

Will 2004 Be the Year We Abandon HTML?

By James Hobart, President, Classic System Solutions

January 2004



Probably not. However, I have a strong sense that this will be the year for a major shift in technology deployment for enterprise Internet applications away from HTML/JavaScript and toward the adoption of rich client solutions. Actually, the term "rich client" has been somewhat abused, so let's start off with a common sense clarification. We define a "rich client" as allowing true business processing, asynchronous data transfer between client and server, and pixel-level graphics manipulation, all without requiring trips to the server.

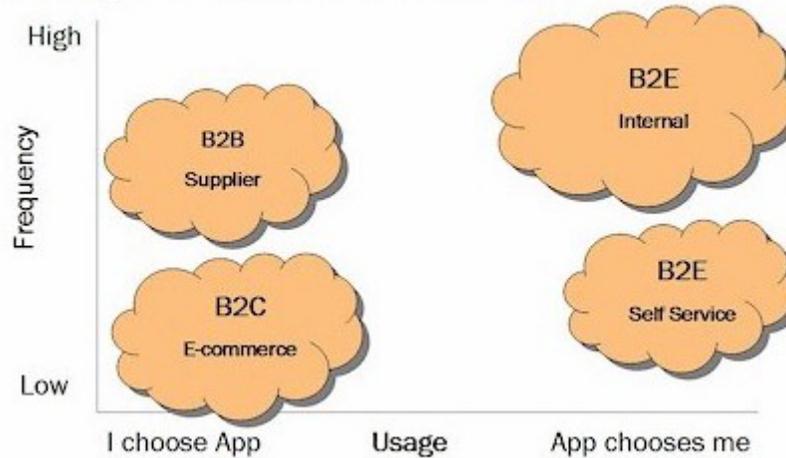
What's Wrong With HTML?

Actually, for 95% of the web, HTML works great for delivering content to users. However, when large-scale enterprise transactional applications are being deployed via the Web, HTML often simply does not deliver what users want. We've been called in on numerous application efforts where the "old" client/server application was replaced with the new web application only to find frustrated users telling us stories about how a typical transaction takes far too many mouse clicks, requires multiple pages to navigate, and produces mayhem if the Back button is clicked. It appears the "click/refresh" issue of HTML/browsers is not going away anytime soon; yet, the main usability issue associated with transactional enterprise Internet applications is slow response time. Noble attempts at solving these usability and performance issues are often handled by hard-working development teams; but, in the end, what we often see is a huge set of unwieldy JavaScript code, late projects, and frustrated end-users.

Which UI Design Patterns Need More Than HTML?

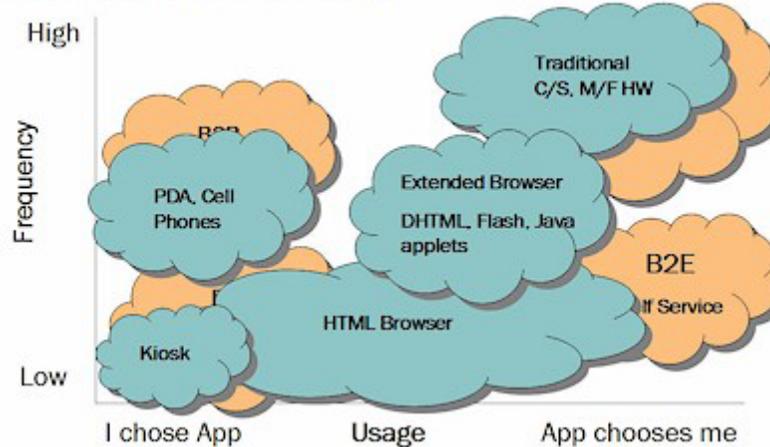
A UI design pattern is a technique for describing in detail a proven solution to a known design problem. We classify UI design patterns into high level "domains" as a way of grouping similar operation models, task patterns, and deployment constraints. While the HTML/browser combination has been an excellent fit for the B2C (business to consumer) domain, it quickly breaks down in more complex domains like B2E (business to employee) and B2B (business to business) where usage frequency, complexity, and scale increase significantly. Often these domains contain applications provided by the company to its employees where there is no choice but to use it (CRM, ERP, etc.). As companies look to leverage what they've learned on their external facing systems, they will often find that the user types are very different and the deployment constraints, like browser support, are not as broad. Extending browser functionality to meet the needs of these UI pattern domains often requires a new set of technologies, standards for deployment, and usability testing to ensure the new patterns are leveraged for an optimal user experience.

