Structuring Unstructured Information

Demonstrating AI Capabilities in Systems Engineering and Governance

May 2025

By Dwayne Phillips

This essay is from Dr. Dwayne Phillips, PhD. Dr. Phillips is available for hire to research, analyze, and report on topics from AI to writing to budget to management. d.phillips@computer.org. This and other research reports are available at https://dwaynephillips.net/MediumEssays/index.html

Summary

Experiments show that current chatbots can create structure from unstructured information in text documents. Such has always been performed by experienced subject matter experts who applied painstaking and diligent attention to detail. The return on investment of these one-dollar-a-day chatbots is high. This is a call to action for experienced professionals. I need to be the person using AI who replaces me.

Demonstrating AI Capabilities in Systems Engineering and Governance

Several experiments were performed to demonstrate AI capabilities in systems engineering and governance. Those two professions involve the linking of unstructured information in different documents. Doing so by hand, historically the prevalent method, requires diligent attention to detail by subject matter experts. It is quite time consuming and expensive. The experiments, however, show that current chatbots can do these tasks in seconds.

For the experiments, OpenAI's ChatGPT 4o was used. This paper is not an endorsement of that system over the many others that are available. This system was used as it was readily available. Other systems, no doubt, can do just as well for these experiments. For the rest of this paper, OpenAI's ChatGPT 4o will be called "the chatbot."

The experiments described in the following paragraphs move from the simple, more structured to the complex, less structured.

As an aside, while writing this paper, I was interviewed for a job requiring a senior, subject matter expert systems engineer. The job description included the tasks that the chatbot performed in these experiments. I told the interviewer about the experiments and how current chatbot capabilities could perform the great majority of the job they were staffing. I was not hired.

Requirements, Design, and Implementation

There were three short documents used in this experiment. The Requirements document listed three basic requirements for a water dispenser. The Design document listed several design choices for the dispenser, and the Implementation document described a water dispenser. These were Microsoft Word documents that contained numbered text such as "Dispenser R1: The dispenser must fit in an apron pocket." While this is unstructured in that it is not contained in spreadsheet cells or a structured database, a practiced systems engineer or governance professional would quickly recognize the structure of the information.

The three documents were uploaded to the chatbot. Without prompting, the chatbot understood what each document contained and how they were related to one another. The chatbot produced a table showing which requirements led to which design elements and to the implementation. That, again, was without prompting.

To complicate the problem, a requirement was added to that document and a design element meeting the new requirement was not added. The chatbot found the unmet requirement. Further complicating the problem, a design element was added that was not needed, i.e., a "nice to have" but not required. The chatbot found this extra, unrequired design element.

In summary, the chatbot traced forward and backwards through the three documents noting how they satisfied, didn't satisfy, and over built the system.

Analysis of a Proposal Based on a Request for Proposal

Two documents about four pages long were uploaded to the chatbot in this experiment. The first contained several dozen paragraphs with each describing what a customer wanted from a type of person. The second document contained several dozen paragraphs with each describing how a provider would supply a person who would meet the required service.

These two documents were Microsoft Word files that had section numbers and paragraph headings. The information was not cells in a spreadsheet or entries in a structured database. A working professional would have recognized the structure of the information, but these contained unstructured information.

The chatbot was prompted to check if the second document described services that would satisfy the requirements of the first document. The chatbot verified that the second document was compliant with the first and created a table listing the corresponding paragraphs from the two documents side by side.

To complicate the experiment, the contents of the second document were rearranged and paragraph headings were deleted. This removed some of the structure that a person would expect. Given the same prompt, the chatbot responded the same way as before and created a listing of the corresponding paragraphs from the two documents side by side.

To further complicate the experiment, several paragraphs in the second document were deleted so that it did not comply with the requirements of the first document. The chatbot correctly identified the noncompliance and listed the paragraphs of the first document that were not met by the second.

In summary, the chatbot understood the text contents of both documents and linked the associated paragraphs in a table as a human subject matter expert would.

Finding Requirements in a Text Document

A thirty-page document that contained the requirements for a dozen persons on a contract was uploaded to the chatbot. While there was some structure to this document (section headings, etc.), the document was poorly organized and written with tasks associated with a type of person scattered haphazardly throughout.

This document was a Microsoft Word files that had section numbers and paragraph headings. The information was not cells in a spreadsheet or a structured database. A working professional would have recognized the structure of the information, but the file contained unstructured information.

The prompt for the chatbot was:

This file contains contract requirements. Find all requirements that relate to a System Engineer position. This includes those in the System Engineer section as well as other requirements in other sections that typically fall to a system engineer. List the original text.

This is an eight-hour task for an experienced person. The chatbot performed this task in seconds.

Finding Tasks in a Speech

The final experiment involved the transcription of a speech. While the speaker had notes, the speech was an ad-lib commentary around those notes. The transcription had a

paragraph for each phrase from the speaker as that is how the speech-to-text software transcribed it. This was completely unstructured information.

During the speech, the speaker would mention “things we should do” and “encouragements to action” typically found in a speech from an executive to an audience of colleagues.

The prompt given to the chatbot was, “The attached file is the transcript of a speech. What are the action items that are contained in the speech?”

In seconds, the chatbot returned a one-sentence summary of the speech and a list of several dozen tasks given directly or indirectly by the speaker.

Overall Analysis of the Results

The chatbot performed all tasks given in the experiments in seconds. Many of those tasks would take hours of painstaking diligent work by a person. All these tasks were performed by persons several years ago. The chatbot’s output was as good as that of the persons.

Implications

Return on Investment: the chatbot costs about a dollar a day. In half an hour, it produced the work an experienced person would need several days to do. Most of the time consumed by the chatbot was uploading files, writing prompts, and copying and pasting the results of the chatbot.

Model-Based Systems Engineering: the chatbot used unstructured documents to perform all the things that Model-Based Systems Engineering (MBSE) does. MBSE is a methodology that uses models to represent and analyze complex systems, enabling a more structured and efficient approach to development, testing, and operation. It replaces traditional document-centric systems engineering with a model-based approach. This means using digital models as the primary way to exchange information, represent systems, and manage requirements. MBSE requires purchasing expensive tools that need lengthy training and practice to master. Practitioners already have the tools and expertise to create the unstructured documents used in these experiments.

Experts in MBSE can legitimately argue this implication. What today’s chatbots can do, however, is significant and should cause MBSE proponents to take notice.

I just replaced myself: The experiments showed that today’s chatbots can do what I have practiced for decades. Jensen Huang recently said, “You will not lose your job to AI, but will

lose it to someone who uses it. I recommend 100% take advantage of AI, don't be that person." (Sozzi)

Mr. Huang is right. Someone using AI will replace me.

I better be that person using AI who replaces my former self.

References

Sozzi, B. (2025, May 6), Nvidia CEO Jensen Huang on AI: Every job will be affected, some will be lost. Yahoo. https://finance.yahoo.com/news/nvidia-ceo-jensen-huang-on-ai-every-job-will-be-affected-some-will-be-lost-221359044.html