Evolutionary E-Commerce
2018

RJB TECHNOLOGY
Smart Custom Software Development

Continuous Evolution is the Key to E-Commerce

While all e-commerce sites need to be able to support search, catalog browsing, and checkout, what really distinguishes one site from another is how it is able to continuously evolve towards giving customers a better, more fulfilling buying experience. Who participates in the decisions about the direction of that evolution? The customer of course! How does that evolution occur? Through the process of:

1) A/B testing to continuously solicit customer feedback through clicks and purchases

2) through continuous improvement of customer search and recommendation to improve the relevance of products that are offered to the customer

This paper discusses an actual implementation of an evolutionary e-commerce site that for our purposes we have called the Webstore.

Contents

Site Configuration

Site configuration in the Webstore is designed to facilitate creating a new site by specifying changes relative to a "default" site. Using this "inheritance" mechanism, new sites are specified with a minimum amount of additional configuration. Each site can optionally have its own unique domain name, and changes may be specified on the following objects:

- CSS
- Templates
- Page Content and Flow
- Product Selection
- Pricing

The above configuration changes are linked to specific pages, to the site itself, or to a combination of both.

Changes to the site are applied in the following order:

- Page level configuration overrides site level and default configuration
- Site level configuration overrides default configuration, and also applies where page level configuration is not specified
- Default configuration applies where neither of the above are specified

Site configuration, at both the page level and site level, allows the setup of any site with any variation from the default site, all running off the same software base. This facilitates running A/B Testing sites concurrently, and the running of Corporate Sites, and Micro Sites. Combined with the RJB Multi-tenancy framework, site configuration also facilitates customized branding in a Multi-tenant Site environment, with unique product selections, pricing, and configurable business rules.

A/B Testing

A/B testing in e-commerce is the process of simultaneously running one or more variations of the site to test a specific change. The goal of A/B testing is to identify changes to web pages that increase or maximize an outcome of interest e.g. registration, product purchase, etc. It's important in A/B testing to focus on one test at a time, i.e. don't run a registration test at the same time as a product purchase test, since one test might affect the outcome of the other. This approach ensures that the outcome reflects only the variable you are testing. The benefits of A/B testing are fully realized when the tests can be performed on almost any aspect of the system.

This approach to A/B testing facilitates testing variations on such things as:

- Page Layout
- Page Style
- Promotions
- Headlines
- Copywriting
- Images
- Page Flow
- Product Pricing

Webstore A/B Testing Framework

"A/B testing and branding is controlled through configuration of the same software base..."

The Webstore includes not only a built-in search engine, catalog browsing, automatic quote generation and checkout, but also a tightly integrated A/B test framework that facilitates continuous evolution through sophisticated experimentation. A/B testing and branding is controlled through configuration of the same software base, allowing multiple different "versions" of the site to be run simultaneously with automatic tracking of conversion statistics such as Visitor-To-Registration (VTR) and Registration-to-Paid (RTP) amongst others.

A/B testing in the Webstore can test an unlimited number of site variations simultaneously against the original site. This is particularly useful when doing price testing which we will discuss shortly, but it can also be used to test several sites against each other with other variations. If the button colors used in your original site are blue, and you want to test two variations such as green and orange, you can have three sites: the "A" site (original) with blue buttons, the "B" site with green buttons, and the "C" site with orange buttons. All three sites are run simultaneously, so as to eliminate other variables during the test. As with most A/B testing, it is typically best to test one strategy at a time so that the results clearly relate to that strategy.

It is important to note that we can govern the amount of traffic that participates in the test. We can stipulate that only 10% of new visitors participate in the test, these being distributed across all sites evenly in a round-robin manner. This distribution is "sticky", meaning that once a new visitor is allocated to a test site they remain allocated until the test completes.

How it Works

"...if you need to make some changes to only one page for your A/B test, you need only specify those changes..."

All configuration is easily performed within the Webstore itself, as it is designed to support this activity. Many of the variations listed above are fairly straightforward, but the last two, Page Flow and Pricing, deserve some more discussion. But first a word of explanation on how A/B testing is built into the Webstore framework.