Design Pattern Domains



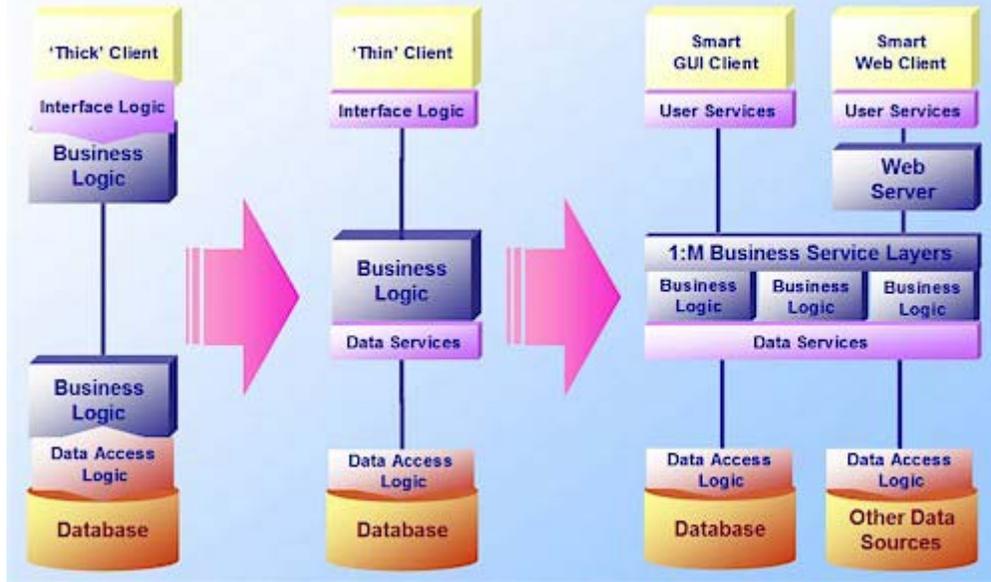
UI Constraints on These Patterns

UI Client Domains



How Enterprise Internet Applications Work

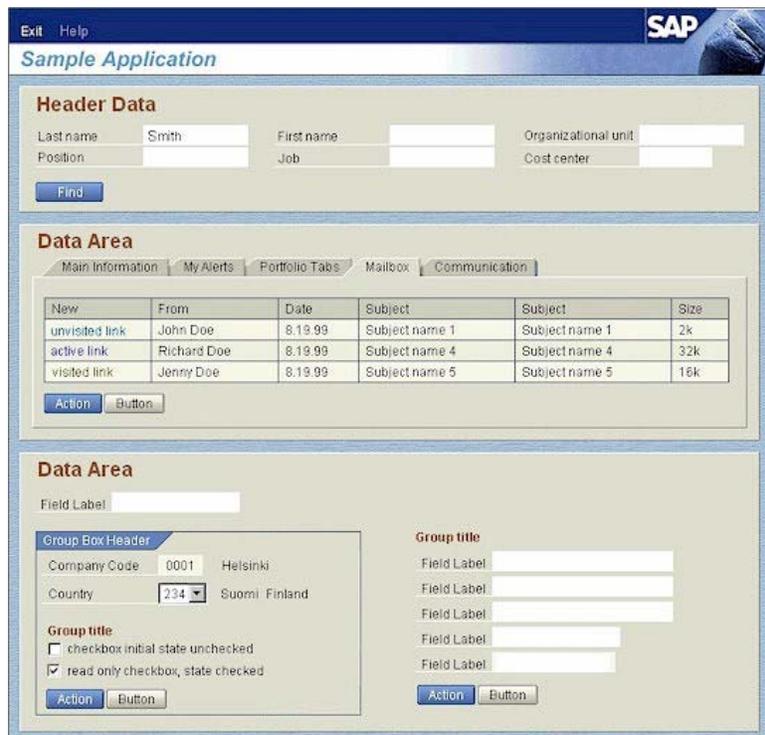
The high cost of supporting "thick client" solutions followed by many attempts at "zero footprint" thin client solutions has finally led us to the next generation of rich UI clients, shown as "smart" clients in the diagram below. These smart clients can use a web server and server extensions to load as much or as little client code as is necessary to optimize the user experience, while meeting the existing network bandwidth and architecture constraints. Often, there is a small Java applet or proprietary software that manages communication with the server-side extensions to facilitate the "smart client" behavior via XML or Java. Flexibility in the way the client and server communicate is critical as you deploy complex enterprise applications. As we learned during the client/server era, it is critical to separate the presentation, business, and data layers. Rich client solutions that force you to place business logic in the UI layer will be less stable and more difficult to maintain, as we already have seen with the HTML/JavaScript solutions deployed over the last few years. The deployment of smart clients can vary from mobile phones and PDA's to browser implementations and stand-alone applications. Because of the wide range of users and tasks in today's large companies, we feel a solid enterprise Internet application platform should fully support the occasionally connected user model, as this UI pattern domain is quickly becoming commonplace.



