

# **Analysis 2009**

***Through the Glass Darkly***

*Kurt Cagle*  
*Editor*

*O'Reilly Media*

## Table of Contents

|   |    |
|---|----|
| Introduction .....  | 3  |
| The Financial Crunch hits IT hard through 2009, eases thereafter.....               | 3  |
| Government Gets Into the Software Biz.....  | 6  |
| The End of Traditional Publishing.....  | 9  |
| Energy Sector Faces Volatile Year.....  | 11 |
| Carbon markets heat up, but so does the weather.....                                | 13 |
| IT departments disappear into the Clouds.....                                       | 14 |
| Application Services come into their own.....                                       | 15 |
| The web services era begins in earnest .....  | 17 |
| Syndication forms the backbone of the Writable Web.....                             | 19 |
| XForms and XML-enabled clients gain traction with XQuery databases.....             | 21 |
| Semantics continues to not be RDF, but enrichment, classification and taxonomy..... | 22 |
| Internet Explorer Fades, Firefox Stays the Course, Google Chrome Surges.....        | 24 |
| Some thoughts for a New Era.....  | 26 |

## Introduction

The art of prognostication is not that dissimilar to understanding weaving. Few things ever occur out of the blue - they just hadn't emerged out of the background noise just yet, and as such when they do appear, you can usually see how the threads extend backwards, and add these things into your model.

I've been thinking about models a lot lately. The human brain is a remarkably powerful modelling tool, probably the best such tool on the planet. It keeps track of those things that it can perceive, adds in probabilities for those things it can't see, and uses this to insure that you can get through the day without walking into the side of a building or stepping in front of a car.

Thus, what most people perceive as "reality" is a model that they keep inside their heads, that's constantly changing, and that is for the most part heuristic - it doesn't try to ascribe reasons to why things happen (reasoning is a considerably higher function that came along fairly late in the game) but works by keeping this floating stream of metaphors going - this is like that, so this will behave like that ... until it doesn't. This fact then gets added into the metaphor, that doesn't-ness is a very distinct possibility for "this".

A set of prognostications, then, is a model, one that tries to incorporate past known history with the interaction with other threads. These interactions are where things get exciting. It's relatively easy to follow a thread into the future as long as it doesn't interact with other threads, but the moment that it does, you have to make a guess, based upon what is known, about something that is currently unknown.

This is where prognostication becomes analysis - trying to think through how the interaction of two or more threads will shape each (or whether in fact they will merge and become a broader thread). It's doable ... in most cases if you mix two known things for which analogs exist that are also known then you can have a pretty good idea about what comes next. But not always. This is why prognostication is still more art than science.

One final aspect about prognostication, before launching into my predictions for 2009. There is sometimes a delicate line that you have to walk between being an observer and being a participant. The observer, having no stake in the final outcome, is usually able to interpret those threads and interactions with comparatively little bias, while the participant has a definite bias that colors his or her perceptions, and that can consequently amplify trends in a given model that aren't that important or mask trends that are.

Unfortunately, in order to understand a system, you also have to participate in it. Walking this line between observer and participant is challenging, and the consequences of it should be obvious if you look at what happened on Wall Street, where analysts were paid significant sums of money in order to not only observe the system but also to game it. They grew to see what they wanted to see, not what they needed to see, and in the end this lack of vision brought The Street down.

I can't claim to be completely unbiased (I actually have some strong ones), so as per usual, take my predictions with at least a few teaspoons of salt.

## **The Financial Crunch hits IT hard through 2009, eases thereafter.**

The recession that started in January 2008 looks to be four phased. The first phase, The housing collapse, actually started in August 2007. The financial meltdown hit in September 2008, and likely will continue through to March 2009 or so. The business firestorm is really just getting underway now, and will be the dominant theme for the next 8-12 months while the final phase, currency collapse, will (if it occurs at all) likely not happen for another 12-18 months.

The business collapse itself is taking place in a number of different verticals, which differentiates it from a traditional oversupply recession. In an oversupply recession, the market produces too much of a good or service for the available demand, which usually means that either companies have to cut back on producing goods (and consequently reduce their own profitability) or they fail and fall out of the market. In time, demand rises to meet supply, and the industry in question recovers.

The housing collapse was a classic oversupply recession - too many houses on the market at too high a price, and eventually demand couldn't meet supply. Had the housing market not been fueled by low interest rates when they weren't needed, had the financial industry not done a dice-o-matic on the resulting mortgages, and so on, chances are pretty good that we would be about 2/3 of the way through this by now, IT, except for the specific housing IT vertical, would be relatively unscathed.

The problem now is that risk became baked into the very core of the global financial system like a series of fault lines, and the collapse of the mortgage business was like a deeply buried mortar going off in that mess. This in turn exposed the very ugly truth about finance - that prices are psychological, and when no one knows the value of things, the ability to plan for the future ends.

This fear has manifested in the credit crunch, when banks are terrified of lending money out because they know that the assets that they carry are far below what they should carry in order to stay solvent - and that if they loan out money, they won't have it when the next wave of credit defaults occur (either credit cards or commercial real estate, take your pick - they'll hit about the same time). There's currently an effort to reliquidate the banks by most of the world's governments, though with at best limited success (more on that in future articles).

The business collapse is occurring because of two factors. First companies that had depended upon having readily available lines of credit are finding these lines being cut or dramatically reduced, which makes them much more vulnerable to the variability of incoming contracts ... at a time when everyone else is facing the same problem.

The second is that this has put significant downward pressure on household incomes, as these same lines of credit (in the form of second mortgage refinancing, credit cards and so forth) are now becoming scarce at the consumer level (along with financial investments having plummeted in the last few months). This has resulted in a consumer strike, as people save rather than spend.

For those businesses with a direct consumer face (or those that IT companies who supply services to these businesses) this translates into reduced revenues and shrinking demand, which in turn has a direct impact upon both those people that produce retail hardware and has an indirect effect upon IT companies that produce software to support these retailers. It also means that companies that had projects in the work for FY 2009 are scaling these back or putting them on indefinite hold until they get a clearer read of the economic situation.

One of the problems with recessions is that while there is an underlying economic aspect to most of them (many people just don't have the money in the first place), there is also a psychological aspect. People stop spending (and start saving), in anticipation of two things - first, that when they need the

money, they may not have it, and second, that when the economy is receding, the overall price of both goods and services drop. It makes little sense to take on new purchases (whether new projects or new goods) when demand is dropping and the possibility is fairly strong that they can get those things for cheaper in a year or two.

At an individual level, this is a rational response. The problem comes when everyone does it. At that point, demand dries up, companies go out of business, reducing the overall stock of those same goods and services. This happens with all goods. The problem that we face right now is that the goods that are at the root of the problem - houses - tend to have a comparatively long shelf life compared to Tickle-me Elmo dolls or iPods. They can't be inventoried or written off, which means that it will take considerably longer for demand to meet supply, and as capital investments destroying houses and restoring the property to a usable state can be painful at best.

Unfortunately, this means that the psychological aspects of businesses far removed from housing will be very much held hostage to the housing cycle. Eventually (for a number of reasons) housing prices will stabilize at a new level of equilibrium (which, if reversion to the mean is any indication, should be about 15% below where most prices are now), though it is also likely that any markets will overshoot this level to about 25% or so below current levels before eventually returning to this mean. While estimates vary as to how long it will take, most economists feel that it will be at least another nine to eighteen months, putting the "bottom" of the recession at or around late 2009 or early 2010.

Yet even that won't necessarily be the end of the troubles. A deep financial depression is a lot like a deep cyclonic depression (a.k.a, a hurricane). In a hurricane, a great deal of damage is done by the winds, as windows break, cars go flying and in some cases houses go sliding into the depths or get turned into kindling. Yet the real damage comes from factors such as storm surges, where large amounts of water start moving quickly in areas not designed for water. Hurricanes knock out power, making recovery efforts difficult, and paradoxically, raises the possibility that fires will start that can't be reached or put out, causing even more damage.

The same thing is happening now - we are within the eye of the financial hurricane, a kind of false calm where the winds and pressures are at an unstable equilibrium, but this only has the effect of relaxing things that had bent in response to the hurricane force winds. Unfortunately, once we enter the eyewall on the other side the forces are just as strong, but going in the other direction (one of the reason hurricanes are so destructive).

When this plays out, many retail companies (and not a few medium to large sized malls) will be out of business, their buildings sitting vacant and often poorly maintained. There will be a significant exurban flight as people are forced to sell their properties (typically at a loss) because of unmaintainable mortgages and either rent closer in to population cores or move to higher density housing (this trend will be exacerbated by the number of baby boomers who are reaching retirement age and are staring at fixed incomes with oversized mortgages and reduced bank accounts), a trend which will result in many "bedroom communities" becoming squalid slums or just abandoned altogether.

Suppliers to these retailers are already starting to disappear as retailers cut back on their inventories, and in areas such as automotive manufacturing the tremors as these highly centralized, monolithic conglomerates continue to shutter plants are causing the extensive supply chains to fragment as companies that weren't sufficiently diversified lose their primary customers and go out of business.

As this happens, it also makes it harder for these same companies to continue to pay their leases, which are typically financed at a much higher rate than consumer mortgages are as most are fifteen year rather than thirty year paper. This will be the next major pressure that a lot of companies, even those outside

of the retail sector, will be facing, and software companies in particular are vulnerable to this particular threat, especially as VC financing continues to decline. Expect by the end of 2009 that office buildings will be even more vacant than they were in 2003 at the bottom of the tech recession.

