Tomorrow's Web, and Why

I used to wonder why the digital revolution seemed so stupid until I realized it isn't a revolution, it's a social theme. Computers allow us – as a culture – to be more of what we are, and faster. Technology is driven by demand, not supply, and reeks of consumption, profit, and entertainment.

This is why Microsoft dominates the world with a third-rate operating system, Apple's success is based on telephones, Google's vision is limited to commerce, and web programming is done with awful HTML and Javascript. I will tell you about the next step and what's being done to implement it.

Communications

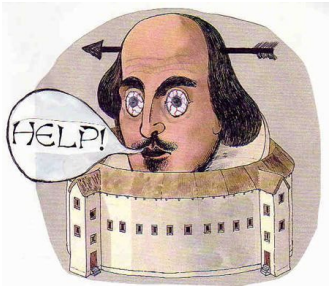
Since communication was written in urine, it has always existed in two forms: vertical and horizontal. Vertical means linear, which means a narrative or story. Horizontal means organized by kind, classified, or hierarchical. Hierarchical is actually just one form of organization, but it's the predominant form on the Web.

Writing in P2P.

When you learn, you employ both methods. School shoehorns teaching into a linear form which explains why school sucks and why manuals never sell products. Horizontal learning introduces choice, possibilities, context, and error. This brings us to the crux of the matter: Why are all web sites fractured into topics rather than integrated into a story that explains their reason for existence? Why is the territory between explanation and taxonomy so barren? I admit I'm painting in black and white, so consider the unpopulated grey areas in between.

The typical website has some main areas with radiating branches. If the site is a store or service, then the areas are departments and the branches are products. This is fine if you understand your relationship to the site and you're simply trying to find an item. At the opposite extreme is an ebook that has one starting point, one story, and one conclusion. You are processed like a cow in an abattoir with little more say in how you'll come out at the other end.

Imagine that you didn't understand the site and you want to. In this case, the department store model is confusing: You won't know what you're looking at, how one thing connects to the next, or what things are used for.



Imagine a Shakespearean play or a thriller recast as a hierarchy of themes and relationships. Not only would the result be unreadable, it would be unintelligible. This is why a real department store always provides a rudimentary map of the whole, and why a successful narrative allows for some simplification into categories. While one or the other form of exposition predominates, few presentations succeed in one form only.

The future lies in the integration of linearity and modularity, and if you're reading this you already agree because blogs bridge the vertical and the horizontal with a variety of breadth and depth. Yet technology follows demand, it does not lead it, and most tools available for Web programming do not facilitate our progress, they impede it. Future tools will support the construction of narratives that usher communication into a world in-between.

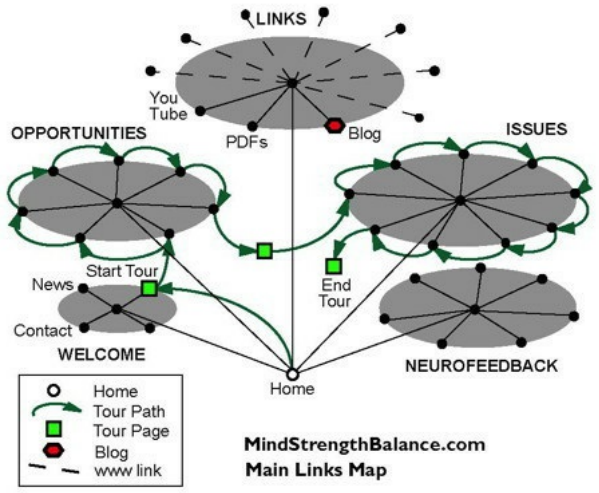
Adding Narrative

I wrote www.mindstrengthbalance.com to sell neurofeedback training. Neurofeedback is poorly understood. Sold as psychotherapy, most neurofeedback providers build their sites on the department store model. These are vendors selling a product, and their sites assume you know what you want. They aim to direct you to it. It's therapy as a commodity, and the commodity is illness.

My interest is in enhancement. In this context neurofeedback is under-appreciated. People would be interested if they knew what I was offering, but simply putting what I offer on a shelf is as uncompelling as bottling insight. There is a story to be told, and hierarchies do not tell stories. This is why, incidentally, most people do not understand society or their role in it: Society is a hierarchy, and hierarchies don't tell stories.

The Mindstrengthbalance.com site is both a resource and a narrative because its pages are arranged as either a hierarchy or a sequence. The hierarchy follows the usual menu and item structure, the narrative is a sequential tour. If the user opts to take the tour, then they'll be led through a connected narrative. At any point the user can stop the tour and the narrative disappears.

