



Rapid Application Development for Creating Knowledge from Data

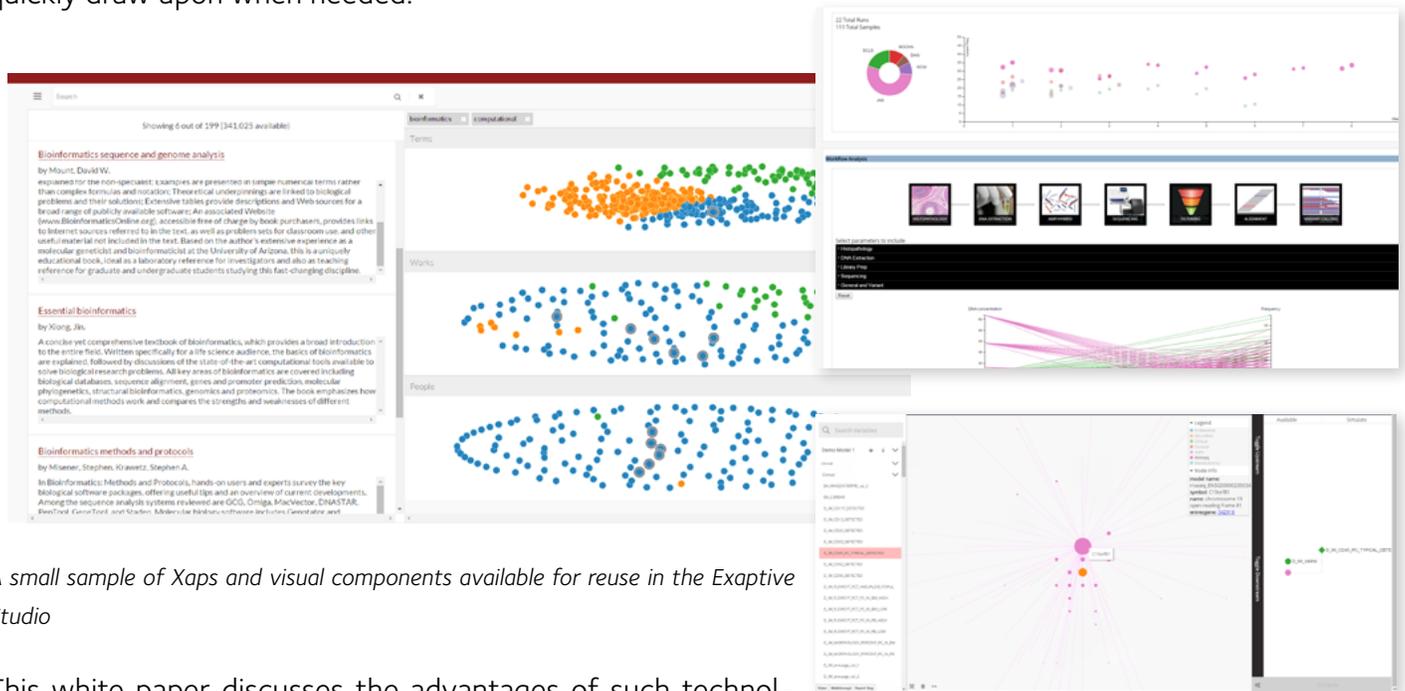




Some data-driven pursuits will always require human judgment. In these cases – where exploration is necessary for creating knowledge from data – researchers, analysts, and business managers are familiar with the frustrations of limiting prepackaged solutions and resource-intensive custom software development. On the other side of things, IT departments experience the pain of ever changing specifications, limited code reuse, and dynamic data. What all involved would prefer is software iteration that is fast and flexible enough to match the users' exploration of data.

At Exaptive, we believe that generating insight from data requires more than building software that is better and better at specific tasks. Specialization works until an unanticipated need arises, whether it's an unexpected question or a need for additional data. At that point, specialization becomes a liability, existing technology becomes a hindrance, and adaption is slow. Most have seen this cycle repeat itself with enterprise software.

The solution is rapid application development that is iterative and flexible. To achieve that, we believe disparate data and analysis tools must be modular, interoperable, and community driven. Modular tools and interoperable data enable rapid iteration. The presence of community expands the array of data, tools, and techniques to quickly draw upon when needed.



A small sample of Xaps and visual components available for reuse in the Exaptive Studio

This white paper discusses the advantages of such technology-agnostic and source-agnostic rapid application development and Exaptive's approach to achieving it.

