The Case for Systematic User Interface Innovation using TRIZ: Case Studies

Ramkumar Subramanian, Wipro Technologies

Abstract
In this paper we propose the usage of TRIZ for systematically designing innovative user interfaces and illustrate the same with examples from practice a case study. User Interface design has been often viewed more as black box activity but with current advances in Human Computer Interaction the state of art resembles a craft rather than engineering with methods spanning art, psychology and technology. The application of TRIZ in this domain will help move User Interface Design/User Experience Design from this realm of aesthetics and functional exposure to engineering innovation.

Traditionally design aspects have focused on the trinity of form, function and fitment as the important elements of user interface. Past work has explored evidence based engineering, persona modeling, Interaction Design, Story Boards, Design Pattern libraries, Modeling Tools, Interface composition primitives and Usability Testing form the best approaches for User Experience across these elements. All these methods serve well when the domain is well defined and the user interface is only viewed as a covering over the functional core. However, a look at practice indicates that user interface is not just a covering but often a deciding factor that drives and influences functionality, quality, user satisfaction and more importantly innovation. For example, we can take the case of search engines that have added innovative functionality like auto-complete, faceted search, inline search results, pearl growing, personalization and multimedia resulting in customer stickiness and improved user experience. Similar examples can be found in social networking arena, business to consumer marketplace, smart-phones and entertainment appliances.

The above reality poses a serious question for innovators as to whether there are serious methods to foster user interface and user experience design. We posit that TRIZ can be used for systematic innovation in user interfaces. We provide a study of the application of methods of TRIZ like 40 principles to user interfaces in a variety of domains like search, retail and intranet; all using a variety of technologies and a variety of devices across all elements of user interface design. We also illustrate with the example of a case study of design of a rich internet application user interface for a subscription engine the usage of TRIZ to develop superior customer experience. We expect the usage of TRIZ along with the latest advances from HCI research and software engineering will lead to a rich phase of growth in engineering the next generation user interface.
Introduction

The design of user interfaces is a challenging and important task and often a good interface has determined whether a product is successful and brilliant over whether it is perceived as mediocre. User Interfaces are the entry point where the user perceives and interacts with the underlying software entity like an application across a wide variety of devices, computation platforms for multiple purposes. Jef Raskin, who designed the Apple Macintosh Interface, has summarized this as: “The interface is the program”.

The design of user interfaces is an evolving field and is around 50 years old. By design, we mean more a process of finding intelligent solutions to problems; sometimes non visual; rather than creating visual artifacts. Design research in this paradigm goes through the stages of prescription of an ideal design process, description on their intrinsic nature, observation of reality in design and a reflection on the fundamental concepts (Cross, 2010). Thus the conception of design in this paper in one of a process rather than an end results, i.e. a verb instead of a noun (Brooks, 2010).

Today, more attention in design is paid to the controlling “all aspects of interaction with the product: perception, learning and usage”, rather than restricting it to the functional aspects; an approach which has been termed as user experience design. As such the user interface design is a combination of art (involving subjectivity and personal preference), science (involving well defined empirical or rational laws) and craftsmanship (involving designer skill). As the field has matured, there has been more standardization in some aspects making it accessible to more people to pursue as well as defined professions like usability engineer.

The benefits of good interface design are substantial and have been well researched and documented. For example, it has been estimated that every dollar invested in system usability returns $10 to $100 (IBM, 2001). Cope and Uliano (1995) found that one graphical window redesigned to be more effective would save a company about $20,000 during its first year of use. Such economic benefits also accrue from reduced training and support costs and boost customer satisfaction. Good design also helps in refining the product concept at initial stages rather than discovering them at later stages and spending considerably more to fix any misalignment.

The roots of user interface design lie in the field of Human Computer Interaction (HCI). HCI is a multidisciplinary field with roots in computer science, cognitive psychology, linguistics, industrial design and human factors. HCI first started as an attempt to apply cognitive science principles to software development. An important model to have evolved from this perspective have been GOMS (Goals, Operators, Methods and Selection rules) by Card, Moran and Newell (1983) to for analyzing routine interactions. Further, many empirical laws or empirical models like Fitts’ law (to predict target
acquisition time), Hick-Hymans law (for choice reaction time), Keystroke-Level Model by Card (to predict task completion times), Buxton 3-state model (for predicting mouse input times) and Guiards’ model (for bimanual skill) were either developed for HCI or were modifications of existing laws for HCI purposes.