This diagram maps the web-site. The open circle is the home page, black dots are pages, and lines are links. The black lines show the hierarchical structure, while the green lines show the linear structure.



www.mindstrengthbalance.com/welcome/about.html

When first visiting Mindstrengthbalance.com, you see menu themes and topics. Pressing the **Take the Tour** button atop the home page opens a navigation panel with instructions. The Tour starts here. Navigation buttons, shown below, persist throughout the tour. Each page is a step in the exposition. Press the **Stop** button and the tour's navigation buttons disappear, leaving you back in a department store of themes and topics.

Outside the Toolbox

Standard HTML and Javascript easily support either a hierarchical or a narrative site structure, but not both alternatives in the same site. To build a site that supported the user's choice required the additional third-party Javascript library called Persist.js. You can download it from the Pablotron web site. This library is free, unsupported, and minimally documented. Persist.js does not do anything fancy; it simply provides that element required for any sort of intelligence: persistent memory.

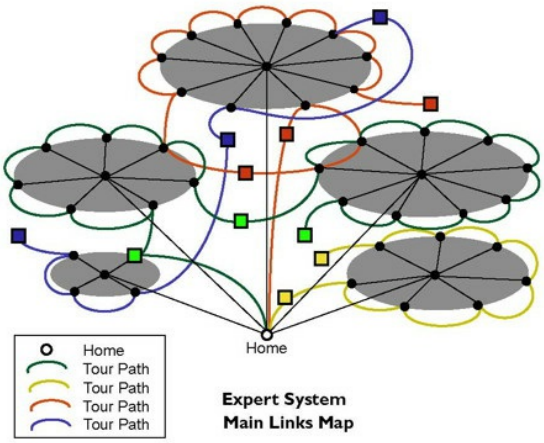
It is true that cookies are a standard memory tool, that server-side programming supports global variables, and that HTML5 – an HTML standard proposed for sometime in the future – moves closer to providing memory access, but cookies are limited and difficult, server-side programming complex and expensive, and HTML5 a mishmash of yesterday's ideas that's already obsolete. The fact that easy access to memory is not supported by HTML or Javascript marks them as defective and, by implication, damns the whole effort to give the Internet intelligence greater than zero.

Being able to store and recall information throughout a user's visit to your website takes you outside the standard Web toolbox. What has this great new tool given the Mindstrengthbalance.com website? Simply the ability to remember how the user wants to see the site. You could hardly ask for less, yet it's not part of the standard tool set. With the simple ability to remember **something,** you can start to build a website that is more than a catalog with an order form.

Fetish,
www.andrewrreed.com

The failure of the internet to live up to its potential does not sit on the doorstep of sites that were difficult to create, but on the graves of ideas that could not be realized.

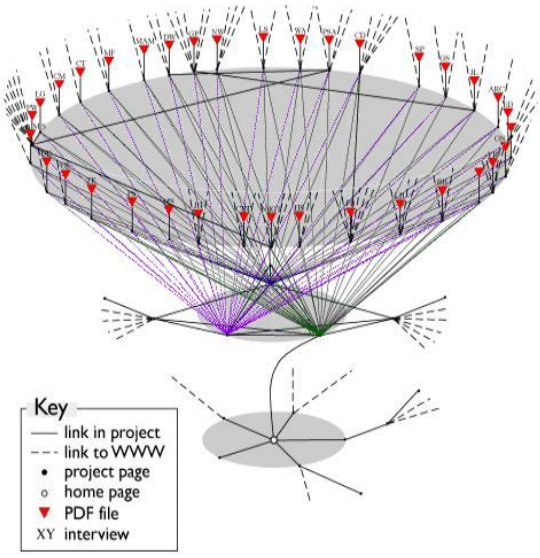
The adjoining map, an invention derived from the MindStrengthBalance site structure, shows various paths through a site. The user selects the path, thereby making this an "expert system." But what is a conversation but a path through ideas set by the speakers' interests? An expert system is just a dialog, but poor tools are keeping us tongue-tied.



Logo-tecture

My hypertext book, *The Learning Project*, expands traditional narrative by adding reader-participation. Primarily a series of interviews, the book's chapters are organized in either of three ways: at random, by topic of study, or by the speaker's age. You initially see three equal paths to the text's completion, but as you move through the text you encounter questions answered differently by each of the structures. Giving you the power to rearrange the content gives you an active role in experiencing the issue.