The Webstore pages are made up of surrounding content like headers, footers, navigation bars, menus, and body content such as search result lists, product details, cart, etc. There are templates in use for each of the aforementioned elements, so it's quite easy to create a page by pointing to "this" header and footer template, and "that" navigation bar template, and "the other" body content template. These choices define the page from the standpoint of Page Layout and Page Style. Everything specified is an override to the default site i.e. the original. So if you need to make some changes to only one page for your A/B test, you need only specify those changes and the rest defaults to the original. Conversely, if you need to make site-wide changes for your test (e.g. change all page headers, or change all buttons), you can do this by introducing a new template, style sheet, or set of images, and by referencing that in your test site. This results in the least amount of work to set up your A/B test. Now let's discuss some specific tests.

Page Flow Examples

A number of Page Flow A/B tests have been performed in the Webstore, so we'll just mention two here today:

- Registration Thank You
- One Page Ordering

The Registration Thank You test involved testing a Thank You page after registration (default behavior of the "A" site) vs a Thank You page that solicited some optional information from the registrant i.e. what industry they worked in (the "B" site). The test was to see how many people would respond to the industry question, and whether enough would respond to make the change worthwhile. Since the Webstore catered mainly to B2B sales, it was possible that categorization of sales by industry might prove to be informative. This turned out to be the case, and enough people provided this new information to make the "B" site a winner.

The One Page Ordering test involved condensing the four page wizard style ordering process (default behavior of the "A" site) into a single page (the "B" site). Additionally, the "B" site now included logic to recalculate charges within the browser without a trip to the server when changes were made to the order. This made the ordering process much more responsive and user friendly. For example, a change of shipping address would recalculate shipping charges virtually instantaneously. In this case, the "B" site won.

"It is vitally important to have the A/B test capability in your e-commerce suite..."

Although we might intuitively think that in both of the above case the "B" site was better prior to the test, this is not always the case. There were a number of tests in which the "B" site did not perform better, and some cases where it actually performed worse.

LESSON: A/B tests are easily defined, and implemented, with the results typically available within a short period of time e.g. a typical test might run for about a week. The time period is related to how much traffic the site is getting, in particular, the area of the system being tested. It is vitally important to have the A/B test capability in your e-commerce suite, as it permits cycles of continuous improvement of the site, and continuous improvement of sales.

Pricing Examples

Pricing is an extremely important strategy in e-commerce, even more so than in brick and mortar stores. It's much easier to jump from one

"...a combination of perspectives gives us much greater insight into how pricing affects ordering on the Webstore and impacts our profitability..."

store to the next in e-commerce, and much easier to compare prices. Here we discuss one approach for A/B testing of prices. In this example we assume that we want to price our products in order to optimize profits at the store level. For a more complete discussion of pricing related topics, see Pricing Analysis.

In order to gather empirical data to support a pricing hypothesis that optimizes our store profits, we can run an A/B test. The test will show our current prices on site "A", and price variations across sites "B", "C", "D", and "E" (more sites could be added if needed). For example, we make sites "B" and "C" have prices 5% and 10% lower than our current prices, respectively, and sites "D" and "E" have prices 5% and 10% higher than our current prices, respectively. As in any A/B test, the sites run concurrently, in order to eliminate any other variables from the test. We can route 10% of our traffic to the test sites for statistics gathering.

After gathering enough data for a suitable statistical sample, we can determine several things. We can determine the "price sensitivity" of our site i.e. how much our revenue and number of orders changes in conjunction with the price changes. This can be plotted as a price sensitivity curve. Using the price sensitivity curve, and knowledge of our product costs and order processing costs, we can determine how to optimally price our products in order to achieve maximum profitability. These findings are based on actual observations rather than intuition. This approach gives a site wide perspective on price sensitivity. We can further refine our price sensitivity curve to deal with specific categories of products, since the curve might vary across categories. We can even price test individual products. Furthermore, we can relate our pricing variations to our competitor prices to account for the competition factor. Combining these perspectives gives us much greater insight into how pricing, ours and our competitors, impacts ordering and profitability in the Webstore. For more information on pricing related topics, see Pricing Analysis.