The most important factor that has been influential in HCI today is the rapid growth of technology. The key technological drivers were the emergence of the personal computer, networking, telecommuting, collaborative work and the emergence and growth of mobile computing. The graphical user interface (GUI) is now an omnipresent model for interacting with applications. Further, specialists from each of these technologies have contributed to evolve models like Window, Icon, Menu, and Pointing Device (WIMP) that are used for interactions. The ubiquitous presence of the internet has lead to the browser or browser like applications being the container of choice of user interface models. Further the domain of user interface has widened from common interactivity tasks to richer tasks like search, commerce and collaboration. This has greatly increased the challenges for HCI in terms of the sophistication required in the interfaces.

To address these challenges, there is a need to systematically innovate in user interface design. The model for innovation used for addressing this problem needs to be one of applied creativity. Creativity in design can be categorized as artistic creativity (personal expression), scientific creativity (a search for truth) and conceptual creativity (giving birth to new business relevant or useful concepts) in line with the model proposed by Razheghi (2008). Thus user interface design innovation is typically addressed more at a conceptual creativity rather than artistic or scientific level as part of the framework; although the importance of aesthetics or scientific empirical laws cannot be understated.

2 Motivation for TRIZ and Literature Review

The field of HCI that has contributed immensely to design of user interface is now suffering from a fragmentation due to its breadth as a consequence of the pervasive nature of computing technologies today (Carroll, 2003). It is difficult for a single dominant perspective or a single specialization to effectively address all the concerns of user interface design due to the richness of concepts, methods, tools and technologies available. Also, due to the enormous amount of choices available at an interface there is combinatorial complexity which prevents a human brute force or even computational approach. One of the pioneers in early design thinking proposed the use of heuristics that “suggest which paths should be tried first and which look promising” as way out of this combinatorial maze to augment human discovery (Simon, 1962).

Today a practitioner, who looks at user interface design, has at best only procedural or methodical guidance (follow these steps to achieve this objective) from theory rather than guidance that is aimed at the specific problem that is being tackled. For example there are numerous guide books, best practices, design templates and theoretical or user oriented models (Task Oriented UI design, Lewis; User
Interface Design, Wood; Effective UI, Anderson) to tackle design problems but there is lack of a tools that serve to explore the problem space with respect to its breadth.

One of the methodologies for problem solving that can be explored is the Theory of solving inventor’s problems (TRIZ) that was developed as a result of Genrich Altshuller study of empirical data with respects to patents. TRIZ aims to create algorithmic approaches for creation of new systems and evolution of existing systems and has been effective in the engineering problems in domains like manufacturing. TRIZ has also been proposed for other domains like drug discovery and aerospace. The author had proposed TRIZ has an effective methodology to understanding problems in software outsourcing (Subramanian, 2007).

Recently, TRIZ has been evaluated to demonstrate contradictions in the GUI and propose certain solutions (Mishra, 2009, 2010). This approach identifies the contradictions in the visual components of the interface and their interactions and looks at existing patents as examples for resolving contradictions. A book length treatment looks at these patents in detail for the graphical components.

However, actual user interface design is situated in the context of a business and domain problem and involves resolutions at all architectural levels impacting multiple stakeholders. For example one of the most famous patents in user interface design called Amazons one-click shopping (US patent 5960411) is much more about solving a business problem of abandoned online shopping carts by reducing repetitive keying than about optimizing keystroke entry. Thus we look at TRIZ in user interface design from its contextual rather than procedural ramifications. The creation of patents as an end process of this activity although possible and intended is not central to the scope of the paper.

3 User Interface Architecture

To apply TRIZ taking into account the complete context of the business (say retail) and the domain (say search), we need a working model of the user interface architecture. The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships between them (IEEE-ISO 1471, 2010).