The Learning Project is a story about itself. The project's central question – what is learning – can be answered in three different ways. The implication of the random structure is that each speaker has a different answer. According to the structure organized by topic, the definition of learning differs according to the speaker's interest. According to the third structure, the answer differs according to his or her stage in life. It's the last answer, that learning differs according to one's age, that shows the question to be one of personal growth. Learning is personal evolution.



www.tengerresearch.com/learn/whatitcontains.htm

This diagram maps *The Learning Project* website. The open circle at the bottom is the home page, black dots are pages, and the lines are links.

The dark blue lines at the back show pages linked according to each speaker's area of interest, work, or study. The violet lines at the left show pages arranged according to each speaker's age group. The green lines arrange the speakers alphabetically, which is essentially at random.

The red triangles at the top of the map represent interviews. The lines crossing between interviews are relationships between those interviewed. The dashed lines represent links to web pages outside the site.

The Learning Project presents the radical idea that learning is not schooling, and schooling is not important. The book is in hypertext form and is freely available at tengerresearch.com/learn.

The Ghost of an Internet Future

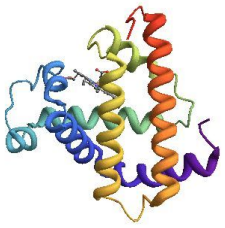
There are many interesting and astonishing sites employing Web technology, from the simple (Craig's List) to the complex

artists	musicians
childcare	local news
general	politics
groups	rideshare
pets	volunteers
events	classes

Simple navigation on Craig's List

(www.wolframalpha.com). The tools needed to build these sites involve a combination of difficulty, drudgery, and expense requiring hundreds of thousands of dollars or more.

The reason for the huge chasm between basic and complex sites is the dichotomy between front-end *interface* tools and back-end database *programming* tools. HTML and Javascript are front-end tools, and the front-end essentially means the presentation layer. The presentation layer is not supposed to do any heavy lifting; it's designed simply to format and present information generated somewhere else.



Myoglobin molecule from WolframAlpha.com

The back-end refers to the generation and management of content, and this is done by database programs like MySQL, host server software like Apache, and programming languages like PHP. All of these sit on top of, and are hopefully isolated from – but in reality not entirely – the operating system.

To make anything for the Web you must have at least a presentation layer. To add sophistication you need expertise in the back-end tools because, as I mentioned, the front-end tools have no brains. This is a huge leap. Each of these tools has its own language; and for every interdependent tool you add you double the number of quirks, incompatibilities, patches, and administration tasks. Either the Internet will devolve into a tower of one thousand languages, or some consolidation will occur. This is exactly the problem that the industry has been struggling with for decades.

There are two main paths to integration: application frameworks and content management systems. On top of every programming language is some framework or system. *Wikipedia* lists over 100 frameworks built on 20 programming languages (such as Django, Pajamas, and CherryPy), and over 150 management systems (such as Joomla!, Drupal, and Wordpress) – some built on top of languages, and others on top of frameworks built on top of languages.

This is reminiscent of the Cambrian Explosion when a plethora of bizarre invertebrates appeared to populate the oceans, all but a few of which soon vanished. The same danger holds for the Web designer: Any of these frameworks might become extinct next year. On one hand these products integrate an intractable diversity of systems and protocols; on the other hand they place all of one's eggs into one basket.

Dinosaur Simple, Dinosaur Complex

Today's complexity is due to a multiplicity of tools, all doing some essential part of the whole. There is no general-purpose tool built to provide all services. There is no tool "species" that has both fingers and thumb attached to the same hand. This is what these frameworks are trying to provide, and this is clearly what lies in the future: an all-purpose tool that is practical and affordable.

The obstacles to this tool are the committees formed to engineer future standards. These committees are made of corporations with similar target markets, different products, and competitive allegiances. Their goal is not to advance the Web but to preserve their markets.

The XML Committee existed for decades to establish standard data formats but still hasn't done so. The HTML Working Group of the World Wide Web Consortium (W3C) is developing the standards for the next version of the presentation layer, called HTML5. There is hardly any attempt at collaboration beyond that. Alexi Boronine comments on his blog that:

... the HTML platform is being designed by people who think they know what's best for you. Which means:

The standards are always playing catch-up with reality.

Browsers are always playing catch-up with the standards.

The standards are made to appease everyone, including people you don't care about.

The CoBi-7 Conference Bike, www.conferencebike.com

– to which one commentator adds:

If we are to have a fresh look at HTML, CSS, (and) javascript technologies, we have to admit that it is just shit ... if it was easier a lot more normal people would write web pages

We are in a "Monsanto moment" where the crop is being destroyed by those who control it. Real evolution is not happening at the committee level; it's happening at the consumer level where frameworks are creating usable environments. Each of these environments is complete in itself, and provides increasing levels of web access and control. What you do not want is what currently exists: a legacy selection of tools, each special purpose, complex, and difficult to combine.