LESSON: Pricing is a key component of an e-commerce suite. Price updates must not only be fast and timely as related to market changes, but they must be accurate or sales and/or profits can be lost. Due to the large volume of data that should be considered when making pricing decisions, RJB strongly recommends using an industrial strength pricing engine to make recommendations. Ideally, that pricing engine considers a number of factors as they relate to product level pricing, including the competitive pricing landscape. Don't be caught navel gazing when pricing! Stay informed about your marketplace so that intelligent pricing decisions can be made in an agile way.

Branding and Multi-tenancy

In addition to A/B testing, site configuration may also be used for branding by running variations on the default site that use a different domain name, different page styles, different page layouts, flow, and content, a customizable product selection, as well as custom pricing. Three examples of this approach are Corporate Sites, Micro Sites, and Multi-Tenant sites.

Corporate Sites

These sites are leased on a subscription basis to 3rd party companies, and are therefore dubbed Corporate Sites. This allows companies to internally offer selected products from the default site to their employees at special pricing, with a branded look and feel for that company.

Micro Sites

Another option similar to Corporate Sites are Micro Sites. These sites are also derivatives of the default site, and are typically used for vertical subsets of the full product selection. Unlike Corporate Sites, these sites are facing the Internet and can be used by anyone to make a purchase. If the default site was a department store with many verticals, such as shoes, clothing, electronics, and furniture, the Micro Site could simply be a shoe store focusing only on that vertical. Often, the Micro Site will have different branding from the parent store.

Multi-tenant Sites

Multi-tenant Sites are the Cadillac of site branding, where tenants not only enjoy all the features of Corporate and Micro Sites, but also a unique managed product selection with associated pricing, and customizable business rules that permit configuration of the site for

different e-commerce options. Multi-tenant Sites are truly unique separate entities in every respect that can also have their own derivative Corporate and Micro Sites. Add to that a powerful cloud-based multi-tenant solution that guarantees security not only at the application level but at the database level as well, and tenants have the most affordable, secure, and flexible e-commerce offering available in the marketplace today.

The RJB multi-tenant database segmentation solution mitigates common failure points presented by less secure multi-tenant solutions. In some cloud based multi-tenant solutions a program bug in the application, or a SQL injection attack, could cross tenant boundaries and allow one logged-in customer to see the information of another customer. Since our multi-tenant solution is segmented at the database level, the database prohibits this request and either returns no data (query) or an error (update). This eliminates the threat of data leakage across tenants, including leakage based on SQL injection attacks or other application-level attacks.

LESSON: Having the capability to create a multitude of branded e-commerce sites, each with different product selections and pricing, all running off the same software base, is an almost unattainable dream of most e-commerce retailers. Not only does this simplify operations by having all sales piped through a single fulfillment solution, it lowers support cost and complexity dramatically. With the addition of the RJB Multi-tenant solution, cost of operation is lowered still further, as hundreds of stores can be supported on a single system running in the Cloud.

Search Relevance

We've all had experiences where we type in some search keywords on an e-commerce site and receive a search result that is really nowhere close to what we are looking for. This happens for a variety of reasons:

- The product selection is very thin for the entered keywords
- The site isn't indexed very well i.e. the content the search engine is indexing on doesn't contain the relevant keywords
- We didn't enter "the right" keywords i.e. the common keywords associated with the product we are looking for

Most search engines index all products on a site with respect to a relevant set of keywords derived from things like product descriptions. Relevance is typically a decimal number between 0 and 1, the higher number being more relevant. What we want to do is:

- Ensure that we have a sufficient selection of products related to a given set of keywords (assuming we want to sell those products)
- Ensure that the product selection is described adequately using the appropriate keywords so the search engine will pick them up during indexing
- Offer alternative keywords suggestions for searches that resulted in a small or low relevance result set

Alternative keyword suggestions are offered in multiple ways: through the use of Auto-Complete as the user is typing, or as hyperlinks after the original search is completed. In both cases, alternative keyword suggestions should be keywords for which there are sufficient relevant results i.e. these keyword suggestions come from the search engine itself. The suggestions also correct for misspellings which have become more common as more searches are initiated from mobile devices. Through actual testing in the Webstore, alternate

keyword suggestions lowered the number of keyword variants from 88% to 32% - this means that rather than free form typing by the customer, the Webstore suggested useful keyword searches (with known relevance) that were then selected by the customer.

