.2019.105098, `https://www.sciencedirect.com/science/article/pii/S2352340919314544`
4. Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. International Journal on Semantic Web and Information Systems (IJSWIS) **10**(2), 7–34 (2014)
5. Schraw, G., Moshman, D.: Metacognitive theories. Educational Psychology Review **7**, 351–371 (1995)
6. Wang, Y., Zhao, Y.: Metacognitive prompting improves understanding in large language models. arXiv preprint arXiv:2308.05342 (2023)
7. Zhou, Y., Liu, Z., Jin, J., Nie, J.Y., Dou, Z.: Metacognitive retrieval-augmented large language models. arXiv preprint arXiv:2402.11626 (February 2024). https://doi.org/10.48550/arXiv.2402.11626, `https://arxiv.org/abs/2402.11626`, accepted by WWW 2024