Guide to Structuring XML Files for AI Prompts

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

April 10, 2025

Introduction

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.

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 a contradiction?</Prompt
     >
  <Prompt>I want to improve a system with TRIZ.</
     Prompt>
</Starters>
```

Include 2-3 simple prompts to guide users.

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.

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.

4 Instructions Section

The `<Instructions>` section contains the main prompt logic. It should not exceed 8000 characters.

4.1 Goal

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

Summarize what the GPT aims to achieve.

4.2 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.

4.3 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.

4.4 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.

4.5 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.

4.6 Examples

```
<Examples>
  <Example title="First_Example">
    <Problem>Describe the parametersroblem.</
       Problem>
    <Solutution>Describe a methodologically
       correct solution.</Solution>
  </Example>
</Examples>
```

Realistic application of the prompt in a 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.

4.7 Additional Knowledge Files

List external files if the content exceeds the XML file limit. Suggested formats include XML, TXT, CSV, and PDF. Reference these in the meta data section and in the instructions part of your prompt. Store all additional files alongside the prompt file in the same folder. Always keep core behavior and interaction rules in the main instructions.

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, CSV, TXT) for compatibility.
- 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 License

8 Contact

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