> *""Order it wrong and choice is oppressive; order it right and it's liberating."* – Chris Anderson, The Long Tail

But is this enough? Research has shown that it is not. Two studies from Columbia and Stanford universities have shown that larger product selections require better organization, else the burden of choice becomes overwhelming. "Order it wrong and choice is oppressive; order it right and it's liberating." - Chris Anderson, The Long Tail. This means that in addition to returning relevant results from search, we must provide filtering mechanisms, sorting mechanisms, customer reviews, and "people like you bought..." recommendations, all to provide the organization on the search result that makes choice easier.

Improved Indexing

We've already discussed a few of the methods by which we provide more relevant search results and organize them in such a way that the choices don't become overwhelming. Research shows that when we do this people respond by choosing and making purchases. But is this enough? How do we solicit keywords that might apply to products, but which haven't thus far been included in our product descriptions, and are therefore not indexed by our search engine? How do we help customers who are looking for products related to a theme, such as "summer", or "beach"? These are not typical product attributes, but they can be associated with products through an approach that takes into account ***human relevance***.

Human Relevance

"...gathering descriptive keywords and categorizing products is related to ...human relevance measurement..."

The Webstore has its own internal search engine based on Apache SOLR, and it can also include Google search results. The internal search engine uses detailed product descriptions to index each product, and optionally uses a crowdsourcing tool for additional keywords. The crowdsourcing tool is another RJB developed feature designed to integrate with Amazon Turk in order to solicit ***human*** responses from Amazon Turk Workers regarding product related attributes. It creates Tag Clouds of attributes for each product, to improve search relevance and category management on

"...new keywords gathered through crowdsourcing must have a place in your site content..."

the Webstore. This approach to gathering descriptive keywords and categorizing products is related to techniques employed by major search engines to measure relevance of results, often termed "human relevance measurement" (see Google rating guidelines). In this case, rather measuring human relevance, we are augmenting human relevance by the addition (or subtraction) of keywords.

The crowdsourcing tool connects the Webstore product selection to the Amazon Mechanical Turk market for online work, a community of over 500,000 people who perform free word association when presented with product images. Hundreds of people are shown a single product and the crowdsourcing tool scores the responses using proprietary algorithms. For example, a picture of a pair of sunglasses might yield additional keywords such as "beach" or

"summer". The Webstore uses this information to provide links to those categories on its website. This results in more product categories, thus enhancing the Webstore taxonomy. When the Webstore sees multiple products sharing keywords, it automatically creates a category. For example, a baseball cap is found by using the search terms "hats", "outdoor sports", or "baseball". This is a form of automated search relevance and category management not found in other e-commerce offerings, and it ties in directly with how customers interpret those categories, since human beings (Turk Workers) create them. Incidentally, this method also picks up local variants of categories. For example, the keywords "touque", "beanie", "knit cap", or "stocking cap" are local variants of words that describe the same product. It's important to note in the above discussion that new keywords gathered through crowdsourcing must have a place in your site content so that they may be indexed by external search engines.

Measuring Search Relevance

Search relevance is defined as the user's perspective regarding the relevance of content, as measured against the way they navigated to that content. So how do we measure search relevance? There are three different approaches that, in order and complexity, seek to improve search relevance:

- User Engagement Metrics
- Human Relevance Metrics
- Task Relevance Metrics

User Engagement Metrics

"...an absolute must have for all e-commerce sites..."

User-engagement metrics are implemented through instrumentation of the e-commerce site, so that it logs all search terms against landing pages. In cases where the landing page is a list of products, we would also like to know the number of results shown. We then measure things like Clickthrough rates (CTR), Conversions, Abandonment rates, etc. to get an idea of "user-engagement" metrics. This is the most basic form of measurement, and is an absolute must have for all e-commerce sites.

The Webstore maintains logs of all search engine keywords and their resulting landing pages, from both internal (SOLR) and external (Google, Bing,...) search engines. User engagement metrics, such as click-throughs, are then are associated with those searches. Additionally, both searches and user engagement metrics are aggregated to provide an overall view of how the site is performing.