Examples of Rich Client Applications in Action

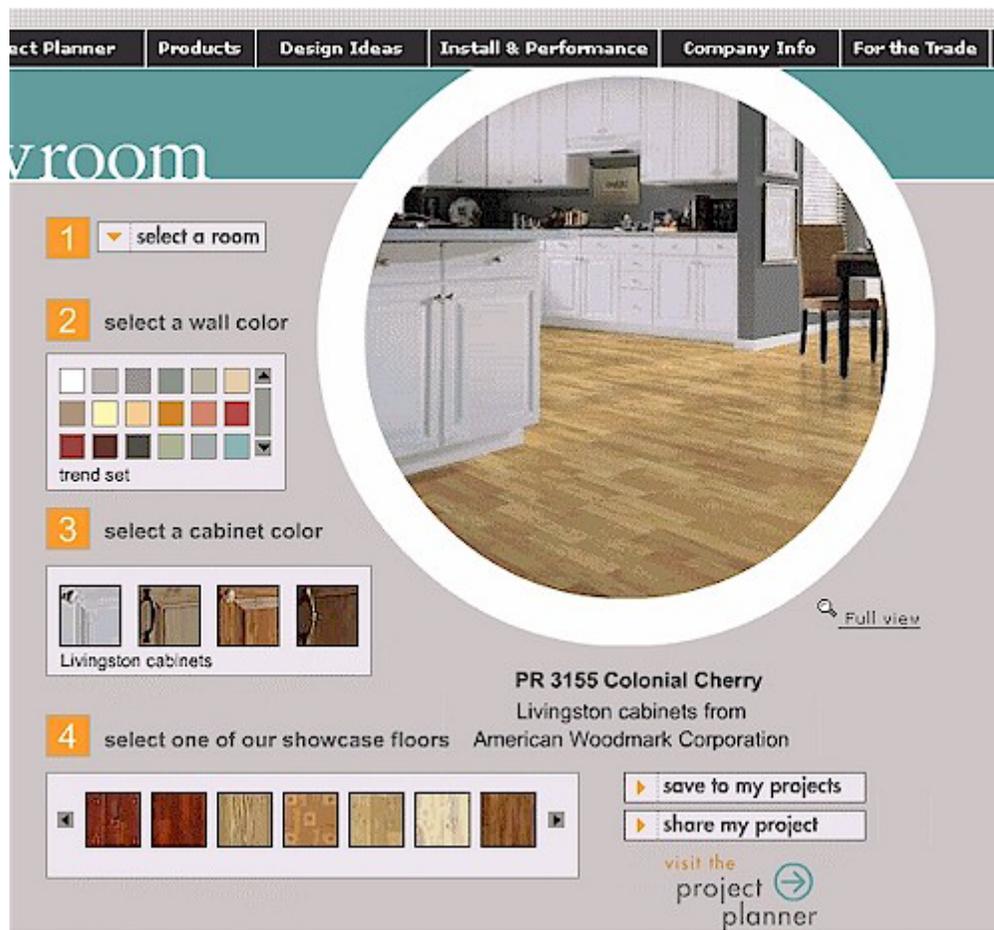
The next generation of Internet applications will be capable of much more than simply rendering pages. They will be able to perform complex calculations, handle data manipulation, send and receive data asynchronously, and allow occasionally connected users to perform tasks in a mobile environment. Presentation issues like redrawing sections of a screen or displaying multiple views simultaneously will be commonplace, and all of this will be available independent of the backend architecture it is connected to.