Venture capital, by the way, is also drying up for much the same reason as credit in general - VCs are investors, and they do not in general want to take a loss on a company, even one with a brilliant idea, if they are concerned that economic pressures will never let it get off the ground. Moreover, those same investors already have extant commitments that they are in many cases having to serve as the primary bank for, and this in turn is reducing their willingness to take on new businesses (and will for some time).

A further process that will accelerate is the disaggregation of conglomerates, as companies shed or spin-off divisions that are not profit centers. It's likely that this will hit the big services companies such as IBM, Fujitsu, Siemens and so forth disproportionately, though it might also affect companies such as Microsoft or Oracle. Many of the newly created spin-offs may not make it once cut off from the sheltering effect of the mothership, but others (most notably those that were already successful companies in their own rights) may very well come out stronger in the process.

The next year will be a poor one for mergers, unless they happen to be pure stock exchange mergers (where the two companies agree to on a common stock conversion rate with little actual money changing hands). Acquisition mergers typically require not only cash on hand, but also require access to bonds or other equities that can be used to leverage this cash on hand. With so much uncertainty right now in terms of equity prices, however, raising such funds becomes difficult, even as it seems like we are awash in a sea of junk bonds.

Toward the end of 2009, expect this trend to reverse. Companies with strong cash positions going into this storm will have a much clearer understanding of the shape of the market by the end of the year, and will be able to buy technically sound but financially distressed software companies at bargain rates. If you're vested in a startup, now is probably a good time to discuss acquisition strategies, though probably not at the premium values that many startups tend to value themselves, with an eye towards that sweet spot in the Q4 2009.

There's currently a running debate among economists about what happens when the recession ends. Some, primarily Keynesians, feel that economic stimulus is a necessity to get us out of the current liquidity trap, and that for the most part the debt that most countries are taking on now will restore us back to a period of economic stability with perhaps at worst only modest inflation (in the 3-4% rate) for a few years thereafter.

Others, especially those of the Austrian school of economics, expect that all of the money being created in order to finance the stimulus will, once the credit crunch eases, result in significant inflation, possibly above 10% per annum, leading to a similar situation (stagflation) that caused the 70s to be so hard (well, that and disco).

My money is on the Austrians, as they were surprisingly accurate in their predictions of the present economic crisis, and as a central part of the problem facing the economy right now is that real interest rates (those charged by banks to their primary customers) continues to remain far higher than the nominal 0.25% percent rate that the current Fed has set the prime lending rate.

In short, even after the storm abates, money may continue to be in short supply for many months (or even years) to come. Expect that government stimulus packages and government works programs may in fact have to substitute for the private economy for some time (more on that in the next section).

## Government Gets Into the Software Biz

The incoming Obama administration has, even before taking office formally, pledged between \$650 and \$800 billion dollars worth of public works initiatives, a massive shift away from the laissez faire approach of the outgoing Bush administration.

Of that, it is likely that a significant portion of those funds will be going into automating health care systems, establishing carbon markets, improving business accounting and accountability processes, upgrading educational IT infrastructure and improving power generating infrastructures. One way of thinking about this is that over the course of the next two to three years, *government is about to move into the Cloud, representing the largest growth of government IT investment in the industry's history.*

What this will mean in practice is that, even as software jobs in particular disappear from the private economy, software engineers will be in high demand in the public economy. This process will likely start at the top and work its way down - with CTOs and system architects being brought into senior administrative positions in order to lay out the necessary personnel and infrastructure investment as well as to lay out the best potential architectures for those investments within the first three to six months of the Obama administration proper, followed both by the selections of companies to bid on software projects as well as significant hiring of software engineers internally to support a long term software development culture.

Obama has shown a surprisingly nuanced understanding of open source philosophy (as well as agile methodologies and a deep understanding of network effects), so it is likely that open source software and languages will play a fairly significant role in the IT infrastructure that is going to be used to power this move to the Cloud. A good indicator for this will be Obama's pick as the head of government IT, as it will likely be this person who will be establishing the broad patterns of software acquisition and usage, but I do not expect this to be an old school, proprietary software magnate. My guess, Bill Joy, co-founder of Sun.

By all indications, however, this does not necessarily mean that the government will directly hire software programmers for all of this (though it is likely that there will be a significant amount of government hiring of programmers - possibly as much as 10-15% of the total will be direct government hires). More likely what you will see is the government opening up bids for government projects to the private software sector, in many cases becoming the primary or even only client for a number of them.

Specific projects will almost certainly be along the lines laid out by Obama recently, amounting in essence to the introduction of intelligent networks that would be run in conjunction with physical or social infrastructure improvements. Physical infrastructure improvements will likely look first at transportation infrastructure, though moving away from the highway-centric approaches of years past and concentrating more on rebuilding the badly eroded train and light rail systems, aligning the systems more along the lines of recent efforts by China and Japan than towards pouring more money into highways that are already straining under their existing load (a significant upshot of this will be the potential of moving far more freight traffic by rail and reducing the number of two and three unit long trucks that do such a disproportionate amount of damage to highways).

Another area I suspect considerably more investment will be made is in hardening areas against weather-related disasters, as well as providing mechanisms to far better coordinate recovery efforts after a natural disaster such as a hurricane, tornado or earthquake. A significant portion of the GIS community had begun focusing efforts on disaster warning and recovery after Katrina and related storms - expect this particular sector to get additional funding as well in order to turn what are for the

most part pilot projects that have been very poorly funded into fully functioning emergency defense systems (expect that FEMA will almost certainly get rerouted out of Homeland Security as well).

Healthcare will likely be a second major area of investment, again with three goals - increasing the number of doctors in a system that is suffering from chronic shortages and automating, standardizing the health care system, an area that likely will be a major source of jobs for IT workers, and providing a mechanism for reducing the complex and labyrinthine managed-care system. I expect that by the beginning of 2010, significant work will have begun on establishing standard electronic medical records, likely built on the HL7 model.

The educational system in the US (and to a certain extent in Canada) is nearly in as much trouble as the financial and physical infrastructure systems. The No Child Left Behind Act pushed extensive testing mandates (and fairly punitive actions for schools that failed to meet what was often a crazy quilt of different educational expectations), while at the same time significantly underfunding the educational system at all levels. This has been exacerbated by the shift in recent years from college grants and scholarships to student loans, loans which are now becoming increasingly difficult to come by and consequently are making it increasingly difficult for even relatively affluent students to go to college.

Again this is a problem that will likely take a multipronged approach to solve - refunding education, making teaching more attractive as a profession, will help, but as with many of the situations that exist now, this is perhaps a good opportunity to rethink the educational structure overall. Negroponte's one laptop per child program turned out to be more daunting than originally hoped, especially given that too much of the focus ended up on the price points rather than the ultimate goal of providing access to appropriate online educational resources.

My personal feeling is that education at the primary and secondary levels need to recognize the shifting relationship that kids have with knowledge and information, and begin a fairly significant set of reforms at the teacher education level to rethink the role that teachers have in that process. Standardization needs to happen, certainly, but that standardization needs to occur (and I suspect likely will occur) not at standardized tests but a more consistent standardization of records and communication systems. Unfortunately I think that at least for the next couple of years, educational reform will be largely inchoate, because what's needed here is an evaluation of education itself.

One idea that has been floated (and I think offers some serious potential) is the development of an equivalent of the GI Bill that can be tied into other reforms, making it possible for people to earn college educations in exchange for government service outside of the military. Another reform may be the formal severing of research universities from teaching universities and a re-evaluation of the role that colleges play in the educational process. Educational costs have risen faster than every other sector but health care, yet the quality of that education has been declining steadily for years.

This is an economic reaction - pure and simple; the number of "seats" in most universities has remained more or less constant for the last two decades, even though the number of people seeking a college education have risen significantly. Building new colleges may help that situation somewhat, but in reality that option is going to be limited. What this means in practice is that distance-learning colleges, online certification programs, and related "electronic" schools will need to take up much of the slack. I expect to see more well-crystallized standards for such programs over the next couple of years that provides means for people to study without being on campus, to study in concentrated "marathon bursts" of two to three week 40+ hour courses and the rise of technical publishing companies as being the next generation of university (see my next point for more on this).

The financial crisis that is currently causing much of the pain in the economy has many roots, and as

such no one solution will magically solve the problems that are now endemic to the financial system. However, improving the transparency of banks and other corporations would have made it much harder for much of the fraud and poor judgement to have taken place. It would also have made oversight of the financial regulatory mechanism much easier (something that's becoming increasingly obvious in the wake of Bernard Madoff's recently "discovered" massive Ponzi scheme (along with those of many of his peers).

I'm not necessarily a fan for increased regulation, but I do think that better regulation is definitely in order here. Again this points to a technological solution to what is ultimately a technological problem - monitoring trillions of dollars worth of financial transactions in real time. Business reporting is still mired at the level of the quarter, yet increasingly the tools are available to provide business reports at a daily level of granularity, especially for Fortune 1000 Companies. Many companies have resisted this, not so much because of the cost of implementing such a system but because such close monitoring makes it considerably more difficult to handle financial irregularities ... which is of course the whole point of the regulatory system in the first place.

