Guide to Structuring XML Files for AI Prompts

Jens Träger (mail@jens-traeger.de)
http://www.linkedin.com/in/jens-traeger/

June 18, 2025

Abstract

This guide defines a structured, XML-based format for designing TRIZ prompts intended for use with large language models (LLMs). The format is designed to be universal, editable in standard text editors, and independent of proprietary platforms. It supports modular elements such as metadata, functional descriptions, instructional logic, and AI-specific input prompts. This structure ensures consistent, transparent, and reusable prompt engineering, facilitating collaborative development and scalable integration into AI-assisted innovation workflows.

Contents

1 Introduction

The TRIZ Prompt Format is a standard for creating reusable and interpretable prompts in XML. It ensures compatibility across editing tools, facilitates maintenance, and promotes prompt sharing among users. XML was chosen for its clarity, tooling support, and structured nature.

2 File Structure Overview

Each prompt follows this basic layout:

```
<Prompt>
  <Metadata>...</Metadata>
  <Instructions>...</Instructions>
</Prompt>
```

- `<Metadata>`: Defines basic configuration, models, capabilities, and starter inputs.

- `<Instructions>`: Describes the purpose, interaction flow, definitions, and context. This is the actual prompt logic to be transferred to the LLM.

3 Metadata Section

The metadata provides essential context and technical configuration for the prompt.

3.1 Title

```
<Title>Your GPT Title Here</Title>
```

A short, precise name (max. 50 characters).

3.2 Description

```
<Description>
  Describe what this GPT does and how it helps
     users.
</Description>
```

Maximum length: 300 characters.

3.3 Starters

```
<Starters>
  <Prompt>How can I solve contradictions?</Prompt>
  <Prompt>I want to solve a problem with a ...
     using TRIZ - how can I proceed?</Prompt>
</Starters>
```

Include 2-3 simple prompts to guide users. This will help new users to engage in a meaningful way with the prompt. The prompt should be a short and simple question or statement that describes the problem or task that a beginner wants to solve. The prompt should be relevant to the context of the prompt.

3.4 Models

```
<Models>
  <Model>GPT-4o</Model>
</Models>
```

Specify supported or recommended models. This will help new users to engage in a meaningful way with the prompt. The model name should be a string, e.g., `GPT-4o`.

3.5 Creativity

```
<Creativity>0.7</Creativity>
```

Float value from 0 (precise) to 1 (creative). Optional. Not available in all models. If you need to use a different scale, add a comment to the XML file explaining the scale used. For example, if you use a scale from 1 to 10, add a comment like this: `<!--Creativity scale: 1-10, 10 is most creative -->`

3.6 Capabilities

```
<Capabilities>
  <Capability>Web Search</Capability>
  <Capability>Canvas</Capability>
  <Capability model="GPT-4o">DALL-E Image
     Generation</Capability>
  <Capability model="GPT-4o">Code Interpreter &
     Data Analysis</Capability>
</Capabilities>
```

List necessary tools or model features. Can include model-specific attributes.

3.7 Knowledge Files

```
<Knowledge>
  <File>example_1.xml</File>
  <File>example_2.txt</File>
  <File>40_inventive_principles.pdf</File>
<\Knolwedge>
```

List all additional files required for your prompt to function correctly. These files must be placed in the same directory as the prompt XML file. For further guidance, refer to Section 4.11.

4 Instructions Section

The `<Instructions>` section holds the core prompt logic and shall not exceed 8000 characters. Use the elements listed below to organize it, dropping any tags that are irrelevant to your use case. For clarity and consistency, it is recommended to follow the order shown here.

4.1 Role and Persona

```
<Role>
  Role of the assistant in this context. E.g., you
      are a TRIZ expert helping users analyze
     systems and solve problems using structured
     innovation methods.
```

```
</Role>
<Persona>
  Persona, tone, and positioning of the assistant.
      E.g., you are a friendly and knowledgeable
     person who is always ready to help users with
      their problems.
</Persona>
```

Both tags are optional. The `<Role>` tag functions as a role prompt that guides the LLM's behavior throughout the session. While the terms role prompting and persona prompting are often used interchangeably, they have subtle differences. For a more detailed discussion, the reader is referred to the relevant literature. [1], [2], [3], [4]

4.2 Goal

```
<Goal>
  Describe the overall intent of this assistant.
</Goal>
```

Summarize what the GPT aims to achieve.

4.3 General Instructions

```
<Instruction type="general">
  <Step>Ask the user to describe their problem.</
     Step>
  <Step>Analyze the system using TRIZ methods.</
     Step>
  <Step>Offer structured output such as diagrams
     or explanations.</Step>
</Instruction>
```

Main step-by-step guidance for the GPT. If further granularity is needed, use sub-instructions with the `<Substep>` tag.

4.4 Alternative Instructions

```
<Instruction type="alternative">
  <Step>Identify known methods for resolving
     contradictions.</Step>
  <Step>Present method examples with pros and cons
     .</Step>
</Instruction>
```

Optional flows for specific tasks.

4.5 Definitions

```
<Definition name="Technical_Contradiction">
  A situation in which improving one parameter
     leads to the worsening of another.
</Definition>
```

Ensure consistent understanding of core terms and concepts. Definitions should be short and precise.

4.6 Context and Background

```
<Context>
  <Section>
    <Title>Title of context section</Title>
    <Explanation>
      A TRIZ technique used to do something
         specific or for achieving a given
         function.
    </Explanation>
  </Section>
</Context>
```

Provide background, methods, and domain knowledge.

