Systemic Analysis and LLM Integration for SVG Generation in a Kaggle Competition

Nelson D. Posso, Edward J. Garcia, Jaider C. Carvajal

2025

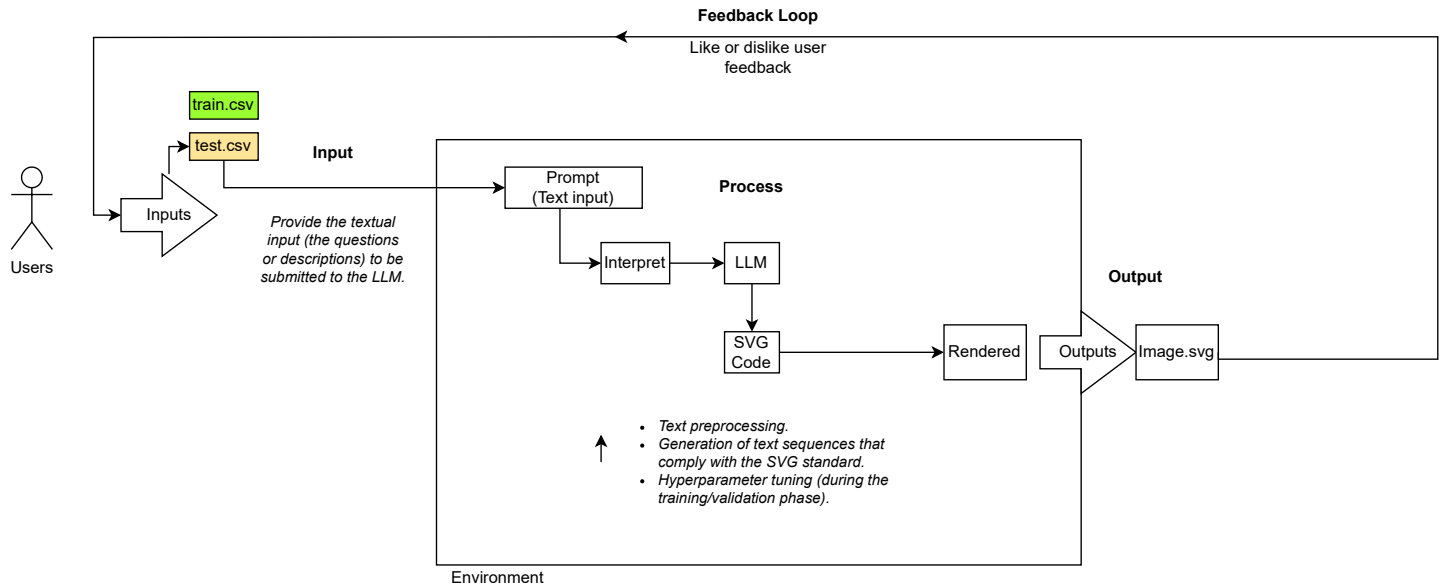
Introduction

- The Kaggle competition "Drawing with LLMs" challenges participants to generate SVG images from textual prompts.
- This requires a system capable of interpreting language, generating graphics, and validating output.
- We approached the problem from a systems engineering perspective.
- Our goal: build a modular, robust, and automated pipeline integrating a local LLM.

Systemic Analysis of the Competition

- Key elements:
  - Inputs: textual prompts, training data.
  - Outputs: valid SVG strings.
  - Actors: users, developers, evaluation gateway.
- Focused on identifying system boundaries, feedback loops, and variability.
- Used systems thinking to structure the problem and analyze behaviors.

Systemic Diagram



Systemic Diagram of information, actors, and feedback

System Architecture - Overview

- Four main components:
  - Description Processor
  - LLM Connector
  - SVG Validator
  - Performance Analyzer
- Designed for modularity and reliability.
- Each component interacts via defined interfaces.

LLM Integration Strategy

- Used Ollama with `llama3:8b` model for local inference.
- Advantages:
  - Offline, fast, and cost-effective.
  - Control over parameters and prompt design.
- Prompt engineering was essential for SVG validity.

Prompt Engineering

- System prompt ensures:
  - Output is valid SVG only.
  - No text explanation.
  - Only allowed tags: `<svg>`, `<rect>`, `<circle>`, etc.
- Reduces risk of hallucination and invalid output.
- Core for successful generation pipeline.

Simulation and Evaluation

- Two phases:
  1. Manual simulation with GitHub Copilot
  2. Automated local LLM with Ollama
- Trade-offs:
  - Copilot: higher quality, manual only
  - Ollama: lower quality, full automation

Evaluation Metrics

- Time per generation: 20s
- Complexity: SVGs with basic shapes and attributes
- Resources: moderate CPU and memory use
- High reliability under automation

Discussion

- System showed:
  - Chaotic behavior from prompt changes
  - Emergent properties from model interactions
  - Variability and noise in output structure
- Simulation tools resemble a digital twin.
- Strong potential for predictive optimization.

Conclusions

- Local LLMs like Ollama provide reliable automation.
- System modularity allows for future scalability.
- Understanding prompt impact is critical.
- Open-source models facilitate education and control.

Future Work

- Distributed deployment and better load balancing
- Integration with visual feedback loops
- More expressive SVGs through fine-tuning
- Expanding prompt complexity with better planning

References

- M. C. Jackson, *Systems Thinking: Creative Holism for Managers*, Wiley, 2003.
- L. von Bertalanffy, *General System Theory*, George Braziller, 1968.
- Meta AI, "Llama 3.1: Open Foundation and Fine-Tuned Chat Models," Meta AI Blog, 2024. `https://ai.meta.com/blog/meta-llama-3-1/`