My gut feeling is that the first year (indeed the first two to three) will likely not see immediate payoffs. The global economy is dealing with nearly forty years of economic excess - the idea that it will be solved in a year is laughable. By this time next year its likely that the initial planning will have been completed, the structures for building the structures put in place and the first real implementation phase just getting underway, which is why I see this as being very much a work in progress for 2009.

## **The End of Traditional Publishing**

For publishing, 2009 is shaping up to be truly ugly. The publishing industry has faced a number of factors that, individually, provided quite a challenge, but collectively they may end up likely significantly altering the industry profoundly over the course of the next year.

If you take a look at the magazine rack of any grocery or drugstore, not to mention book supermarkets like Barnes and Nobles or Borders, what you will see is a nation awash in newspapers and magazine publications. What you may not see is the fact that each of those magazines typically exists on fairly thin margins achieved primarily through the sales of targeted advertisements. A significant proportion of such magazines (perhaps as much as 80% in some places) may actually sell only one copy for every four copies that are displayed on the newsstand, and subscription rates continue to decline for most magazines.

Indeed, one of the most disturbing phenomenon in that sector has been the rise of a whole class of magazines that are essentially distributed for free and are paid for only by specialty advertising. Such magazines have been around for some time, but until comparatively recently these magazines were usually focused in specific trade sectors, such as magazines specifically devoted to electrical line workers. The last decade has seen the rise of a large number of very narrowcast magazines that nonetheless appear on newsstands, at the ragged edge between trade magazines and general interest magazines.

One of the most badly hit areas in the recent financial crisis has been advertising - companies pulling back on their marketing, reducing ad buys, especially for products that have most recently come to market. This will have a domino effect, one that is already discernable if you're watching the racks - magazines being folded, fewer issues being produced, magazines shifting away from monthly to bimonthly publication. Retail shelf space for magazines in outlets such as Barnes and Noble are already noticeably barer than they were a year ago at this time, and it is likely that this trend will accelerate as

advertising revenues continue to drop.

This same phenomenon is hitting newspapers even more dramatically. The last year has seen continued consolidation in the newspaper sector as organizations such as McClatchy buy up regional dailies and weeklies, and the merger of McClatchy and Gannett late in 2007 has left the newspaper industry dangerously entrenched. Now McClatchy is laying off reporters, editors and production staff and is reportedly facing problems financing its debt in the acquisition of Gannett. Should McClatchy fall, it could very well take dozens or even hundreds of city newspapers down with it. Similarly the Chicago Tribune went into bankruptcy in hopes of reorganization, but in the tough economy it may have a hard time getting back on its feet.

In addition to the poor advertising climate, newspaper and magazine publishers both are being challenged by the growth of the Internet, which is not only becoming increasingly competitive for advertising dollars but also is becoming the primary news providers for a significant segment of the population. In general, publishers have made the transition from print publication to the Internet with only moderate success, and the costs associated with producing news and editorial content make the move to Internet-only publication a dicey proposition at best for many publishers. Recently, Forbes become one of a number of companies that consolidated their online and print publication staffs, generally to the detriment of print, and this model will likely become the norm for many companies that had maintained distinct units before.

Book publishing is coming under similar pressures, in part because book sales are declining with personal disposable income. The common wisdom that books are "recession proof" is being significantly challenged ... with book retail costs now in the \$25-\$40 range for most hardbound books and in excess of \$10 for paperbacks, books are no longer "inexpensive" alternatives to other forms of entertainment, especially since the cost of a couple of books is now on the same order as most broadband internet connections.

University and academic publishers are facing even more problems in this regard. Textbook costs in excess of \$100 are not uncommon, and the last year has seen returns for textbooks exceeding 90%, serving to drive up prices even more. Meanwhile there has been a quiet revolution in universities as professors move the content for their courses online, in order to keep the costs manageable for their students.

Technical book publishing is under the double whammy of facing the same costs as other book producers while at the same time facing a technical audience that is becoming increasingly comfortable with finding documentation and related content about technology on the Internet. In some respects, the technical publishers may have an advantage - being closest to the technology, they have also been most active in experimenting with alternative delivery mechanisms and different business models, but it is likely that such technical book publishers will increasingly seek to position themselves as providers of online instruction, perhaps as the new breed of certificate oriented online universities. [Editorial note: O'Reilly is doing this with it's O'Reilly University programs - more information on this shortly].

Amazon and related companies are also beginning to finally hit their stride, and in many ways are effectively becoming the virtual surrogate of the publishing world. Amazon recently acquired Victoria, BC based Abe Books, one of the largest online distributors of used books, and has been heavily pushing its e-publishing and print on demand services. The Amazon Kindle meanwhile seems to be succeeding to become the "must-have" platform for the e-book reader (a lack of a single consistent platform has been one of the biggest detriments to adoption of e-books), and it is likely that as the price point for the Kindle drops, it may very well become the defacto standard for e-book publication.

2009 will almost certainly be the breakout year for e-books. Already a number of specialty fiction publishing companies, such as romance publisher Ellorah's Cave and science fiction publisher Moonlight (?) have emerged as major players in the e-book space, either in conjunction with publication of print books or exclusively specializing in e-books and print-on-demand titles.

Publication on the Internet continues to grow, and its worth noting that while even online publishing staffs are being cut, in general they are not being hit anywhere near as badly as their print counterparts in either book, magazine or newspaper publishing. It's likely that 2009 will see online production reduced slightly, but there's a fair amount of anecdotal evidence that publishers generally see their online divisions as being growth sectors once the economy stabilizes (companies such as Electronic Arts have announced some consolidation in their studios and layoffs, but they have been comparatively small compared to their overall workforce)

A similar phenomenon is occurring in specialty markets. Game publishers such as Wizards of the Coast have been increasingly shifting their focus towards their online gaming divisions, and overall interactive networked gaming seems to be the biggest growth area for many game companies, while comic book publishers such as Marvel or DC have been actively working to develop their commercial properties into interactive online games and related products (as well as increasing the synergy with their movie and animated production units).

The landscape at the end 2009 and into 2010 will likely look considerably different than it does today, with far fewer newspapers and magazine publishers still standing, and many book publishers having either folded imprints or gone under completely, while self book-publication, e-books and hybrid electronic forms (games and interactive media) will end up being far more prevalent. Indeed, it is likely that 2009-2010 will be seen by many in retrospect as the year that traditional publishing finally "died".

## **Energy Sector Faces Volatile Year**

Here in Victoria, my corner gas station has a liter of regular unleaded gas for CAN\$0.80, about US\$3.00 a gallon. Six months ago, a similar liter cost nearly \$1.50, more than \$6 a gallon when factoring in the dramatic change in exchange rates. While the driver in me is perfectly happy to see gas prices drop by 50%, the economist in me gets very, very jittery. It is very likely that this drop is an aberration brought about by uncertainty in the market, not just in the US but worldwide, rather than reflecting that significant a drop in real activity. Some shippers are parking oil tankers offshore, filled with oil, hoping to see higher prices, while others are trying to get out of the market now and are dumping oil at these prices rather than see prices drop further.

Oil producing countries are now faced with their own internal collapsing markets and are consequently shutting down production. Meanwhile, areas that were looking at developing fuels that are cost effective at price points about \$90, such as extracting oil from shale in Alberta, are having to stop production and wait, in many cases with projections halfway to completion, while exploration efforts in search of new oil are similarly being mothballed.

This slowdown also applies to the push for alternative fuels and power sources. There is a certain specific price point at which alternative energy technologies become feasible - if oil stays below this point, investment and research into these technologies slow or stop altogether.

Ironically, however, it is very likely that all of this will end up causing serious disruptions in both oil and electrical power supply, especially by mid-summer. It's definitely worth understanding that the drop in prices of crude oil is *not* actually reflective of a drop in demand. In the fourth quarter, demand

for oil dropped approximately 8%. Price, on the other hand, plummeted 75% from its high of \$140 to a low reached around Christmas of \$36. Most of this drop came as a result of investors selling off oil stocks that had been purchased as speculation because they were forced to cover their market positions. It's noteworthy that in the first few days of January, the price has climbed back to nearly \$50 (a nearly 40% rise) as real demand begins to push the price back up. This means that the existing oil stock that's already been transformed into distillates should draw down very quickly, especially as consumers see the lower prices as an excuse to drive more.

The problem comes once those supplies are drawn down - it takes a certain period of time, proportional to the time that oil producing facilities are mothballed, to bring them back online, to ship the oil, to process the oil into fuels and other goods and to distribute it, and as countries have nationalized their oil supplies over the last several years, those countries will likely wish to hold onto their oil for as long as possible in the hopes of driving prices back up.

Additionally, if the prices remain relatively depressed, this will make critical infrastructure improvements (especially after 2008's fairly severe hurricane season) less attractive. This means that by late summer, the possibility of one or more refineries being forced offline becomes fairly high, and the likelihood of new exploration to augment nearly depleted oil fields drops beyond the ten year window.

Unfortunately, these may prove to be major problems. The Ghawar field in Saudi Arabia, one of the largest and most productive in the world, is showing signs of running dry, especially since pressurized ocean water has become increasingly used to push oil to the surface, reducing the quality of the oil and causing damage to the delicate spongelike lattice of chambers that hold the oil. It's very likely that, even with depressed demand world-wide, Saudi Arabia may be unable to meet its existing contracts by the end of 2011 if not sooner.