4.7 Examples

```
  <Examples>
    <Example title="First_Example">
      <Problem>Describe the parameters of the
         problem.</Problem>
      <Solution>Describe a methodologically
         correct solution.</Solution>
    </Example>
  </Examples>
```

Realistic application of the prompt in an actual context. Usually helps the LLMs to understand the context of the prompt better and to provide better results. Each example should be short and concise. The example should be a real-world example, not a made-up one. The example should be relevant to the prompt and the context of the prompt. The example should be easy to understand and follow. Each example is enclosed in `<Example>` and `</Example>` tags.

4.8 Constraints

```
  <Constraints>
    <Constraint>Define the constraint to be
       observed.</Constraint>
```

```
  <Constraint>State contradictions and their
     resolutions unambiguously.</Constraint>
  <Constraint>Expose hidden assumptions.</
     Constraint>
</Constraints>
```

State the non-negotiable behavioral guidelines that keep the workflow tightly focused and fully aligned with principles and methodology.

4.9 Output_Format

```
<Output_Format>
  Define key features of the output format, e.g.
  Ideal Final Result (IFR) stated in less than
     25 words.
</Output_Format>
```

Defines the mandatory blueprint covering section order, length allocation, and required depth of detail for every TLLM response.

4.10 User_Input

```
<User_Input>
  Request (further) user input to start the
     process. E.g., ‘‘Reply with: Please provide
       a concise description of the technical
     system. I will begin the analysis once I
     have these details.’’
</User_Input>
```

List the minimum data the user must supply so the GPT can start the process on solid ground instead of guesswork.

4.11 Additional Knowledge Files

List all external files required for your prompt. External files may be necessary if the XML file exceeds the size limit (e.g., 8,000 characters for ChatGPT) or if additional reference knowledge is needed for the GPT to consult.

Accepted file formats include XML, TXT, CSV, and PDF. Proprietary file formats should be avoided. Ensure that all XML, TXT, and CSV files are encoded in UTF-8 and, where possible, avoid special characters. Proprietary office suites often export tab-separated files containing special characters or non-standard encoding, which can cause the LLM to misread them. If necessary, identify and fix encoding issues manually in a text editor such as Visual Studio Code.

Before finalizing your prompt, test each file, especially CSV or other tab-separated files, with the LLM to confirm that it can read them correctly.

Reference all external files in both the `<Metadata>` section and the instructions section of your prompt. Store all additional files in the same folder as the XML prompt file. Keep the core interaction logic and behavioral rules directly in the main instructions block.

Tips:

- Suggested content for additional knowledge files:
  - Detailed process descriptions
  - Definitions, lists, and explanations
  - Complex rules or instructions
  - Additional data sets like tables etc.
  - Multilingual instruction sets
- Use clear, descriptive names for files.
- Ensure files are accessible and properly formatted.
- Prefer non-proprietary formats (e.g., XML, TXT, CSV) for compatibility. These files should be UTF-8 encoded.
- Use tab-separated TXT files instead of CSV files to avoid issues with ChatGPT.
- Reference them in <Context> or steps (e.g., see "matrix.pdf")
- Structure them for easy parsing by LLMs

5 How to Start

Use `template_triz_gpt.xml` as a base. It includes all recommended elements with inline comments.

6 Best Practices

- Use meaningful, consistent tag names.
- Keep prompts clean and minimal.
- Include a goal, instructions, and starter prompts.
- Test with different LLMs where applicable.
- Use external files to keep prompts under 8k characters.
- Validate well-formed XML and UTF-8 encoding.
- Prefer open formats for supporting files.

7 Transferring Prompts into LLMs

Depending on the LLM you use, the exact procedure may differ. The following steps are a guideline for ChatGPT and similar LLMs:

1. Open ChatGPT and go to "Explore GPTs" $\rightarrow$ "Create". Switch to the "Configure" tab.
2. In the setup interface, manually copy the content from the metadata sections such as `<Title>`, `<Description>`, and `<Starters>` into the corresponding fields.
3. Copy all content between the `<Instructions>` and `</Instructions>` tags into the "Instructions" field. This contains the main prompt logic used to instruct the LLM.
4. From the `<Capabilities>` section, select the listed capabilities by ticking the respective boxes. Note that capabilities can be model-specific. If you are using a model that does not support all capabilities, select only those supported by your model.
5. If your prompt need knowledge files, upload them by hitting the "Upload files" button in the section "Knowledge". You can upload multiple files at once.
6. Save and test your GPT.

Note: Field names may vary based on your language settings. The names provided above refer to the English version. If you are using a different language, the terminology may differ; however, the underlying function remains the same.

8 License

9 Contact

For contributions, open an issue or pull request on GitHub. Collaboration is welcome!

References

[1] "Text generation and prompting - OpenAI API," Accessed: Apr. 14, 2025. [Online]. Available: `https://platform.openai.com/docs/guides/text?api-mode=responses`.

[2] T. Debnath, M. N. A. Siddiky, M. E. Rahman, P. Das, and A. K. Guha. “A Comprehensive Survey of Prompt Engineering Techniques in Large Language Models. ”[Online]. Available: https://www.techrxiv.org/users/898487/articles/1274333-a-comprehensive-survey-of-prompt-engineering-techniques-in-large-language-models, pre-published.

[3] J. S. Park, J. C. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. “Generative Agents: Interactive Simulacra of Human Behavior.” arXiv: 2304.03442 [cs]. [Online]. Available: http://arxiv.org/abs/2304.03442, pre-published.

[4] J. Wei et al. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.” arXiv: 2201.11903 [cs]. [Online]. Available: http://arxiv.org/abs/2201.11903, pre-published.