The Generic User Approach: Reimagining AI-Browser Interaction

Introduction

As artificial intelligence advances, one persistent challenge remains: enabling AI to interact with web applications as naturally as humans do. Current approaches to browser automation often involve complex DOM parsing, brittle selectors, and application-specific coding that breaks whenever websites change. But what if we approached this problem differently? What if, instead of forcing AIs to understand the complex internal structures of websites, we enabled them to interact with browsers just as humans do—by seeing what's rendered on screen and interacting with those visual elements directly?

This article proposes a fundamental shift in how we think about AI-browser interaction: the Generic User approach.

The Current Landscape: DOM-Dependent and Fragile

Today's browser automation approaches typically rely on one of two strategies:

1. **DOM-Based Interaction**: Tools like Selenium and Puppeteer access the Document Object Model (DOM) directly, using selectors to find and manipulate elements. This approach requires intimate knowledge of a website's structure and breaks easily when that structure changes.
2. **Pure Visual Approaches**: Some newer tools use computer vision to identify UI elements from screenshots. While more flexible, these approaches lack the semantic understanding that comes from accessing the browser's own interpretation of the page.

Both approaches suffer from fundamental limitations when applied to complex applications like healthcare systems (e.g., Epic), banking portals, or enterprise software.

The Generic User Insight

The key insight of the Generic User approach is simple yet profound: **We don't need to rebuild the browser's understanding of the page—we just need to access it.**

A browser already solves the incredibly complex problem of converting HTML, CSS, and JavaScript into a visual representation with interactive elements. It already knows:

- What elements are visible on screen
- Which elements are interactive
- The hierarchical relationship between elements
- Which text belongs to which elements

- The semantic meaning of many elements (buttons, form fields, etc.)

Rather than forcing an AI to rediscover all this information from either raw DOM or screenshots, we can provide a hybrid interface that gives the AI access to the browser's own understanding of the page.

The Implementation: A Browser-Integrated Hybrid Approach with Specialized AI Architecture

The Generic User approach would involve a dual-AI architecture that mirrors human cognitive specialization:

1. The Generic User AI

A specialized AI system trained specifically for browser interaction:

- Understands general UI patterns and interaction models
- Translates high-level intents ("find the patient record") into specific browser interactions
- Develops expertise in navigation patterns across different interface types
- Acts as an intermediary between domain-specific AIs and the browser interface

2. The Domain-Specific AI

A separate AI focused on the task domain (healthcare, finance, etc.):

- Understands domain-specific knowledge and goals
- Makes decisions about what needs to be accomplished
- Communicates with the Generic User AI through high-level intents
- Remains free from needing to understand browser interaction details

3. Technical Implementation Components

This dual-AI system would be supported by:

1. **Browser Integration Layer**: A browser extension or modified browser that exposes the rendered page structure to the Generic User AI
2. **Semantic Mapping Interface**: A standardized API that translates between the browser's internal representation and concepts the Generic User AI can understand
3. **Action Interface**: A mechanism for the Generic User AI to issue interaction commands (clicks, typing, scrolling) that are executed through the browser's own event system
4. **Visual Context**: The rendered visual state of the page, giving the Generic User AI the same information a human would have

This approach provides a separation of concerns that mirrors human cognition: just as we have specialized brain regions for visual processing and motor control separate from our higher-level reasoning, this architecture separates browser interaction expertise from domain expertise.

Advantages Over Current Approaches

The Generic User approach with specialized AI architecture offers several critical advantages:

1. **Universality**: Once implemented, it works across any web application without application-specific programming
2. **Robustness to Change**: Since it operates at the level of rendered elements rather than DOM structure, it's far less sensitive to website updates
3. **Natural Interaction**: The AI interacts with elements as they appear to users, making behavior more predictable and understandable
4. **Simplified Training**: The specialized Generic User AI can be trained intensively on browser interaction patterns, while domain-specific AIs can focus on their areas of expertise
5. **Future-Proofing**: As long as applications maintain human interfaces, the Generic User AI will be able to use them
6. **Cognitive Division of Labor**: By separating browser expertise from domain expertise, both systems can evolve independently and specialize in their respective domains
7. **Scalable Learning**: The Generic User AI becomes more capable with each new interface it encounters, building a generalized understanding of UI patterns that transfers across applications

Real-World Applications

This approach would be particularly valuable for:

Healthcare Systems

Medical applications like Epic present enormous complexity and change frequently. The Generic User approach would allow AIs to navigate these systems just as medical professionals do, without requiring extensive reprogramming with each update.

Financial Services

Banking and financial applications could be automated securely, with the AI interacting through the same interfaces humans use, subject to the same security controls.

Enterprise Software

Complex business applications with frequent updates could be used by AI assistants without breaking existing automation workflows.

Technical Implementation Considerations

Implementing the Generic User approach would likely involve:

1. **Browser Extension API**: Extending existing browser extension APIs to expose rendered element information
2. **Accessibility API Integration**: Leveraging and extending existing accessibility interfaces that already provide semantic information
3. **Event Simulation**: Creating mechanisms to generate native browser events that accurately simulate human interaction
4. **AI Training Framework**: Developing training methodologies that teach AIs to understand general web interfaces rather than specific applications

The Path Forward

While current automation tools like Skyvern and Browser Use are moving in similar directions, the full realization of the Generic User approach still represents a meaningful advance in the field.

By focusing on the rendered interface rather than implementation details, we can create truly generic AI systems capable of using any web application—just as humans learn general principles of web interaction rather than memorizing the structure of each website they visit.

This approach doesn't just solve a technical problem; it represents a philosophical shift in how we think about human-computer interaction and the role of AI in that interaction.

Conclusion

The Generic User approach to AI-browser interaction offers a more elegant, robust, and universal solution than current DOM-based or pure vision-based approaches. By accessing the browser's own understanding of the rendered page and employing a specialized AI architecture, we eliminate an entire class of brittleness in AI automation while simultaneously making the AI's interaction more human-like.

The separation of browser interaction expertise from domain expertise mirrors human cognitive specialization, allowing each AI system to excel in its respective domain. The Generic User AI becomes a universal interface layer that can translate domain-specific intents into effective browser interactions across any web application.

As we continue to develop AI systems that assist humans in complex tasks, this dual-AI architecture offers a critical pathway to making those systems more capable, more reliable, and more adaptable to the everchanging digital landscape.

About the Author: This concept was developed by a retired technologist with decades of experience observing how humans interact with computers and how systems evolve over time. The insight came from recognizing that we often force AI systems to solve already-solved problems rather than giving them access to existing solutions.