The Mexican Cantarell fields are facing many of the same problems, and Mexico was expected to become a net oil importer by this year, given 2008's consumption levels. As Mexico is the second largest provider of oil to the United States (Canada is first, Saudi Arabia third) this has already begun having serious policy implications for Washington and has been the root cause of a fair amount of the political instability in Mexico. It's likely that Canada will also likely be reducing supply beyond that already committed given that the oil sands are not profitable to liquify at prices below \$85 a barrel.

Similar issues apply to natural gas, though there the US is probably better positioned than Europe. Most of Europe's natural gas comes from Russia. While the US has been hit hard by the financial collapse, it is in far better shape than Russia, which has watched the Ruble tumble in value compared to the Euro and has seen its primary export of oil essentially collapse to perhaps 30% of its value a year ago. This has made Putin considerably more belligerent (announcing earlier this week that after years of mothballing their nuclear arsenal that Russia will start producing nuclear weapons and ballistic missiles delivery systems for those weapons again).

This political instability will likely become increasingly common as standards of living decline precipitously throughout Europe, the Middle East and Southeast Asia (the violence in Greece over Christmas can be laid directly to economic conditions), which in turn is likely to further undermine oil development and exploration efforts.

There's a very real possibility that we are entering into a period of extreme oscillations in supply and demand, where oil prices go up dramatically due to spot political issues or infrastructure collapses, and then drop just as dramatically, making investment into oil based systems neither profitable nor even all that possible, at least for the near term future. This is a big part of the reason why reducing dependency upon foreign oil will be such a critical part of any administration's efforts - oil demand is one of the

most significant factors in the current economic crisis (though that dependency is far from obvious), and by reducing the overall dependence upon oil, the administration also reduces the exposure of the US (and secondarily of Canada) to the coming oil shocks.

Overall, it's likely that oil (and hence gas prices) will be all over the board, and it is conceivable that there may be spot shortages in certain areas as gas stocks run lower, pushing up prices and resulting in scenes similar to the rather agonizing time after Hurricane Ike significantly reduced refinery capability out of Texas.

### **Carbon markets heat up, but so does the weather**

Another area where the rise and fall of oil will have a big impact is going to be on climate change amelioration efforts. Reducing carbon emissions is a considerably more hot button issue politically when the price of gasoline is high, ironically enough, because people are more willing to cut down on their use for economic rather than environmental reasons.

As gas prices decline, so too does environmental consciousness and with it political will. If the Obama administration can keep the focus on reducing carbon emissions going, the country should be better positioned once the next oil shock comes, but this will require likely heavier subsidization of alternative energy deployments than may be presently planned.

One thing that that likely will get a major boost this year is the formal establishment of a North American carbon market, likely employing a cap and trade option where carbon credits become warrants (just like stocks) that can be bought and sold on the market. A company can buy warrants if it is likely to need to pollute more, and other companies or organizations can sell such warrants by introducing carbon reducing efforts such as reforestation. While such a system isn't ideal (the European market got off to a very rocky start) it's a critical first step. Look for companies such as Victoria, BC based CarbonNetworks to be significant players in that space.

This may prove critical in the next year. Despite the first "white christmas" for much of the US in some years (likely caused by the extremely quiet sun over the last eight months with no sunspot activity at all), I was a little surprised this year to find geese honking merrily away in the football field abutting our back yard here in Victoria. Visiting geese aren't that unusual - the field makes a natural landing spot for them, and it's not unusual to have several dozen of them in the field at any given time during the spring and fall - but Canadian geese are migratory birds. By late December, most geese have made their way as far south as Northern California.

The forecasts from the US National Oceanic and Atmospheric Administration (NOAA, aka, the Weather Bureau) for 2009 indicate that overall next year will be warmer than usual for most of the Southeast through the start of the year, then this area will moderate, while the West will end up being well above average temperature wise for until early Fall 2009. The winter of 2009-2010 looks considerably more warmer as well, as it will be well above average from the Mid-Atlantic States west as far as Arizona, and from Minnesota to Florida. Even the Pacific Northwest, which spent much of 2008 being at or slightly below normal temperatures, will not be completely spared, with temperatures above normal throughout the summer.

Additionally, this will likely be a very dry year through much of the US until the hurricane season in August through October. During that period, rainfall will be significantly above normal as well along both the East Coast and along the Gulf Coast. Drs. Philip Klotzbach and William Gray of the Tropical Meteorology Project also predict that there will be fourteen named storms this year, of which seven

will be hurricanes and three of which will be major (Category 3+) hurricanes. There is a 63% chance that at least one of those major storms will make landfall. This compares with 13, 7 and 3 for predicted for 2008, while the actual totals came to 16, 8 and 5 storms respectively. *Gray and Klotzbach have been successful predicting hurricane seasons for more than twenty five years.*

It is thus likely that another Hurricane Ike or Hurricane Katrina may happen again this year, with the correspondent attendant disruptions - especially given that Klotzbach and Gray's predictions indicate about 5% more activity than last year and more than 20% above the average level. From the standpoint of developers in the Southeast US, its beginning to look like planning for workplace disruptions due to severe weather are almost becoming normal, sad to say.

## **IT departments disappear into the Clouds**

While other IT sectors may be struggling, one area that will likely be quite hot will be in the cloud computing/hosted services market. This particular market has been the subject of a great deal of hype over the last year, but it is likely that the overwhelming factor in cloud adoption this year will less be promotional marketing than it will simply be cost.

IT plants are expensive - they take up significant real-estate, require a significant amount of air conditioning and electrical hardening in order to take power loads and properly disipate heat, they require system administrators monitoring and repairing these systems and have to be replaced and reconfigured periodically as systems go down.

The principle benefit of cloud computing (or at least of hosted services) is that much of this cost is effectively offloaded to the hosting service. Combine this with the development and deployment of hosted applications (which are increasing both in capabilities and reliability), either VPN or web-based, and what emerges is a very compelling story for many companies that are struggling trying to contain IT costs while tightening their belts.

Most cloud setups typically consist of "supercomputers" that are built as hundreds or even thousands of commodity server units that share memory and processing power and that are in turn tied into large scale storage arrays. Within this sea of memory and processing power, its possible to launch various "instances" – virtual machines that use the resources of the host system but exists as its own, for the most part independent "computer".

What's more, because these machines are effectively software only computers, they can be saved as if they were computer documents, then can be reloaded later, starting off once reloaded at precisely the point where they were saved. This means, consequently, that it becomes possible to create templates that can be stored then automatically loaded later whenever a given application (such as an operating system or configured database) needs to be restarted. These applications are known as appliances, and appliances make possible all kinds of interesting computing within the cloud.

Many of the larger traditional hosting providers have started testing the virtualization waters, in many cases able to offer virtual computer systems to developers at a fraction of the cost of a dedicated server. Virtual machines tend to have a somewhat slower performance ratio that a typical dedicated system, as there is a virtual machine layer between the virtual computer and the real one that takes CPU cycles, but as understanding of how virtualization works within clouds has improved, so too has the performance metric.

My projection is that those companies that are in the space right now (Amazon, Google, Microsoft as well as several of the larger hosting companies) will do quite well this year, in great part because

moving IT departments into the clouds can provide significant cost savings. In essence, companies would be outsourcing their IT departments, letting the host take care of the electrical bill, the infrastructure maintenance, bringing new systems online or pulling systems out when they are no longer functional.

As financial pressures (most specifically the collapse of the corporate real estate market as phase two of the great mortgage meltdown) continue to mount, a lot of companies are going to be at a point where their income is low enough (or their lines of credit have been diminished badly enough) that they can't continue meeting their monthly lease payments, and will be either downsizing their physical plant or in some cases ditching the real estate altogether and going to a nearly virtual company.

A second factor that will start factoring into cloud virtualization will be issues of electrical grid support. The NOAA forecast indicates that by the summer of 2009, temperatures across the US and Canada should be well above average, especially in the Plains states, the Southeast and the Southwest. While the economic slowdown may cut down some on electrical usage, the grid is likely to be stressed considerably by air conditioning, and rolling brownouts are likely to become more common (the current fiscal problems in California may very well add into that).

An advantage to developing distributed but concentrated computing centers is that these facilities can be "hardened" against adverse conditions. Such facilities usually set up power generators to keep systems running and that let them gracefully bring them down non-essential systems, rather than having the power kick out all at once. They can dedicate more resources to cooling computer systems and better handle the problems of heat generation (a number of pilot projects are in fact using the heat produced by server farms to drive heat pumps within buildings built from the ground up with server farms in mind). They can also more effectively negotiate power rates and draws with electrical utility companies that thousands of individual companies can do.

As the worst of the immediate crisis abates then (near the end of 2009 or into 2010) its likely that such cloud centers will become more attractive from an energy efficiency standpoint. Either way, it is increasingly likely that both public clouds (such as those of Amazon, which let anyone subscribe to the cloud centers) and private clouds (typically cloud centers that have a few large, dedicated clients) will become far more prevalent, especially for the small and medium size business. Microsoft's efforts with Azure should also start bearing fruit towards the end of 2009 and into 2010 (expect there to be a strong synchronization between Azure, Hyper-V and the next Microsoft Windows operating system).

### **Application Services come into their own**

As cloud computing goes, so do two complementary technologies – application services, and web services. It's easier to split these into two distinct sections, though it should be kept in mind that they are simply different manifestations of an overall move towards distributed architectures that has been ongoing for the last several years.