The most common technical architecture for developing an application involving a user interface element is the Model View Controller architecture as depicted in Fig 1. The Model is the underlying domain, the View is the interaction model and the Controller provides the linkage between the Model and View by routing events. It would be incorrect to focus on any one of the components in isolation as the point for applying TRIZ in user interface design. For example the simple case of a new View innovation (like in case of Yahoo Widgets as depicted in Fig 2) is driven by data representation in backend (the Model) as well as newer ways of reporting and managing events by the Controller (for say automatic docking). Thus we need to look at all components of the User Interface in the given context to identify innovative opportunities. In the next section we propose how TRIZ can enable the same.
4 TRIZ for productive innovative thinking in user interface design

Although TRIZ can be applied in user interface design to search for patentable mechanisms or incremental improvements as has been done in literature, this paper looks at TRIZ from two different perspectives with detailed case studies. The first perspective of TRIZ is to identify innovative opportunities in web search engines for general information retrieval. For this we look at a wide variety of internet search engines for public domain information retrieval and identify the spaces where innovation can happen and provide examples.

The second perspective of TRIZ is a tool to solve practical problems innovatively in the context of user interface design in line with suggestions in the literature (Frameworks for Thinking, 2010). The methodology of TRIZ enables the user to look at a specific problem in terms of a particular kind of problem, identify potential solutions using tools like 40 principles and translate this into a specific solution. Rather than guarantee that a solution exists; which may be more difficult for non-material
solution spaces; TRIZ enables the designer to think in productive directions. This approach is illustrated by a detailed case study in the design of an innovative user interface for a commercial subscription engine product which is being used by one of the major telecom operators in the world.

5 Case Study I: TRIZ in web search engines for general information retrieval

A web search engine is typically used by a person on the internet to search for information of various kinds including text and multimedia. Search engines have become the primary port of entry into the world-wide-web.

Initially the search engine was a simple list of servers where content could be found. As of 2010, the world’s leading search engine in terms of number of queries served, Google Inc, has more than 63.2% market share, a 40 billion market-cap, more than 5 billion revenues, an index of more than 50 billion pages, and averages more than 3 billion searches per day. The search engines typically fund the growth through advertising, enterprise services or related offerings and have been successful at capturing a huge amount of online marketing revenue. It is estimated that search engines or search engine related revenue will have double digit growth in next five years with newer business models.

More often than not, innovations in the search space have resulted from providing a good interface to the user to bring out the latent potentiality in the data. For example, at Google, one VP, Marissa Mayer personally maintains control at code level over the front page of the search engine. Also Google was one of the first companies to memorably innovate with minimalistic user interface, separated advertisements, auto complete and image and book searches. Other innovations in search space have been driven by newer entrants like Bing with respect to inline multimedia results.

Sporadic attempts to understand the search space in terms of design patterns (Search Patterns, 2009) but till date there has been no systematic attempt to identify innovations in the search user interface. Further literature has primarily focused on enabling technologies like Asynchronous JavaScript and XML (AJAX) and related patterns for achieving specified goals rather than on the end goals themselves. We seek understand the existing search engine space and indentify the gaps that can lead to potential opportunities in the interface that are implementation idependent.

To apply the TRIZ principles, the following approach was utilized

1. Identify the various contradictions in the aspects of the search domain interface architecture
2. Map the search domain vocabulary to the vocabulary of TRIZ implicitly. For example an object could be a visual element, a segment a part of the screen-space, a means could be a search modality or interaction behavior and an ends could be a successful result.
3. For these various contradictions, formulate the question and look at the potential solutions in the TRIZ matrix. Identify the architectural layer (view, controller or model) that is primarily changed by this resolution.

4. Document known empirical solutions as a guide and provide suggestions for the domain

We now provide some key empirical data which reveal the TRIZ principles in play. After indentifying these principles, we formulate a user interface design matrix for the search domain.

5.1 Segmentation

The user interface of search engines can be cluttered if multiple information types are being searched at the same instance. At the same time, users need to search across these different types from same interface. Similarly, the search engine needs to cater to different user segments from the same interface but the search interface should be similar across all these interfaces. To handle these search engines, typically provide navigational links across the top for different types like images, video as well as pointers within search results for links to a type different from the type searched for by the user. To segment based on user needs, the search engines typically use multiple mechanisms ranging from simple advanced search buttons to complex ones like advanced searches.
5.2 Universality

A search engine may want to offer multiple features like a dictionary search, a computational search, a transformative search and so on. The key contradiction is here between the limitation on available screen estate and the requirement of providing multiple benefits in the same screen space. One option to do the same is to use the same search engine box as a universal interface for all the operations. A second option is to use the query to return all forms of possible answers across wide varieties of concerns with the probability that one of them will be useful to the user.
5.3 Mechanics Substitution

Typical web search interfaces have been based by the limitations of textual entry via keyboard entry coupled with mouse based navigation. However, users may be able to describe their search terms using other means like describing the tempo or beat using tapping on keyboard, describing a picture through a simple sketch or even supplying an image of their choice and asking for related images.
5.4 Feedback

Search engines typically have a lot of data about the probable outcomes of a user search based on their domain databases as well as the context information of the user derived from their personal or browsing information. This can be used to aid the search by providing hints, do error correction or even make pre-fetch content and show it in a phased manner.