There are clearly two kinds of consumers who will fuel the development of two kinds of frameworks: the unsophisticated Web developer (i.e. "most people") and professionals (i.e. businesses). The unsophisticated Web development tools offer "click and drag" frameworks that don't require programming skills.

Such easy-to-use frameworks exist for different target markets. There are frameworks specifically for blogging, music, photography, and retail. There is a secondary market to extend these simple frameworks supporting vendors like Pagelines and StudioPress who offer better "themes" for the Wordpress platform. Here are some of the basic platforms:

Blogging and Content Management: **Wordpress**, Wordpress has little competition, see interesting articles on that topic here and here.
Presentation and Conferencing: **Prezi**
Data Management: **Knack**
Form Management: **WooFoo**
Photo Management: **Stock.xchng** and **Flikr**
Online Print Presentation: **Behance**
Collections and Exhibit Publication: **Omeka**
Web "pages, galleries and blogs": **Squarespace**
Music Delivery: **Bandcamp**, **SoundCloud**, and **TuneCore**.
Newsletters: **MailChimp** and **Madmimi**

If you're interested in building a site using one of these tools, then consider these two issues:
1) Which tool is the best for my task?
2) Which tool is the most reliable in terms of performance and longevity?

I cannot answer the first question but with regard to the second question I note this interesting comment, taken from *Forbes* magazine, that appeared in September 2012 at *Digitopoly*:

Today WordPress powers one of every 6 websites on the Internet, nearly 60 million in all, with 100,000 more popping up each day. Those run through its cloud-hosted service, which lets anybody create a free website online, attract 330 million visitors who view 3.4 billion pages every month.

All Sorts of Dinosaurs Eating Their Lunch

The professional frameworks expose the underlying programming layers: the presentation layer, the database layer, the various protocols like FTP, HTTP, database, Secure Socket layer, and so forth. Forget these professional frameworks unless you're ready to devote your life to Web development.

A review of these tools appears on Ricardo Zuasti's blog in a 2012 series entitled "Web Development Frameworks". His top three choices are Play Framework, Ruby on Rails, and Django. Sebastian Hennebrueder has a free e-book titled "Choosing Web-Frameworks".

For a more prejudiced view – meaning vendors can pay for higher ratings – a google of "web framework reviews" will generate magazine articles. A review of 32 frameworks on the Memeburn site is titled "32 Web Frameworks to Choose From for Your Next Project." Their top two choices are Ruby On Rails and Django.

Here is an interesting observation. Whenever there is a movement with commercial potential, you always see big corporations trying to be there first. Here is a list of 10 Promising Web Platforms from 2008 offered by leading Web companies; all these platforms are now defunct or incidental. Never trust a big corporation to help you move through a rapidly changing world. One company's platform will

never willingly support a competing company's products.

Gobble, Gobble, Nibble, Nibble, Munch, Munch, Scrunch

Robot dreaming

Asituation similar to Web development existed in the Database development world in the 1980's. There were simple database tools, complex tools, and little in between. Task-specific database development platforms were being created to simplify creating specific applications. I was interested in developing such a platform for business automation.

Because every business needs accounting, accounting is a need common to all business software. I designed and marketed my own accounting toolkit for the creation of middle-sized business management applications. The program is called 4th Quarter Accounting and is still servicing this niche.

The chasm between simple and complex business software tools still exists today in spite of my product and many others. It is instructive to consider why the chasm persists and what this means for Web development.

In my field the problem was not so much a lack of tools as a lack of skill. I was able to create a serviceable bridge over the complexity chasm, but my clients had not evolved the smarts to cross it. After all, automation is not just a piece of software – it's a paradigm that requires a different management structure, work force, working environment, and investment of resources. These are still lacking to this day. My 4th Quarter Accounting was a bridge few small to mid-sized companies could cross.

When I wanted to implement the MindStrengthBalance concept I used only rudimentary Web frameworks. I wanted something that I understood so I used Adobe's Dreamweaver to manage my pages and Javascript for functionality. Perhaps a simple framework provides the tools I needed – certainly the complex platforms do – but I doubt it. The ideas I'm exploring are not being offered as Wordpress "themes" or anywhere else. Web tools are still creating sites that are brochures with order pages, and content is still a matter of record keeping, much as it has always been.

There are tons of Web services, lots of buzz, and plenty of networking but I am not impressed. Tools realize ideas they don't create them. If you want to see farther than others then you must get off the bandwagon for "the next big thing" and start thinking. To get on to the next wave, first get off this one.

2001: A Space Odyssey

To receive future issues, subscribe to the *Culture and Neurofeedback Newsletter*.