Good Example (SAP) - Tabs and Data Manipulation



This example illustrates the use of a basic summary/detail form. Yes, this can be done with HTML; however, the page refresh delays associated with each click of a tab in high-volume transactional systems will create excessive network traffic and slow the response time of the system. Users accustomed to sub-second response times of traditional client/server and mainframe applications will often forget data as they move tab to tab or simply avoid navigating to other tabs to avoid the frustrating delays. When implemented as an enterprise Internet application, this type of visual design pattern should enable the development team to tune the application to quickly load the first tab being displayed and asynchronously load secondary tabs as a background task if necessary. In essence, a robust enterprise Internet application should allow you to tune the client deployment load based on the various constraints of the platform, usage models, and network bandwidth.

Good Example (Flash) Visual Floor Selector



This example illustrates a good use of Flash to provide a rich, interactive method to quickly select and display the visual rendering of the selection without making round trips to the server. We feel visual configuration scenarios are one of the best uses of Flash-rich client interfaces.

Choosing an Enterprise Client Platform

We believe choosing a rich client platform will be a strategic decision similar to J2EE vs. .Net on the server. At first glance, some teams may feel they are simply purchasing widget libraries; however, for enterprise applications, a solid infrastructure and framework that easily integrates with your middleware platforms will be critical for long term success. Also, if you are deploying high-volume transactional systems, you will need to fully understand how your rich client solution can be tuned to allow either more or less code to run on the client or server.

Right now, the playing field for enterprise Internet applications seems to be falling into a few distinct camps. The following table is a summary of the key things to consider for high-volume transactional Internet applications.

Key Features to Consider for Enterprise Internet Platforms

- Standards-based (XML, J2EE)
- Server-side database integration
- Extend server objects to the GUI
- Support off-line operations on the client
- Support stateful persistent client sessions
- Support for automatic server push to the client
- Support standard security approaches (data encryption, digital certificates, client sandbox)
- Allows a less complex development model

Platform Flavors

A certain amount of "market validation" is occurring as vendors in this area all strive to prove their architectural approach is best for the long term (3 - 5 years). We're currently performing a detailed analysis of key players in each area and will report on our findings in a later article. For now, here's a sampling of the flavors and players.

Presentation Server	Nexaweb, Canoo
DHTML/JavaScript	Isomorphic, CanyonBridge
Flash	Macromedia, Lazlo
Pure Native Client	Curl, Dreamfactory
Windows (.Net Winforms)	Microsoft

Want More Information on Enterprise Internet Applications?

Classic System Solutions has been designing enterprise Internet applications for over a decade, and we are excited to include these solutions in our [consulting practice](#), [training](#), and [design guidelines](#). Give us a call or e-mail jimh@classicsys.com, and we'll provide you with a detailed set of critical success factors for building enterprise internet applications along with a [free usability kit](#).

About the author:

James Hobart is an internationally recognized user interface design consultant based in California, USA. He specializes in the design and development of large-scale, high-volume client/server and web applications. He is an expert in GUI design for transaction processing systems and strategies for migration to thin-client graphical user interfaces. He can be reached at jimh@classicsys.com

Find more articles on Usability at www.classicsys.com