5.5 Nested Dolls

The nested doll is useful when there are multiple facets to a given search that the user may want to explore. For example, a search for a song needs to allow search across artists, across genres or across years. Instead of preloading all the facets in the search page and cluttering up the view, these facets can be progressively exploded based on user searches. Since all facets resemble each other, this often has the appearance of a nested doll either horizontally or vertically.
5.6 Preliminary action

By preloading search results based on context and user information, it is possible to enhance the user experience of search and reduce the time taken to retrieve information. For example, the below displays the Volkswagen car interface which loads all car information and then allow user to narrow down based on search criteria. Such an interface may also aid user exposure to newer search results and thereby increased opportunities.

![Faceted Navigation maximizes use of screen estate](image1)

5.7 Anti-weight
There are certain aspects of the search interface that are heavy weight and typically cannot be exposed on the search interface due a conflict with the simpler aspects. For example, the advanced search feature is a complex option requiring multiple query boxes to refine the search based on Boolean conditions. However, using the anti-weight principle these options can be merged in-line with other options as demonstrated below.

5.8 Merging

It is likely that a user who is searching for specific information is interested in other information which is related to the information being searched. However, a search engine cannot prompt the user for all available related items and has to use some similarity score based on past information. One way to obtain the similarity score is based on the data being searched and its similarity based on content to other data. Other ways to compute similarity scores include proxies based on past browsing history of other users or leveraging social networking information.
5.9 Partial or Excessive Actions

In case of a typical search engine, for a given query a multitude of results are returned. There will be a difficulty in displaying all the results. One way to solve this problem involves displaying partial results with more links to retrieve the full content. The partial result is typically the segment of the information matching the query with the important terms visually highlighted.

5.10 Asymmetry

The search by the user typically results in multiple information sources. However, all these sources may not be similar across all dimensions like relevance, popularity or size. By introducing an element of asymmetry in the user interface such information can be communicated with ease increasing search efficiency.
5.11 Continuity of Useful Action

One of the problems with respect to search engines is that the users may expend considerable think time on a given page of search results, after which they need to navigate further down the hierarchy by retrieving the next set of results. This can cause a cognitive break and requires reloading of the entire page as well as the attendant network delays. One way to achieve the same is by providing the ability for an infinite search results page by using mechanisms like infinite scrolling as depicted below.

5.12 Equipotentiality

Often times, the search engine results need to be tailored to the existing context or environment of the user beyond the search query. For example, a user who is searching for a restaurant on his or her mobile
phone need not enter the location which can be inferred from other contextual data.

5.13 Copying

A typical information search may require retrieval of information that may be proprietary, copyrighted or heavy weight in nature. As a result of this, the information cannot be stored in search engine. One way to resolve this contradiction is to store copies of lower fidelity in search engine that make the search usable.
### 5.13 Other TRIZ principles

<table>
<thead>
<tr>
<th>TRIZ Principle</th>
<th>Example in Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheap short lived objects</td>
<td><img src="image1.png" alt="Cheap short lived objects" /></td>
</tr>
<tr>
<td>Self Service</td>
<td><img src="image2.png" alt="Self Service" /></td>
</tr>
<tr>
<td>Another Dimension</td>
<td><img src="image3.png" alt="Another Dimension" /></td>
</tr>
<tr>
<td>Discarding and Recovering</td>
<td><img src="image4.png" alt="Discarding and Recovering" /></td>
</tr>
<tr>
<td>Preliminary Anti Action</td>
<td><img src="image5.png" alt="Preliminary Anti Action" /></td>
</tr>
<tr>
<td>Intermediary</td>
<td><img src="image6.png" alt="Intermediary" /></td>
</tr>
<tr>
<td>The Other Way Around”</td>
<td><img src="image7.png" alt="The Other Way Around”" /></td>
</tr>
<tr>
<td>Composite material</td>
<td><img src="image8.png" alt="Composite material" /></td>
</tr>
</tbody>
</table>
6 User Interface Design Matrix with TRIZ principles for search

Based on the empirical data, we identify key contradictions and their solutions that can be useful for user interface designers in search space to formulate newer solutions to solve business problems.