Applications services consist both of traditional desktop applications that are increasingly being "replicated" on the web, and larger scale, typically data-centric applications that are usually targeted towards a specific vertical market like customer relationship management (CRM) applications such as Salesforce.com. This category of services also contains what I see as an increasingly dominant theme for 2009 – community management systems.

Webmail has long been the "success story" for application services to the extent that standalone mail clients such as Outlook are being increasingly replaced with web-mail based counterparts (including

Microsoft's entries into that area). This is a profound shift from a services standpoint, because as mail moves primarily into this space, it can be aggregated with other messaging services – syndication feeds, chat services (increasingly across XMPP), XML messaging via SOAP and so forth.

Yet web-mail is also important because such mail applications usually tend to occupy a significant place in the work-flow for most people, and the more this moves out of the realm of dedicated desktop devices, the more that it diminishes people's use of other non-desktop systems in favor of browser oriented ones.

While web-mail went mainstream in 2005-06, other pieces of the traditional "work desktop" – word processors, spreadsheets, personal information managers, presentation creators and vector drawing tools – seemed to reach a similar tipping point in 2008. Google formerly released its [Google Apps](#) suite, consisting of a word processor, notebook system, spreadsheet editor, and presentation tool, perversely set using the Open Office ODF formats as their defaults. [Zoho](#) released a similar (and more highly powered) set of tools to accomplish these same tasks, again (as with Google) for free for individual users and at a low subscription rate for dedicated corporate accounts.

The uptake for these web-based desktop applications has been, until recently fairly slow. Part of this had to do with the ability of these applications to work offline, a capability that's critical to the success of application services. Most suites now recognize this, and are utilizing different tactics – from persisting critical pages within browser embedded databases to writing modular extensions for browsers to give enough support for these tools to let them work well in an offline setting. As this capability has improved, so to has the uptake of interest in these tools.

My expectation for this class of applications is that they will continue to see steady growth and adoption through 2009, though not really explode in usage until 2010. One reason for this is that the cost argument, so critical to general cloud computing in 2009, is not as applicable here. While high priced office suites will certainly be hurt by the slowdown (expect significant drops in the price of Microsoft Office as the recession worsens), there are enough freely and inexpensively priced desktop based office systems already in deployment that the cost argument does not become as big of a factor, and the other arguments – ease of access across multiple systems and the ability to work collaboratively - I see being much less significant through this year.

By mid-to-late 2010, however, I see broadband access actually improving significantly, as WiMax and related networks get the bugs worked out of their systems, as the cellular networks and cable networks begin to merge significantly and as more money gets pumped into boosting broadcast Internet coming from the Fed and the states in order to try to jump start an anemic employment scene. This will have a major influence on the application services space, including the desktop analogs.

The second class of applications that I see making a huge jump in the next year are going to be those dedicated to specific industry verticals. My gut feeling on this is that there's been a major shift going on in distributed systems as various XML industry standards have finally stabilized. These applications are going to be one driven by XML databases or XML/SQL hybrid databases, utilizing declarative constraint languages in order to build and edit XML documents in specific language domains. While a lot of the foundational work is in place now, I really see a lot of these begin to emerge in concert with government infrastructure efforts, as these applications actually work best in environments where a number of different actors have to interoperate.

I'll have more to say on these when I talk about web services and SOA in the next section.

The third big class of applications I see, as mentioned before, is in the community software space.

We're reaching a stage where the blogging and community tools that were first spawned in the early part of this decade have reached a deep level of maturity, have garnered rich development communities, and now have hundreds or even thousands of community-developed extensions to act as building blocks. Applications such as Drupal, Movable Type, Joomla and others, as well as corresponding commercial tool-sets such as Microsoft Sharepoint, which I see continuing to evolve and mature over the next year, possibly merging with the broader Azure initiatives.

On the other hand, what I don't see much of is non-IT in-house software development. Beyond the financial pressures there (though that will be a huge factor this year, of course) I also think that we're reaching a stage where application development increasingly involves threading data streams through filters, processors and viewers. This is a big part of what AJAX programming is all about, after all. The era of huge, enterprise-wide monolithic applications is nearly over, and with it the primacy of the IT department and the need to create the massive projects that consumed so much time and money in the last decade.

## **The web services era begins in earnest**

(Warning, this gets technical).

This may seem a rather odd statement – after all, "web services" in the traditional SOA sense have been around for the last decade, give or take a few years. I believe, however, that while there have been some real (and important) success stories in that period, for the most part we're only just beginning to understand what distributed services is all about.

In the desktop application era, the dominant theme was efficient coupling of interfaces. As an application developer you focused on performance, on building as much of the model directly into the application as possible, and on providing as many features as you could manage into the application, in order to make it appeal to the broadest potential market. For desktop applications, these are still appropriate considerations.

The *web services* era, or perhaps more accurately, the *distributed services* era that we are now entering into, has considerably different priorities of focus. In most distributed services, it is likely that different nodes within the network may very well utilize different vendors or open source tools in order to perform the processing on their end. This in turn places a much higher premium on interchangeability of information, which means that simple messaging and conformity of interfaces becomes more paramount than robust feature sets. (It also makes it harder, and generally less desirable, for one vendor to provide "all-in-one" solutions.

Desktop applications were typically intermittently asynchronous – user interface components typically had to be asynchronous and event driven, but most of the underlying logic was still built assuming a high degree of synchronization of libraries. In a distributed application, asynchronicity – a lack of knowledge about when a given packet of information would arrive to be processed – is the rule rather than the exception. This means that the dominant form of operation in such systems are message queues that process information independent of when that information arrives.

Desktop applications typically baked the logic for the application within the binary code for that application, in order to facilitate the most efficient turnover of content. Distributed applications, on the other hand, usually work far better by treating the business rules for the application as a data streams themselves, streams that can be changed depending upon other factors in the system. Such switchable logic isn't as efficient in terms of processing speed, but it is usually far better when dealing with a

complex business environment.

Desktop applications are usually built assuming that state changes exist on a finite set of atomic properties, and the role of document serialization is simply to freeze those properties then thaw them out at some later point. Distributed applications, on the other hand, actually work upon the assumption that documents themselves are the "atomic" properties of the system, and that the role of the messaging system is to move various and sundry documents between processors, document producers and document consumers.

This is a subtle shift, and represents a different level of programming that is closer in spirit to systems architecture. It is also, generally a complementary relationship to more traditional programming, as the processors themselves still need to be written by traditional programmers, but these processors become increasingly agnostic about the business logic that runs over them. This holds true whether that processor is a web browser, a router, a transformation engine or data filter.

Notice in all of this that I haven't mentioned XML once. This was deliberate. My sense is that what seems to be emerging on the web now is a paradigm of macro-document formats and micro-document formats. Macro-documents are typically narrative and sequential in structure, have contextual identity (a document can be uniquely identified by some form of "name") and are usually fairly deep. Micro-documents, on the other hand, are usually unordered or semi-ordered bags of properties (each property of which is uniquely named within its siblings), typically do not have a unique identity, and usually employ a combination of hash tables and arrays to store content.

Whether that combination is XML and JSON or some other set of formats is of only peripheral interest, although it is my suspicion that both are sufficiently well entrenched at this stage that no other formats are likely to get traction for the next decade anyway (I think when the semantic web reaches its own level of maturity and mass adoption that we'll see something else emerge, but this will be at a different level of abstraction).

In 2008, I saw a number of different people come up with the same realization at the same time that syndication, which employs the concept of REST at a very profound level, was actually a remarkably good carrier of more information than simply blog posts. About the same time, the final version of XQuery began to appear within both purely XML and XML/SQL hybrid databases, along with effort finally being made to address the issue of an update standard for XML content (the XQuery Update Facility, or XQUF – okay, the name isn't exactly euphonious).

What's more, people have begun to combine such XQuery content, whether serving up Plain Old XML (POX) or syndicated content via Atom or RSS or messaged content via SOAP, with an XML enabled front end client – XForms, obviously, but many other tools are also moving into the mode of an XML model driving a writable user interface. My friend and colleague Dan McCreary termed the term XRX – Xquery/REST/XForms for this assemblage, and overall I think this seems to be the meme that's actually sticking in discussion.

However, the idea here is that this approach, which uses the concept of XQuery collections – either in a database, a file, a syndication stream or some other resource – as vehicles for passing both content and links to content to an XML-enabled editor that can then roundtrip these documents back to the server, is something that is intrinsic to the web, something that is already having a huge impact on the way people write applications. Overall, the concept is one that seems to be gaining the name *RESTful Services*, thanks in great part to a book written by Sam Ruby and Leonard Richardson by that name.

In many respects the concept isn't new – after all, much of AJAX employs this model, one in which

components use the XMLHttpRequest object to change the state of the components within a web page – but XML RESTful services in particular work at a somewhat higher architectural level, especially since XML (and HTML, the *almost* XML) provide the substrate on which most JavaScript services operate.

I think that RESTful services will be a major area of focus, especially toward the latter half of 2009, in the industry. As a concept, it sees the web as being very much like a database, and this approach I think is resonating with people in a way that SOAP based remote procedure calls, in its attempt to turn the Web into a sea of COM objects, failed to do outside of very sandboxed environments

I don't see SOAP going away, however. It is a messaging format, and one designed specifically for high reliability, high security transport across a wide range of protocols, though I sometimes wonder if in the attempt to make it work across protocols the designers of SOAP didn't make a mistake – trying to build the ultimately conformable box when all that was needed in most circumstances was an envelope.

I see Atom emerging as the RESTful counterpart to SOAP, at least over the HTTP protocol.