Human Relevance Metrics

"...pinpoint search queries that perform sub-optimally from a human relevance perspective..."

Human relevance metrics are derived from engaging a number of human beings to "rate" the relevance of search results returned from keyword search(es). Specific search keywords can be derived from the output of user-engagement metrics, or they can be injected e.g. in the case of new products. Rating guidelines must be designed in advance, relating the rating rules to the particular relevancy outcomes. The ultimate goal is to compute the Normalized Discounted Cumulative Gain (NDCG) for each search query. This will help to pinpoint search queries that perform sub-optimally from a

human relevance perspective, and indicate where remediation is necessary. For information on how eBay employs human relevance metrics see the blog post Measuring Search Relevance.

The Webstore crowdsourcing tool accepts a keyword search, which it then redirects to the Webstore search engine, returning the search results for rating as part of a Mechanical Turk Human Intelligence Task (HIT). It's important that the ratings are "blind", in order to eliminate factors such as reputation, brand name etc. during rating. The rated results are stored and, after a statistically significant number of results are obtained, the NDCG is calculated. The Webstore can be configured to use the SOLR search engine, or an external search engine (Google, Bing,...), or both, when searches are performed. The particular search engine(s) used are recorded with the results. In addition, the crowdsourcing tool can be configured to use a competitor's search, or use an external search engine to search a competitors site. Historical results are maintained, permitting tracking over time of the Webstore search performance against competitor site search performance.

Task Relevance Metrics

> *"...complex tasks such as comparative shopping require multiple queries to complete the task..."*

Historically, many improvements in search relevance have focused on "single query" searches i.e. returning the most relevant results for that query. Human beings often operate with ***intent*** on tasks that might span more than one query. Ideally, search engines should recognize that, and provide results that are relevant to the task. For example, complex tasks such as comparative shopping require multiple queries to complete the task.

There are two basic challenges to task recognition:

- Recognizing when a task begins (and ends) as human beings are capable of multi-tasking
- Recognizing the ***intent*** of the task with respect to the preferred search results i.e. recommendations based upon diversity of intent

While the discussion around establishing user intent is beyond the scope of this article, suffice to say that it is a relatively new area of research, and numerous methods abound.

One topic that continuously surfaces is the concept of ***diversity*** of results, which is dependent on the intent of the user. Given the potentially large range of users' interests (i.e. intent) the nature and form (i.e. diversity) of the search result should correspond with that interest.

Current methods of identifying and satisfying diversity are mainly resorting to machine learning techniques. The problem then becomes one of establishing suitable training datasets for appropriate recommendation engines.

"...e-commerce suites must allow you to "score" your keyword search results, by taking into account user engagement and human relevance..."

LESSON: Maintaining high quality search relevance is extremely important because it relates to how your customers view your products, and sometimes whether they see and choose relevant products on your site. Historically, both search relevance and category management are often subjective, based on precedent, and often fairly static over time. In today's e-commerce world, where search engines are used heavily to find products, that's not good enough. Products and categories must be defined in a way that

makes sense to your customers, and your relevant products need to be found with appropriate keyword searches that account for local dialects in product naming. Strong filtering, sorting, and other mechanisms are essential in helping customers make choices. Good e-commerce suites must allow you to "score" your keyword search results, by taking into account user engagement and human relevance, to ensure that you are returning a sufficient set of relevant results. In the future, relevance will include the concept of establishing user intent, and diversifying results based on that intent. In combination, these techniques help boost sales and increase repeat business.

Scalability

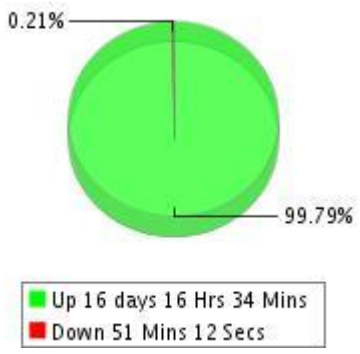
"Due to its robust system and infrastructure, the Webstore operates at close to 100% up-time"

The Webstore supports a substantial amount of traffic, with about 29 million page views annually and approximately 13,000 new visitors each day. It also supports significant upwards scalability, due to its hardware configuration.