<table>
<thead>
<tr>
<th>Aspect of Interface</th>
<th>Aspect of Interface</th>
<th>Contradiction</th>
<th>Applicable principles</th>
<th>Architecture component involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarity</td>
<td>Feature richness</td>
<td>As the number of search features increase the interface clarity reduces due to excessive clutter</td>
<td>Universality, Taking Out, Segmentation, Self Service, Thin Films</td>
<td>View Controller</td>
</tr>
<tr>
<td>Search-friendliness</td>
<td>Discovery-friendliness</td>
<td>As the interfaces facilitate focused retrieval of information (search), the ability to discover related resources by design or</td>
<td>Segmentation, Mechanics, Substitution</td>
<td>Model View</td>
</tr>
<tr>
<td>Breadth</td>
<td>Depth</td>
<td>As interfaces facilitate breadth of information retrieval across various sources, the depth of information for each content is reduced</td>
<td>Asymmetry</td>
<td>Nested Doll</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Specialization</td>
<td>Generalization</td>
<td>As the search engine interface becomes more specialized for a given content type or user, it cannot handle generic information.</td>
<td>Local Quality</td>
<td>Other way Around</td>
</tr>
<tr>
<td>Interface Richness</td>
<td>Load Speed</td>
<td>As the interface becomes rich in terms of the components, the loading speed of interface reduces</td>
<td>Preliminary Action</td>
<td>Anti Weight</td>
</tr>
<tr>
<td>Structure</td>
<td>Flexibility</td>
<td>As search is more structured (say number of categories), the flexibility of interface to accommodate newer content goes down</td>
<td>Dynamics</td>
<td>Skipping</td>
</tr>
<tr>
<td>Number of Free Results</td>
<td>Paid Results or Ads</td>
<td>As the number of paid results increases to</td>
<td>Segmentation</td>
<td>View</td>
</tr>
</tbody>
</table>

Copyright ©2010 by the original authors
7 Case Study II: TRIZ in design of Subscription Engine Interface for major telecom operator

A recent application of TRIZ for user interface design was undertaken when developing a solution by a leading software firm for value added subscription deliveries to end users by a major telecom operator. The telecom operator wanted an interface that was friendly to both operating personnel and business users, easy to modify and customize and highly extensible.

The user interface also served as the point of discovery for new functionality between the architects and the customers and needed to be highly flexible. Also, there was an additional requirement that the interface solution should not be purely problem specific but sufficiently general to allow customization for other domains and problem sets.
The brief system constraints and components were identified for producing an innovative design to satisfy the requirements.

<table>
<thead>
<tr>
<th>System Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Run Time Java Virtual Machine</td>
<td>JDK 1.6 or higher</td>
</tr>
<tr>
<td>Independent services composing the system like Publish Handler, Subscription Handler, Purchase Handler and User Manager</td>
<td>Loosely coupled Java Objects</td>
</tr>
</tbody>
</table>
Management services like Logging, Simple Network Management and Queue Manager | Wrapper services over industry strength utilities
---|---
Audit services like Reporting | Data collection
Application Server | JBOSS or Web Sphere
Web Browser | Fire Fox and Internet Explorer

| Hardware Resources | Multi-Core machines |

<table>
<thead>
<tr>
<th>Type of Constraint</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Temporal</td>
</tr>
<tr>
<td>Resource constraints</td>
<td>Physical like memory, CPU time Virtual like queue connections, user session tokens Quasi Physical like database connections</td>
</tr>
<tr>
<td>Client constraints</td>
<td>Screen Space Browser Memory Biological end user constraints (information processing)</td>
</tr>
<tr>
<td>Other constraints</td>
<td>Licensing constraints Security Constraints</td>
</tr>
</tbody>
</table>

To apply, the TRIZ principles, in line with the system, its constraints and components, the following contradictions were identified and solutions were formulated using a similar process as in the case of search engines