On the other hand, I think that a format to watch carefully over the next year is the Extensible Messaging and Presence Protocol (XMPP). XMPP was designed as the message transport protocol for the Jabber project. XMPP is actually a number of different messaging formats as well as a communications protocol, and the idea behind it is that it is intended to facilitate connected synchronous communication, rather than the stateless asynchronous communication that's a hallmark of HTTP over TCP/IP. Because of the way that this protocol is set up, XMPP messages can generally be transferred far more efficiently than either SOAP or syndicated Atom. Not surprisingly, organizations that need real-time messaging systems are looking at XMPP very closely, especially since it can, in theory even be used as a transport protocol over which Atom or SOAP can ride.

I look to see XMPP becoming a critical part of the web developer's toolkit by the end of 2009, as he or she becomes more comfortable with generalized messaging architectures.

## **Syndication forms the backbone of the Writable Web**

The syndication model has long been a major facet of the way that the web works, but for the most part its been a largely single direction notification mechanism – you publish content, this updates a syndication queue, then next time you query the queue you'll see the new content.

Increasingly, however, programmers are beginning to recognize that "news feeds" are messages, and as such they can be used as a way to initiate actions elsewhere. One facet of that, hinted in the previous section, was the notion that you can in fact contain messages – documents - within the payload of a given Atom entry that can in turn create or update a collection of such content. Moreover, one area where most SOAP proponents are also moving is the idea that if you do work with message queues that it becomes the responsibility of the queue processor, not the publishing system, to take that content and handle the interpretation of the content based upon other factors, typically those contained in the message itself.

Jeff Barr of Amazon [recently pointed out](#) a new service that I find fascinating – [Tarpipes.com](#). This service uses a combination of email (itself a messaging service) syndication feeds and social network APIs to make it possible to create "programs" in which you can do things like post an email with a picture enclosure to a drop box which will then update that image to Flickr, send out a twitter to that image and send a confirming email back to you, all through various syndication services (I tried it myself and found that while some of it works, a few components are still in development).

Intriguing to me is that Tarpipes represents the concept of the mashup in reverse – rather than using client components within a web page or widget that draw in external data feeds, Tarpipes effectively creates a "mash-down", creating complex orchestration across disparate services using feeds. I expect to see more of this (and more concern about the security implications of what this represents). The irony here is that what I see emerging in all of this is ultimately what SOAP and Web Services promised in late 1990s and early 2000 – the ability to control and update complex distributed systems via messaging – albeit with a different set of messaging protocols and a very different architectural vision.

Syndication is also at the heart of the writable web in other ways. The AtomPub protocol, which I wrote about extensively in last year's analysis, is definitely getting picked up now by players all across the industry, from Microsoft to Google to IBM. At the 2008 Web Services East conference this year, I was gratified to see IBM showcase a syndication server, built around Atom and AtomPub, that figured very nicely in this model (I believe it is now out as WebSphere Portal, but I'm not sure), though there was still too much strict client/server there.

I think one big story in 2009 will be the rise of orchestrated synchronization. Orchestration, getting asynchronous processes to work in a fashion that still ultimately produces a meaningful result, is hard, especially when those processes work across a network as broad as the Internet. I think pieces are emerging there now, however – watch the [W3C XProc Working Draft](#) in particular ... while XProc still tends to be a little too tightly focused on synchronicity, it is beginning to break out of this viewpoint as people start thinking about the asynchronous side of things.

I met two of the three editors of the spec for the first time, Norm Walsh at the [2008 MarkLogic Conference](#) in San Francisco, Alex Milowski at [Balisage 2008](#) in Montreal (I've corresponded with Henry Thompson of the University of Edinburgh over the years, but haven't yet met him, alas). Each of them see XProc as fulfilling a very real niche by not only making it possible to create pipelines (and to blackbox pipelines for abstraction purposes) but also to make people understand that ultimately such pipelining architectures represent the real future of messaging and syndication services.

XProc is in its Candidate Recommendation status now, so, unless something radical changes between now and then, it's likely that XProc will end up being a full recommendation by March 2009. On its heels is another W3C spec worth looking at – the Service Modeling Language (SML), which despite its name (the W3C really needs someone coming up with more appropriate names) is actually an official endorsement of the use of Schematron within the W3C XML Schema Definition Language (XSDL). Schematron, written by O'Reilly blogger, standards expert and [avowed hermit](#) Rick Jelliffe (sorry, Rick, couldn't resist), provides a constraint language to be used in conjunction with other schema languages that makes it possible to use XPath to determine whether a given element or attribute pattern holds a business-logic valid value. Combine SML/Schematron with XProc and you have the ability to create extraordinarily rich pipelines capable of processing according to dynamic business rule changes.

Expect to see a number of working applications built around these principles well into 2009.

One final area to watch in this space is [Twitter](#). While I'm not exactly a hard core twitterer ([http://www.twitter.com/kurt\\_cagle](http://www.twitter.com/kurt_cagle)) I've begun trying to make sure that I pass on those cool things I find each day over twitter, not only as a means of communicating with others but as a relatively painless way to bookmark these with context. I see twitter in conjunction with tarpipes being a powerful tool – in essence you can launch cascading services directly from your browser (I actually have two twitter editors in Firefox that I alternately launch from). Again, while I think that twitter will stay largely confined to the blogosphere, its utility as both status message carrier and command line shouldn't

be overlooked.

## **XForms and XML-enabled clients gain traction with XQuery databases**

I'm beginning to despair about XForms, which is perhaps a good sign. XForms is perhaps the oldest of the W3C technologies that has yet to either die completely or really dramatically take off, and for all that it has some real beauty of design and thought to it there are pieces of XForms which are just plain ugly.

2008 has seen a fair number of these ugly edges of XForms sanded down, however, and the XForms 1.1 specification should actually become a formal recommendation early this year as well. These should help resolve a number of the glaring holes in the 1.0 specification and offers at least the promise of integration with XPath 2.0 as well (which is in fact a huge step forward).

More to the point, however, XForms has now become the focus of not just one but several different implementations. [Orbeon](#) has continued to establish its position as the dominant player in the nascent industry, though XForms products by IBM ([Lotus Forms](#), formerly the rather unwieldy IBM Workplace Forms), the XForms extension for [Mozilla Firefox](#), the Server/AJAX based [Chiba XForms](#), [FormsPlayer](#), [Picoforms](#) and the [Ubiquity](#) project, all attest to how full (and competitive) this field has become. Additionally, tools such as xFy

There are two factors that have long held XForms in check. The first is that XForms is not a complete solution to anything – it depends upon having an XML data server that can also persist or process the XML sent to it. Without that, XForms is simply a very complicated toy language.

I do not find it at all coincidental that as XQuery has emerged to provide a data abstraction layer for document-centric content, XML databases have also gained traction, and 2008 provided a good example of that. MarkLogic released their [MarkLogic 4.0 XML Server](#) in September (I [reviewed it last November](#)), bringing their XQuery implementations to contemporary standards (and hired Micah Dubinko, noted XForms guru and ). The eXist project (a personal favorite of mine) released their [1.2 branch](#) this year as well, giving a significant boost to the performance of the product and expanding its native library significantly. IBM released updates to their widely used [IBM DB2 PureXML Server](#) to bring it in line with current XQuery specifications, and has significantly stepped up their marketing efforts there. Oracle's acquisition of Sleepycat in late 2006 resurfaced in 2008 as [Oracle Berkeley DB XML](#) while EMC/Document acquired XML database company X-Hive to incorporate into its offering as the EMC's [X-Hive/DB8 Server](#). I'm planning on reviewing all of these, along with [DataDirect's XQuery bridge](#), later this year.

XML Databases drive XForms. That has been demonstrated over and over again in this industry – indeed, much of the XRX patterns described earlier in this series are a direct result of this realization. Once you have an XML database in place, the desire to work with the content via some type of forms interface becomes very compelling, especially for mixed data/document structures.

This realization has obviously occurred to those buying up the XML databases as well. IBM bought up Pure Edge a couple of years ago, in anticipation of this, but more recently, they and FormsPlayer creator Mark Birbeck joined forces in order to build a new open source XForms engine called [Ubiquity](#). Using a progressive capabilities architecture, Ubiquity is an AJAX based client engine that is intended to be run in any browser that has contemporary JavaScript support. Given that the project started early last year, the progress that Ubiquity has made in implementing not only an XForms engine but a SMIL layer and RDFa as well is nothing short of stunning. This is definitely a project to watch in

2009.

My expectations for XForms in 2009 are consequently quite high. I think that by the time the year is out, every XML database will be paired with either an open source XForms component or will have produced an embedded XForms layer of their own, and most of these will be running at least a subset of XForms 1.1. Because many of these are designed to work in an AJAX layer on top of existing browsers, it also short circuits the other major block to XForms adoption – Microsoft's lack of interest in the technology.

I think another company to watch this year in that space is [JustSystems](#) with their [xFy](#) product, though I personally think that they should make the investment and move to a full XForms compliant system (they have something similar at the moment, but as IBM found with their PureEdge solution, the market seems to be moving away from pure proprietary languages, even XML languages, and the benefits of maintaining an XForms layer will likely outweigh the development costs).

### **Semantics continues to not be RDF, but enrichment, classification and taxonomy**

Within the realm of computational semantics, there is still a fairly broad disconnect between triple pair semantics, the use of RDF (or turtle notation) to create atomic assertions, and the realm of semantics as reflected on the web. I do not expect this to change much in 2009, save perhaps that the gulf between the two will likely just get wider.