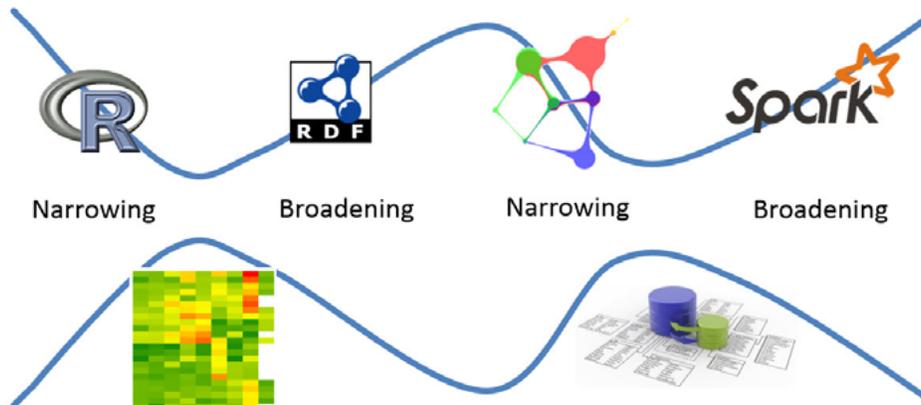
Anybody who has tried to generate insight from data is likely familiar with the unpredictability of the process. You begin with a particular corpus of data, analyze it, focus on something interesting, and then, in most cases, realize you need to analyze it differently, need additional data, or need to change perspectives. If you find an answer it is usually accompanied by more questions. The road to knowledge is rarely straight.

At Exaptive, we often refer to this process as the Cognitive Hourglass¹. The process requires a yet unknown array of tools to support a human decision-maker's unpredictable cognitive process, often involving a series of narrowing and broadening tasks.

¹We have written in greater detail about this concept specifically in life sciences and healthcare in another whitepaper. You can download it here: <http://www.exaptive.com/exaptives-cognitive-hourglass-whitepaper>.



What is all too familiar, for both the data-driven decision maker and IT workers, is how frustrating it can be to string together technologies into applications that address evolving needs. Specifications change quickly. The turnaround feels too slow. Speed doesn't just push back the timeline; it can mean other projects never get off the ground or that a project gets defunded.



An illustration of the data exploration process that Exaptive calls the Cognitive Hourglass

What if rapid and flexible iteration could facilitate the cognitive process and make the feedback loop between users and IT less painful? It can, and the benefits are:

- faster pursuit of insight with rapid iteration;
- more efficient software development with greater modularity; and
- more data-driven innovation with a lower barrier to experimentation.

Rapid, flexible application development enables faster pursuit of insight because end-users, the subject matter experts and decision makers, get software that addresses their new questions and enables the next step in their exploration of the data landscape. Iteration also pushes their cognitive process forward. Exploration uncovers the next questions. Sometimes, you don't know what you need until you see something that isn't what you need.

Modularity of code and high code reuse are, to a certain extent, the gold standard of software development and rightly so. Normally, piecing custom software together with different technologies requires glue code and customization from project to project. Modularity enables more efficient code reuse and faster, more flexible adaptation to changing specifications.

Speed and flexibility also lower the barriers to experimentation and outside-the-box problem solving. If software developers and end-users can experiment easily with new data and analysis techniques – whether they are algorithms or visualizations – without getting bogged down in glue code², hard problems get solved faster and innovative approaches get implemented instead of discarded because they are too experimental to justify.

²For an explanation of glue code, see https://en.wikipedia.org/wiki/Adapter_pattern#Glue_code.



Where your data-oriented pursuit requires exploration and involves unknown unknowns, driving toward better and better specialized software, whether pre-packaged or custom, will work until you reach an unanticipated curve in the road. Rapid software prototyping and iteration, as a part of a feedback loop with the subject matter experts who are the end users, is actually the straightest line to insight, discovery, and knowledge.

Code Modularity and Reusability

The Exaptive Studio lets developers and data scientists modularly couple back-end, data-access, and algorithmic tools with front-end technologies, without having to retrofit old code or mess with glue code. Customary development tools (e.g. R, Python, Hadoop, Spark, SPARQL, SQL, Angular.js, and d3.js) work within the Exaptive environment. Users work with what they know. The Studio makes the technologies modular. Users build web-based data applications – what we call “Xaps” – that are easy to deploy to end-users and iterate.

Every code module in Exaptive is, furthermore, agnostic to the next, meaning they can be reused to adapt to new specifications or a new project with customization. Our component library provides immediate utility and unexpected inspiration. Programmers can use their skills to code the fraction of the application that is unique to their need while leveraging existing components they have already written, components provided by Exaptive, or components created by Third Parties³.

Data Interoperability

We also apply the principle of modularity to achieve greater data interoperability. Exaptive’s unique data model and service-oriented architecture allow Xaps to access multiple data sets and run large processes without requiring additional data warehousing. Users can explore a seamless data landscape to generate insight faster, using fewer resources. Developers can use or code new components to access and interpret between data sets instead of having to move and organize it all before analysis can happen.

Community-Driven Productivity and Experimentation

This level of modularity and interoperability also enables users to springboard off of and be inspired by others’ work. Developers and data scientists have specialties. They can rely on modules outside of their expertise and they can give their expertise greater reach in an environment where the modules are agnostic to each other and can be easily repurposed. The result is a development environment where productive play is a reality.

Often innovation happens by chance, when someone is serendipitously inspired or encounters an unexpected collaborator. Exaptive’s mission is to facilitate this serendipity. So we enable our users to expose their ideas and draw on others’ ideas, whether it’s the user community as a whole or within institutional boundaries⁴. Programmers find inspiration or get unstuck from suggestions or collaborations enabled through cognitive community-driven networks.

³For more information you can read about the Exaptive Xap Store at <http://www.exaptive.com/combinatorial-marketplace>.

⁴For more information, check out <http://www.exaptive.com/cognitive-network>.