<table>
<thead>
<tr>
<th>Aspect of Interface</th>
<th>Aspect of Interface</th>
<th>Contradiction</th>
<th>Major Decisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator Friendliness</td>
<td>Business User-Friendliness</td>
<td>As the interface became more usable to operating personnel (speed), it becomes less usable to business personnel (usability)</td>
<td>The principle of segmentation was extensively used and the interface looked like traditional desktop application like Outlook. Also the principle of taking out was used to separate complex operations into separate interfaces.</td>
</tr>
<tr>
<td>Specialization</td>
<td>Generalization</td>
<td>As the interface became more specialized to handle subscription problems, it had less generality for other problems</td>
<td>The interface used universality principles to ensure that a component developed could be used for other with minor modification to the text.</td>
</tr>
<tr>
<td>Correctness</td>
<td>Simplicity</td>
<td>As the interface added features added more options to ensure correctness, the simplicity of the interface reduced.</td>
<td>The interface used the principles of dynamics to ensure that only parts which are useful to the user appeared as needed.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Richness</td>
<td>Load Speed</td>
<td>As the interface used richer components, the load speed decreased.</td>
<td>The interface used partial actions so that things were loaded only as required and not at the start.</td>
</tr>
<tr>
<td>Extensibility</td>
<td>Consistency</td>
<td>As the interface became extensible to accommodate newer components, the consistency across these components decreased.</td>
<td>The interface used an intermediary managerial component at interface to ensure consistency. Using a self service mechanism any new object could appear under the manager.</td>
</tr>
</tbody>
</table>
The development of this interface enabled the software firm to win additional business to port other interfaces to the framework, significantly improved client satisfaction and also allowed rapid prototyping of related interfaces when responding to queries for developing similar components for partner management and customer relationship management. Further features like analytics were easily integrated into this new framework.

8 Summary and Future work

We motivated the rationale for good user interface design and documented the key challenges in this domain. We find that the combinatorial complexity of design as a process requires the use of algorithmic procedures like TRIZ for innovation over and above guidebooks and theory. We also find the user interface design should be looked at in the context of the domain and underlying architecture.
rather than as a separate artifact for TRIZ to be effective. The case study with search engine demonstrates a procedure where empirically valid TRIZ principles can be identified and cataloged for future designers. A similar case study with an interface for commercial product demonstrated how TRIZ was used to design innovative solutions that satisfied multiple stakeholders in a commercial engagement.

For future work the exploration of new domains like games or virtual reality, newer problem spaces like 3D interfaces or enterprise computing and specific design challenges is possible. Further on formalizing the solutions discovered in form of the TRIZ matrix by mapping the concepts is useful for practitioners. These efforts should be aimed at integrating TRIZ into the mainstream of user interface design by more scientific studies under laboratory conditions that can demonstrate its validity.

9 References
15. IEEE Std 1471–2000, Recommended Practice for Architectural Description of Software-intensive Systems
10 Appendix A: List of public search engines

1. www.google.com
2. www.yahoo.com
3. www.amazon.com
4. www.landsend.com
5. www.evri.com
6. www.ambiently.com
7. www.delicious.com
8. www.flickr.com
9. www.loki.com
10. www.bing.com
11. www.blackle.com
12. labs.systemone.at/retrievr/
13. www.volkswagen.co.uk
14. www.dogpile.com
15. www.metacrawler.com
16. www.corbis.com
17. www.snap.com
18. www.timesearch.info/
19. www.etsy.com
20. www.zakta.com
22. search.twitter.com
23. www.viewzi.com
24. www.exalead.com
26. www.chemspider.com
27. www.jinni.com
About the Author

Ramkumar Subramanian is the Senior Architect in Global Communications Service Provider at Wipro Technologies. Subramanian has ten years of industry experience in telecommunications in areas of service delivery platforms, switching, computer-aided telephony, application servers, interactive voice response systems and Voice over IP with leading equipment vendors and service providers. His current research areas are in innovation, service orchestration and composition, software architecture and interface design at all levels.

Ram holds a degree in engineering from NIT and has pursued executive management qualifications at IIM Bangalore. Ram’s previous publications in national and international journals and conferences have focused on service orchestration, Web 2.0 business strategies, wireless sensor network interfaces, and TRIZ in outsourcing.

Contact Ramkumar Subramanian at ramkumar.subramanian (at) wipro.com.