While I think that RDF (or more likely a successor to that set of specifications) will eventually go on to becoming the overall semantic tier of the web, its rather depressing just how far we really are from RDF actually becoming widely adopted. Instead, the approaches themselves are still running largely into the proprietary realm, though there are a few interesting areas that should be watched fairly closely.

One of the open source projects that received a fair amount of buzz at OSCON 2008 was Freebase (<http://www.freebase.com>), a site which mines Wikipedia in order to establish an extraordinarily comprehensive RDF database about topics gleaned from the linkages found within Wikipedia. One benefit of this is that you can do such queries as "List all movies containing aliens", and with the tools at hand, it will show content that matches this particular query. This in turn makes it possible to create relational queries on Wikipedia data, making it particularly useful as a research and data mining tool.

My sense is that you'll see more of these types of applications showing in 2009, apps built around the power of RDF (and increasingly SparQL, the RDF/OWL query language). However, its also likely that few, if any, of these sites are likely to tout their Semantic Web credentials (or even acknowledge that this is what is going on under the hood).

One area that I feel is poised to really take off in the next year is content enrichment. Enrichment involves taking a collection of text, running a series of rules and contextual filters on the data looking for names, events and patterns, then encasing this content within specialized XML markup. Depending upon the database, the source, and the service agreements involved, such enrichment performs an invaluable service in being able to establish the context of a given phrase within an article, and by extension being able to provide both an abstract of the content and specialized search looking for meta-content within a document.

For instance, an article about Barack Obama and John McCain could be abstracted as being about the presidential contest, while an article about Barack Obama and George Bush might talk about transitions of power from one president to the next, with specific terms for each of these people (and related

people determined by this context, highlighted as tagged content).

Again this is a service that both commercial and open source XML databases and content management systems are beginning to provide to their customers, and this also illustrates what is increasingly becoming the norm in business applications, situations where critical processing of data streams are applied through web services by third party providers.

This is an area that is ripe for standardization. I suspect that the RDF crowd will probably be jumping up and down at this stage screaming "Use RDF! Use RDF!!" but I'm not really sure that will end up happening, at least not directly. I wrote last year about CURIEs and RDFa, which is an attribute-carried RDF descriptor language for text content, and with the specification now made into a full Recommendation (as of October, 2008), I suspect that it may start making its way in as an alternative offered format by many vendors, which raises the very real possibility that it could become the de facto standard for enrichment (or form the foundation for same) by late 2010.

My central problem with RDF is that it is a brilliant technology that tried to solve too big a problem too early on by establishing itself as a way of building "dynamic" ontologies. Most ontologies *are* ultimately dynamic, changing and shifting as the requirements for their use change, but at the same time such ontologies change relatively slowly over time.

This means that the benefit of specifying a complex RDF Schema on an ontology - which can be a major exercise in hair pulling - is typically only advantageous in the very long term for most ontologies, and that in general the flexibility offered by RDF in that regard is much like trying to build a skyscraper out of silly putty. It's possible to do so (maybe), but the drawbacks in the increased complexity of code (especially given that most people are still having trouble understanding the relatively simple syntax of XPath) makes it a dubious proposition at best except for those highly interconnected information spaces with comparatively few constraints acting on it such as Freebase.

What I see happening instead is that there should be fairly significant consolidation of specifications down to a few consortia standards in any given domain – such as XRBL in business reporting, HL7 in health care, S1000D in airline specifications and so forth. Even five years ago, most industries tended to have two or more distinct standards competing for adoption, but in the last year many of these dual standard industries have either settled on one or merged these two standards together. Thus I see 2009 being devoted towards application development around an industry's preferred vertical ... with opportunities especially for those who work in developing such standards in the first place.

In other words, its very likely that in order for the RDF/Semantic Web approach to gain credence in these spaces, ontologists will have to start with these specific industry schemas and develop RDF-based tools that model them. Given that I see XML databases increasingly carrying the load in working with these schemas, this will also likely result, at some point in the not too distant future, of a need for a meeting of the minds between the XQuery working group and the SparQL working group in order to develop a SparQL analog that can be run in XQuery, probably as a set of optional modular extensions to the language. I don't know if this is on the agenda at the W3C yet, though if its not, then its likely we won't see significant traction there until 2011 at the earliest.

The other area where there's been something of a "small s" semantic revolution has been the growing awareness of the intimate link between web navigation and knowledge navigation among both web developers and semantics specialists. As web sites grow, they become more complex, deeper, and far more difficult to maintain in terms of their underlying structure.

Ultimately this comes down to a question of classification and partition of the topics within the site

itself, and this in turn points to a potential semantic solution for managing large and topically interconnected content. The folksonomy "tagging" revolution (which I think is probably running out of steam) was a significant first step, but folksonomies are by their nature unstructured and poorly regulated.

I think this is going to be the year that a lot of both web design and web framework support is going to embrace semantic tools and concepts (the inclusion of RDF support within the taxonomy-heavy Drupal system is a good case in point).

## **Internet Explorer Fades, Firefox Stays the Course, Google Chrome Surges**

Poor IE. Like the late comedian Rodney Dangerfield, it seems to have a hard time getting much respect these days. Within Microsoft it has long been the unwanted stepchild – ignored when Microsoft shifted gears towards server-side technologies in the late 1990s, Internet Explorer was intended to fade away into the woodwork with the coming of Longhorn (now known as Vista), but the rebirth of Netscape Navigator as Mozilla Firefox caught the company off guard and the sudden shift to concerns about security after 9/11 put into the bind of having to improve its browser just to be seen as staying competitive.

The year 2008 saw the next major salvo in the "Second Browser Wars" with the emergence of Google Chrome. This locks Microsoft into a frustrating position – while it is possible to view Mozilla as an annoyance, Google entering the fray forces Microsoft to either radically improve their browser or to fade out altogether, at a time when the stand-alone application market, even with web services capabilities, is being replaced whenever possible with web-based alternatives.

Between them, Mozilla and Google have the potential to force Microsoft below the 50% threshold of adoption by the end of 2009, though I think a likelier target would be mid-2010. This is actually even worse for Microsoft than it sounds, because a significant percentage of Microsoft's deployments are on legacy systems or on OEM machines that may be present, but used comparatively rarely. As of Q4, 2008, IE has 68.15% usage, Mozilla 21.34%, Safari 7.93% and Chrome 1.04%, according to Net Applications.

I think that the Safari numbers may improve a little bit as Apple laptops continue to eat into Microsoft market-share, but Safari's penetration is very much dependent upon the relative size of the Apple vs. Microsoft markets. I think that Mozilla's also probably near the top end of its penetration range, perhaps improving another 4-5% points overall to become a consistent quarter of the market.

Google Chrome, on the other hand, has the potential to do some damage. Overall, Chrome's performance is somewhat better than a Mozilla Firefox and considerably better than IE7, assuming no encumbering extensions. However, those extensions are where Firefox has the edge on both IE and Chrome right now – the add-on mechanism for Firefox makes it possible for Firefox to be configured to do just about anything, and while IE also has an extension mechanism, it requires considerably more programming expertise in order to build these extensions ... which will be even more true once [Ubiquity](#), Mozilla's browser "command-line" reaches a stage where it can be deployed as an integral part of the browser.

Extensibility is becoming a key factor in the browser wars because it draws in the community as a participant in the development of the browser itself. Without the thousands of extensions, Firefox would likely have stalled fairly quickly in 2006.

In December 2008, Aaron Boodman, the founder of Greasemonkey and de facto speaker on all matters

Chrome at Google, revealed that while the initial release of Chrome does not have an extension mechanism, this was not an oversight – pointing to a design document that showcases [Google's extension strategy](#). I expect to see extensions surface by late Spring, 2009, designed to work within Chrome's architecture, designed to be simple to build and to share, and designed to be sufficiently contained so as not to impair the security of the browser.

Once this happens, I think that you're going to see a major explosion in the usage of Chrome, partially by developers who aren't happy with either Microsoft or Mozilla, but more-so by a cadre of new developers who see this as a potential platform (or market) on which to build. On the other hand, I don't necessarily think that this will be a unique group – up to a certain point, developing extension components is likely going to be similar enough between the platforms that you'll see extensions designed to work for both Firefox *and* Chrome.

Moreover, I can see a move on the part of both organizations at some point to develop an API for component extensions that can be standardized as well, which in turn will likely also be supported by both the KDE foundation and possibly by Apple (Chrome, Safari and KDE Konquerer are all built on WebKit, so such standarization between these organizations would make a lot of sense). I don't necessarily see this playing out much before 2010, but I'm reasonably confident that it will happen.

There are two additional factors that should be discussed while focusing on browsers. The first is the likely release of ECMAScript (Javascript) 3.5 early in 2009 which will reflect the next "baseline" of JavaScript development. This work, code-named Harmony, was scaled back considerably from the rather extensive attempts made to refactor the language on the part of Mozilla in ES4, but this baseline is important because it will likely indicate the foundation set of AJAX capabilities for the next decade in all browsers. It's debatable whether IE8 will include a Harmony-compliant JavaScript engine though I'm inclined to believe they won't. I suspect it more likely that this may be an IE8.1 feature delivered as an auto-update.