Clustering and Failover

For critical infrastructure, there is built in redundancy and fail over. For example, the Webstore runs on multiple load balanced servers as does the Order Management System. Failure of a component causes traffic to be automatically routed to the remaining available servers. Serving these critical systems is a database server which has failover capability to the backup database server.

This is supported by a modern UPS system to provide temporary backup power in the event of a power outage. Due to its robust system and infrastructure, the Webstore operates at close to 100% up-time (see diagram right showing up-time - source www.site24x7.com):



Content Delivery Networks (CDN)

The goal of a Content Delivery Network (CDN) is to serve content to end-users with high availability and high performance. Significant benefits can be achieved through use of a CDN, both in terms of site useability, and Search Engine Optimization (SEO). Simply stated, improved end-to-end site performance results in improved user experience (UX), fewer abandonments, and improved page rank on searches. There are very few limits on the content that may be served, however, dynamic content involves some special caveats.

Dynamic content is content that ***may*** change between two subsequent requests to your site, and therefore in many cases, it must come from your site and not from the CDN. To qualify that statement, some dynamic content might only need to be served from your site once within a particular time interval, e.g. daily, because it only changes once within that time period.

Overall site performance can be affected by the "long pole in the tent", so if your site pages are often referencing dynamic content this will degrade performance and ultimately reduce UX and SEO page ranking. To that end, many e-commerce software suites use various mechanisms to limit the impact of dynamic content as much as possible, in order to take full advantage of CDN's.

Sophisticated CDN's offer options to reduce the impact of dynamic content on performance. For example, some offer a cookie based "hand-shake" between the CDN and your site, to determine whether

cached dynamic content can be served. This hand-shake involves very little data exchange and consequently very little impact on performance. Done properly, the hand-shake is only performed once for a particular piece of dynamic content. For example, when running an A/B Test the Webstore must decide which version of a particular page gets served to an end-user. When the decision is made, the information, e.g. site "B", is placed in the cookie exchanged with the CDN, and the CDN caches the page under a logical key of "site B". On subsequent requests, the CDN, on seeing the cookie will look for the cached page under the logical key of "site B" and serve the page directly. When choosing or evaluating e-commerce suites ensure that dynamic content is handled in a manner consistent with taking full advantage of CDN's.

Cloud and Service-Oriented Architecture

As discussed in this article, operating an e-commerce site involves more than just supporting search, catalog browsing, and checkout. Product selection needs to be managed, prices need to be maintained, and orders need to be taken and fulfilled. Merchandising needs access to maintain site banners, product recommendations, landing pages, and descriptive content for search engines. Where inventory is being maintained, inventory and warehousing systems help ensure that there are no stock outages. Customer relationship management (CRM) systems keep track of customer history, including customer interactions such as phone calls, order history, promotional emails, account summaries, etc.

That's a lot of potential systems! Now, your business may not have all of these requirements, but some online retailers do, and more. The question naturally arises as to where these systems will be housed and maintained. Cost is certainly a factor, as is the complexity of managing the environment. Today, many are looking to the Cloud as a panacea to reduce complexity of IT management. However,

depending on your requirements, that tradeoff can also affect cost either positively or negatively.

Without specific details about your operation, no one can tell you whether it's better to run your operations in the Cloud, from a co-location site (colo), or on your own premises. But one thing can definitively be said: it would be great if any or all of the systems mentioned above could be run in any of those locations, and moved if circumstances warranted.

RJB develops and maintains systems for all aspects of e-commerce using a Service-Oriented Architecture that permits each system to be moved to the place it needs to be. All systems, including applications and databases are platform independent. All applications highly leverage industrial strength open source software, and communicate with each other through Representational state transfer (REST) or RESTful web services. Each application is fully transportable from on-premise, to colo, or to the Cloud as needed. A RESTful architecture can be the dominant factor in user-perceived performance and network efficiency, including scalability to support large numbers of components and interactions among components (see Representational state transfer). For more information, or for advice on your e-commerce project, don't hesitate to get in touch.