www.pipelinepub.com
Volume 21, Issue 5

GenAI Agentic User Assurance

By: Mark Cummings, Ph.D.

2025 is being declared as the year of Agentic GenAI. The challenge in ensuring resulting good user experiences lies in avoiding the “Sorcerer’s Apprentice" problem. The old “be careful what you ask for” problem. In this case it is: how can we make sure that the agents do what we really want them to do? Existing GenAI guard rails and previous work on policy interfaces can help. But they are not sufficient. We need a formal, rigorous way to: tell the agents what we want them to do and not to do; make sure that when first deployed they only do what is expected; then follow their ongoing operation to make sure they continue to do what is expected. This can best be done with explicit descriptions of objectives, algorithms, and constraints, plus an oversight mechanism.

Agentic AI

In the search for how to best take advantage of the power of GenAI, attention has turned to using it to create agents. In the beginning of 2025, the Associated Press released an article titled: “This Year AI Was All About Putting its Tools To Work.” It contained the following passage: “… An Agentic Future … eventually AI agents will come together and perform a job the way multiple people come together as a team …” This is a good way of describing the vision behind the “deep tech” being developed by some in the GenAI community. Shortly after, Forbes published an article titled “Understanding and Preparing for the 7 Levels of AI Agents.” Although the article has some technical shortcomings, it underlines the focus on agents.

Nvidia anticipated this trend. Late in 2024, Nvidia announced their board for Edge Computing devices called Orion. At CES in early 2025, Nvidia announced that in May 2025 they will start offering their Digits mini desktop AI computer. It is about the size of a Mac Mini but it is basically a slimmed-down Grace+Blackwell for $3,000. It offers very high performance and enough memory to run a 200 billion parameter model. Two Digits can be connected to enable running a 402 billion parameter model on your desktop for $6,000.

Against this background, tools are appearing to simplify the process of creating agents. And companies are gearing up to create targeted solutions. Some are in-house efforts. Others are targeted at selling to industry segments.

Thus, there is money, energy and technology focused on making GenAI agents. Good user experiences will lead to success in these efforts. Bad user experiences will be hard to overcome. In this respect, the history of autonomous driving can be instructive. Good user experiences will depend on things not going wrong with GenAI agents.

Sorcerer's Apprentice Problem

There are many things that can go wrong with GenAI-based agents, including problems with hallucinations, data normalization, dynamic systems, cybersecurity, and general technical breakdowns. These kinds of problems can be addressed by general improvements in LLM technology, careful architectural structuring, and careful choice of application areas. What is harder to overcome is the "Sorcerer's Apprentice" problem.

The story of the Sorcerer's Apprentice goes all the way back to ancient Greece and forward to recent movies. All the stories share a single element: a young person using magic to create an agent to do their work for them, then running into trouble when the agent does exactly what the young person asked it to do. For example, in one telling of the sorcerer's apprentice, a young apprentice to a wizard uses magic to create agents out of brooms. He tells the agents to carry buckets of water and wash the floor. The brooms do exactly what is asked, over and over, eventually creating a flood. The apprentice tries to create another agent to control the first ones and things get further out of control. The fundamental problem is that the apprentice doesn't consider all the consequences of what he is asking: what will happen when the agents do exactly what he has tasked them with. The result is that the agents run amok. Finally, the Master Wizard comes in and fixes everything.

For simple agents the problem may not be so great. For example, an agent that takes data from two different data sources, combines it, and puts the result into a report. For more complex systems the situation can be more troublesome. This has serious implications for systems that involve interactions between multiple agents. Particularly for agents that overlay critical infrastructure, work in medical environments, and other essential functions.

Specifying Behavior

To avoid a problem at the outset, existing GenAI guard rails and previous work on policy interfaces can be helpful. But we need a more rigorous way to tell the agents what we want them to do and not do. This can best be done with explicit descriptions of objectives, algorithms, and constraints. Using the management by objective process in modern organizations can be helpful in thinking about specifying objectives for each of a group of AI agents that will be working together. Each agent should have a clearly defined set of objectives in clear priority order. Algorithm specification is important both for managing behavior of individual agents and for cooperation between groups of agents.

There is also a relationship between the type of algorithms used and the objectives and constraints. For example, mathematical or statistical processes may be an algorithm. How these algorithms express their results may determine how objectives and constraints are stated.

Of course, an LLM (Large Language Model that underlies a specific GenAI system) is itself an algorithm. The selection of which LLM to use can be a critical choice. Which one in an agent structured such that it has multiple to choose from. Which one is in use by another agent that a particular agent is cooperating with can also be important.

Finally, constraints are statements about actions, behaviors or conditions to never allow. An example might be a group of three agents. Each monitoring the operation of a machine driven by electricity. Each agent has an objective to run their machine at the highest possible efficiency. In such a situation, a constraint might be to never allow the total power consumption of the three machines to exceed X Watts.

The Role of the Wizard

In the Sorcerer's Apprentice, the Wizard comes in to save the day. Of course, if there was a wizard present at the beginning, there wouldn't have been a problem to start with. Following on the analogy with GenAI, there has to be a "Wizard" present at the beginning, and at the end.

There has to be a process for testing a GenAI agent before it is deployed in production. Most software is tested before being introduced into production. For GenAI agents, that kind of testing should be done. If the normal software testing a vendor and an organization use before deployment doesn't include running the agent in a simulated environment; that should be added. These simulated environments are sometimes called "sandboxes." This testing must not only cover expected behavior. It must also test for unexpected behavior.

This kind of normal testing is good, but not enough. Additional precautions are prudent. When the agent is first introduced into production environments, it needs to be monitored — something like being watched by a wizard. The "wizard" can be a manual process, an automated process, or a combination of the two. It looks to make sure that the agent is doing what it is supposed to be doing and meets expected quality metrics. In LLM terminology, "quality" has a specific meaning — a percentage measure of how often the LLM produces the correct response.

Ongoing observation is also prudent. Recent reports indicate that agents can suffer degradation of quality over time. The observation wizard may be configured to have a lower duty cycle (time between observations, for example) later than that at initial deployment. Also, budgets and project schedules should take into account the contingency of refreshing the agent periodically. Measurements should also take into account the time required for refreshing the agent. If that is not done, the danger is that quality falls below acceptable levels before a refreshed agent can be deployed.

Conclusions

We are still in the process of determining what and how GenAI can produce the best benefits. GenAI agents are the current focus of that effort. Turning the push to deploy GenAI agents into good user experience may turn out to be the greatest 2025 challenge. The Soccer's Apprentice problem is real. We need a formal, rigorous way to tell the agents what we want them to do and not do, make sure that when first deployed they only do what is expected, then follow their ongoing operation to make sure they continue to do what is expected. This can best be done with explicit descriptions of objectives, algorithms, and constraints, plus an oversight mechanism.

In closing, it is always a good idea to remember that old saying: "Be careful what you ask for."