The second major factor will be the release of HTML 5, which may happen in 2009, but more likely will be released in 2010, though the browser vendors will no doubt be integrating prospective features in point releases prior to that. HTML 5 is the first major revision to the HTML specification since the 4.01 specification in 1999. It incorporates a number of new tags as well as recognizes XHTML as being a legitimate "variant" of HTML ... and that all browsers will need to be able to support both forms of syntax.

One of the more intriguing aspects of HTML5 has been the fact that the XForms working group has formed a joint committee with the HTML working group for the purpose of creating an XForms for HTML 5. This will make it possible to set up XML model instances, to bind content to HTML elements and to otherwise provide a lot of the same functionality that has been intrinsic to XHTML+XForms into HTML+XForms, at a considerably lower barrier to entry for developers. This should provide even more fuel for the growth of XForms within the 2009-2010 period.

I will be doing an analysis of HTML 5's features in an upcoming article for O'Reilly later in the winter, so won't cover specific features here. However, given the broad consensus of companies working on HTML 5, its looking increasingly likely that the move to HTML 5, especially if it is done in concert with the release of ECMAScript 3.5, should provide a much higher degree of interoperability between browsers by late 2010.

I'll add one more prediction into this particular section – this is going to be a hard year for producers of JavaScript toolkits, and considerable consolidation in the field, especially following last year. While perhaps one of the most mature, the Yahoo GUI interfaces will probably fade fairly quickly as people

lose faith in Yahoo's ability to support it, and my feeling is that the prototype based toolkits such as scriptalicious will run into problems as Ruby hits a glass ceiling of development. I'm not a Ruby developer, so will stay away from predicting too much about the language, but I do know that there are a number of signs that indicate that Ruby has exceeded its appropriate market somewhat, and the general trends for Ruby at this stage are for a slowdown of usage, though not likely a wholesale abandonment of it.

Overall, I think that the ones best positioned in this space moving forward are those with significant backers – jQuery (with strong support from Microsoft), ext JS (which has an extensive customer base), Dojo, Nexaweb and Tibco likely all have enough legs to survive through the shakeout. To me, AJAX libraries and frameworks are beginning to approach "mature" technology, and as such, is really just beginning to come onto the radar of enterprise buyers.

I did have an interesting conversation recently in which the topic of AJAX support in the Enterprise came up, and one of the things that I'm definitely sensing is that the primary reason that you're not seeing the uptake at the Enterprise level has much more to do with AJAX's ability to work with minimal change with the XML messaging architecture (while consumer-level "mashups" are messaged in the main over JSON, the enterprise is unequivocally XML, and I see no real shift away from that stance any time soon). This is one of the reasons why I suspect that one of the key directions of AJAX library development will be in that area of broad data model integration, and until that happens, enterprise adoption will remain tepid and largely confined to read-oriented "dashboards" rather than full read/write capabilities.

## **Some thoughts for a New Era**

This particular look forward is definitely longer than what I have written in years past, and for those of you who have managed to wade through the admittedly voluminous text I both admire your fortitude. This has been a hard report to write – both because of the level of detail that I felt needed to be covered, and because there are relatively few bright spots that I see happening within the next twelve months.

It is possible that things may ameliorate quickly – perhaps because we are increasingly aware of what could happen, its possible that we may in fact be able to blunt the edges of a recession that some are predicting may be as bad as anything we've faced since the 1930s. I'd like to believe that, though frankly I'm not optimistic. Programmers have been through recessions before, but a surprisingly large number were not even born the last time the economy got even remotely this bad, in the 1970s, and in many ways the economy is far worse off now than it was even thirty years ago.

Yet for all the obvious similarities to economic crises of times past, there are also some profound differences. One of them, of course, is the existence of the Internet, and of the web of computers and other devices that provide access to the Internet. The Internet has many failings, and many dangers for the unwary, yet at the same time, what the Internet does provide is the ability to communicate, to mobilize, and to debate.

In the 1970s, the dominant communication medium was the television and radio, great for broadcasting if you happened to have the resources to own a television or radio station, lousy if you didn't. Person to person communication took place over phones, and even then only if everyone was at a known place to receive those messages. In the 1930s, even radio was in its formative stages, a letter could take weeks to arrive, and newspapers, then perhaps the dominant communication medium, for the most part reflected very local information about the world.

So long as we can keep these grids going (and we should make it a priority to keep the information grids going especially) what this provides is information about how best to survive, about how to live at a more primitive level if necessary, and about who else is doing these same things in order to emulate them or to improve upon them. The Internet makes it possible to coordinate community action, to express need or available to help with a rapidity that those even forty years ago would have been astonished by.

I watched last September as Hurricane Ike came up the coast, with visual images and communication feeds going on for as long as they could, before the storm brought the grid down, and even then people were communicating on battery power or generators across alternative networks. Within hours after the storm passed, emergency rescue was coordinating their action over ad hoc networks and when they could over satellite uplinks, aid began pouring in from across the country coordinated mostly over the web, and the first thing that people did when power was restored (or they could get access to a coffee-shop or friend's house where power was restored) was to get online and let people know they were safe and what the status was.

Even in the worst of times, this isn't going to go away. This ability to harness collective actions, to inform not only at the coarsest levels of granularity but also the finest, will prove crucial in solving the problems we're about to face. The Internet is not a panacea – resources will become harder to come by and much more expensive when they are available, giving the lie to the idea that the economy is moving towards bits not atoms – but what the Internet will do is to open up new and creative solutions to some of the oldest problems of providing enough for everyone.

In addition, to this, however, I think we're also seeing the end of the proprietary value proposition for software. When publishing companies are going out of business in droves, it's difficult pricing a database at \$100,000, or a content manager at \$20,000. Instead, what will happen is what has been happening for the last two decades – alternatives spring up, open source or mixed license tools that have most of the same functionality as their proprietary equivalents. They'll be crude at first, but the good ones will continue to get better, accrue an ever larger support community of users and extension builders and testers, until they often can compete point for point with the proprietary versions.

Similarly, data services (web services, application services, fill-in-the-blank services) represent a transition away from the proprietary productized world, supplying an abstraction layer that effectively hides the technology on the other side of the cloud with an (increasingly standardized) API. These are complementary to open source technologies, and indeed, because one no longer needs to know that service X runs on Linux or Windows or Solaris it also significantly reduces the reliance on the brand.

Open source is occurring not because there are a bunch of ideologues pushing it – I suspect that there are as many partisans on the other side of the equation that tends to balance things out there. Rather it's occurring because open source and open standards provide real reuse of legacy code and applications and even ideas – open source is proving successful precisely because there is a stronger economic advantage to being able to make use of what's already been developed than in consistently rebuilding the application because third quarter earnings necessitate such a product.

Agile methodologies are also playing a part in the transformation of software design and development, and I see signs that such practices are making their way into the broader world. Agile promotes the notion that software development is always an ongoing process, and rather than creating broad timetables and intricate Gantt charts we should just take it as a given that the software we develop will just continue through successive layers of iterative development.

Perhaps this is the same methodology that we should be employing to our economy and our lives ...

keep the goals in front of you, but make such goals discrete and obtainable, revisiting the reason for those goals in the first place as often as necessary. Make everyone involved in the decision making process collaborators in achieving those solutions, rather than just foisting the problem (and the responsibility) off on someone else by throwing money at it. Plan your activities such that at any given time, within a couple of weeks, you have something that works to a certain degree, rather than trying to find solutions that will work perfectly on or before a given deadline. Keep teams small and focused on specific goals, and make meeting those goals the responsibility of everyone on that team.

When you listen to the litany of economic "mishaps" that have led up to the current crisis, one thing begins to emerge (at least if you listen to the mainstream press)– most of the problems came about because "rogues" engaged in activities that were either illegal, unethical or both, and there was no one who was aware of the activities being done under the table. In practice of course, rogues tend to emerge in cultures that encourage them, because those same practices may make their businesses a great deal of money ... at least in the short term.

My suspicion (indeed, my expectation) is that the next few years will see a growing importance placed upon personal responsibility, on collaboration not only on production but also on oversight, and on increase in personal ethics. Hard times tend to reinforce these traits, but there is also a burgeoning change in how we view the way that we work as technical modes of thinking affect the rest of the world.

Agile methodology, open standards, data services, data abstraction and open source also all fit well into the emerging movement towards sustainability that is being seen in the broader culture. We've had nearly eighty years of growth culture, but growth cannot be sustained indefinitely. Most of the problems we face today ultimately derive from the fact that our culture (especially in the US) has been built on the foundation of consumption, initiated in the last great depression in order to get money moving in the economy again.

At the time, it was a good idea – there was a great deal of pent-up demand, and as the standards of living climbed dramatically, so too did the ability of people to contribute back into the economy. Yet we've been facing diminishing returns on our energy investments even as we've filled our houses with goods we don't use and fill our landfills with the detritus of the planned obsolescence introduced in order to sell more goods.

Economic solutions work until they don't. When they don't there is a tendency to look back to the past to see what worked before and try those solutions again – Friedman monetarism gives way to Keynesian public stimulus (the experiment underway now with the Obama administration). This will work for a little while, but in the end I suspect that even state stimulation of business will ultimately fail because the world approaching 2010 is not the world of the 1930s. New economic theories will emerge that ultimately will do the trick, but these will be based not on orthodox theories but on new ones, ones that are only just now emerging from that area that is perhaps the great testing ground for society at large ... the IT world.

*Kurt Cagle is an editor for O'Reilly media.*