July 2025

Enough AI copilots! We need AI HUDs

In my opinion, one of the best critiques of modern AI design comes from a 1992 talk by the researcher Mark Weiser where he ranted against "copilot" as a metaphor for AI.

This was 33 years ago, but it's still incredibly relevant for anyone designing with AI.

Weiser's rant

Weiser was speaking at an MIT Media Lab event on "interface agents." They were grappling with many of the same issues we're discussing in 2025: how to make a personal assistant that automates tasks for you and knows your full context. They even had a human "butler" on stage representing an AI agent.

Everyone was super excited about this… except Weiser. He was opposed to the whole idea of agents! He gave this example: how should a computer help you fly a plane and avoid collisions?

The agentic option is a "copilot" — a virtual human who you talk with to get help flying the plane. If you're about to run into another plane it might yell at you "collision, go right and down!"

Weiser offered a different option: **design the cockpit so that the human pilot is naturally aware of their surroundings.** In his words: "You'll no more run into another airplane than you would try to walk through a wall."

Weiser's goal was an "invisible computer"—not an assistant that grabs your attention, but a computer that fades into the background and becomes "an extension of [your] body."

Example of the different points of view: Pilot's Assistant

The goal of the pilot's assistant is to enhance the ability of the pilot to fly the airplane

- **Interface agent: like a co-pilot**
  - watching the planes systems and the situation
  - offering advice, answering questions
- **Ubicomp: like a simpler, better airplane**
  - the arrangement of controls, displays, windows, seats, etc.
  - the ability to act is enhanced by the total system
  - no locus of expertise
- **For example: being alerted of a potential collision**
  - *agent*: "collision, collision, go right and down"
  - *ubicomp*: background presentation of airspace information for continuous spatial awareness, as in everyday life. You'll no more run into another airplane than you would try to walk through a wall.

Weiser's 1992 slide on airplane interfaces

HUDs

There's a tool in modern planes that I think nicely illustrates Weiser's philosophy: **the Head-Up Display (HUD), which overlays flight info like the horizon and altitude on a transparent display directly in the pilot's field of view.**

A HUD feels completely different from a copilot! You don't talk to it. It's literally part invisible—you just become naturally aware of more things, as if you had magic eyes.

Copilot

"Person"
Natural language interface
Responds in seconds

Heads-up display (HUD)

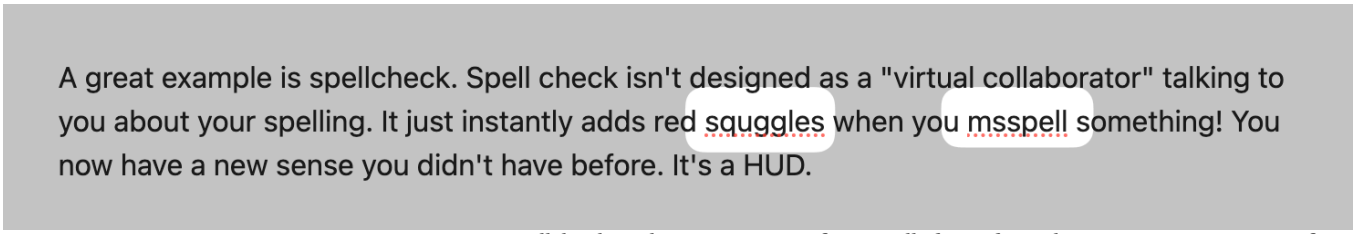
"Tool"
Live data visualization
Responds in milliseconds

Designing HUDs

OK enough analogies. What might a HUD feel like in modern software design?

One familiar example is spellcheck. Think about it: **spellcheck isn't designed as a "virtual collaborator" talking to you about your spelling.** It just instantly adds red squigglies when you misspell something! You now have a new sense you didn't have before. It's a HUD.

(This example comes from Jeffrey Heer's excellent Agency plus Automation paper. We may not consider spellcheck an AI feature today, but it's still a fuzzy algorithm under the hood.)



Spellcheck makes you aware of misspelled words without an "assistant" interface.

Here's another personal example from AI coding. Let's say you want to fix a bug. The obvious "copilot" way is to open an agent chat and ask it to do the fix.

But there's another approach I've found more powerful at times: **use AI to build a custom debugger UI which visualizes the behavior of my program!** In one example, I built a hacker-themed debug view of a Prolog interpreter.

With the debugger, I have a HUD! I have new senses, I can see how my program runs. The HUD extends beyond the narrow task of fixing the bug. I can ambiently build up my own understanding, spotting new problems and opportunities.

Both the spellchecker and custom debuggers show that automation / "virtual assistant" isn't the only possible UI. We can instead use tech to build better HUDs that enhance our human senses.

Tradeoffs

I don't believe HUDs are universally better than copilots! But I do believe **anyone serious about designing for AI should consider non-copilot form factors that more directly extend the human mind.**

So when should we use one or the other? I think it's quite tricky to answer that, but we can try to use the airplane analogy for some intuition:

When pilots just want the plane to fly straight and level, they fully delegate that task to an autopilot, which is close to a "virtual copilot." But if the plane just hit a flock of birds and

needs to land in the Hudson, the pilot is going to take manual control, and we better hope they have great instruments that help them understand the situation.

In other words: routine predictable work might make sense to delegate to a virtual copilot / assistant. But when you're shooting for extraordinary outcomes, perhaps the best bet is to equip human experts with new superpowers.

Further reading

- A nice discussion of one approach to this idea can be found in Using Artificial Intelligence to Augment Human Intelligence by Michael Nielsen and Shan Carter.
- A more cryptic take on the same topic: Is chat a good UI for AI? A Socratic dialogue
- A discussion of how the the HUD philosophy intersects with on-demand software creation: Malleable software in the age of LLMs

---