



INTERACTIVE INTELLIGENCE

Interactive Intelligence Embeds Rich Reporting into Their .NET Platform by Using the Flexible Architecture and Programmability of ActiveReports

Fast Facts

Customer: Interactive Intelligence

Web site: <http://www.inin.com>

Industry: IP Communication

Customer Profile

Interactive Intelligence is a global provider of unified IP business communications solutions, providing innovative contact center software, IP telephony solutions and business process automation across a variety of vertical industries; as well as document management solutions for the insurance industry.

Solution Profile

Interactive Intelligence takes a unified approach to business communication and process automation and has an application architecture that is configurable and easy to extend over time. ActiveReports, with its flexible architecture and programmability helped to seamlessly embed reporting into their applications.

GrapeCity Software

- ActiveReports



“The key feature for us was the completeness and simplicity of the API.”

Thomas Kashin, Team Lead, Client Group, Database and Reporting Team

Business Situation

We need to deliver reports to customers in a wide variety of vertical markets using Oracle or SQL Server. As a software company, we had to be sure that whatever product we chose had enough flexibility to be fully integrated into our larger product and framework.

Business Challenges

I was tasked with finding a reporting tool to meet our needs, or if that proved impossible, deciding whether to write one ourselves. Our existing reporting solution was not fulfilling the needs of our customers, sales partners or customizers.

Development Challenges

In order to integrate well with our data environment, we require packages that allow complete customization of data sources. Ones that only allow a simple connection string and a database driver would never meet our security and flexibility requirements. Furthermore, to source a report, we require complete control of the SQL used to retrieve the data to allow for in-house performance tuning with respect to server vs. client load, as well as user-driven filter customization. A complete, sensible API was critical to a successful delivery. In order to give

our customers maximum flexibility, we had to have the ability to manipulate every aspect of the report and graphical elements during report processing. We required this API to integrate seamlessly into the .NET framework. We wanted our metadata—managed in our environment—to control the report parameters, data, delivery method, etc. The ActiveReports API allowed us to create as much metadata infrastructure as required by our product to customize our users' reporting experience.

Why GrapeCity Products?

We reviewed the products available from:

- SAP (Crystal Reports)
- Perpetuum
- Component 1
- Cognos (although this product was ruled out about six months prior to my employment)

In my research to find a reporting product, the first thing I always did--for each product--was to see how long it took me to have a finished report using .NET indirect data retrieval methods. This meant that I didn't just rely on the product's connection string and database driver; I used a connection method similar to the one I planned to use in our completed product.

In one case, I was not even able to easily use their API to configure the data source, much less to manipulate data at run-time. Not only were their help and samples inadequate, it took their support personnel two days to return my call. Ultimately the solution they provided was nothing like their samples, and was so convoluted and non-intuitive as to render it unusable. All this was just to bind a report to a custom data source. (Just for some background: I had about 10 years professional programming experience when I did this evaluation, and have been using .NET since 2003. I've been using databases since 1991 and the .NET DataSet structure since 2003.)

Each of the other products I tried, though better than my dismal experience mentioned above, did not function as well or as easily as ActiveReports for .NET 3.0.

I had one or two questions during my evaluation of ActiveReports, and in each case I was able to get through to a Data Dynamics support individual who had the answers either right off the top of his head, or was able to email me simple, workable solutions within a few hours. I tried the support hotline for each of the products I evaluated: Data Dynamics came out on top. The fact that I had not yet even purchased a license for the software was not an issue.

For my evaluation of the quality of the selected vendors, I perused the Data Dynamics support forum. I followed specific question threads to their resolution. From a non-scientific perspective, the trend appeared that the Data Dynamics support organization was relaying bugs and product deficiencies to developers or level-3 support personnel, following up with the reporter regarding the status of the bug, and completing the thread with a post of where the fix could be found. Having been a developer in a

few different software organizations, I found (1) the time-frame for bug correction to be reasonable, and (2) the urgency attached to bugs to be appropriate for the given level of exposure and severity.

Like almost everyone else I spoke to at the time, I had never heard of Data Dynamics, whereas just about everyone had heard of Crystal Reports and Cognos. Thus I can't say that notoriety was a factor in my decision.

One other aspect we did consider, however, was the financial stability of the vendor. This was a bit of a concern with Data Dynamics as it was smaller than other competitors. I spoke to one of their higher-level managers at the time who was able to describe the company's capital structure and ability to remain a going concern in enough detail to quell my concerns.

Solution

With complete control of all the report and graphical objects, we had the freedom we needed to create our own metadata and supporting infrastructure to deliver reports to our customers. (The licensing cost and terms were also favorable--this was a critical, though non-technical, factor as well.)

The key feature for us was the completeness and simplicity of the API. We had the ability to create a base report and inherit our 'out-of-the-box' reports from it as well as allow customers to deploy their own reports using our metadata infrastructure and base report. This complete API allows our metadata to control many aspects of the reports--without over-complicating deployment and execution. It also enables us to integrate the Data Dynamics report presentation layer with our back-end, while delivering the report in a distinctively Interactive Intelligence UI.

The amount of effort required to build the system has been significant. Most of this time is spent on (1) designing and coding the metadata management system, and (2) building each report using the ActiveReports editor (which is on par with competitors, but in some cases, could be improved).

We've also been able to define drill-through paths in the reports using the ActiveReports hyperlink feature. Again, with full control of the hyperlink value and the code in the resulting event, we decide how the next report is rendered, which parameters are selected, etc.

Our customer sizes range from small (reporting on tens of thousands of rows) to large (reporting on tens of millions of rows). We are attempting to tune our metadata layer and the code we have in each event during report generation now to avoid problems later. Reporting on millions of rows is always a performance issue. Though we have not yet deployed the product in a customer site large enough to receive complaints, I am convinced we eventually will.

Benefits

We had an old (MFC) integration to Crystal Reports in previous versions of our product. Though I'm told that we got it to work after significant debugging and troubleshooting, our customers and customizers find it overly complex to understand and to glean usable business intelligence from the old system. Additionally, the existing reports appeared to be "bolted on," or written as the need arose rather than designed to fit strategically into the operational intelligence requirements of our particular market. Though this last point has nothing to do with the specific reporting tool, we feel that ActiveReports enables us to deliver

more tailored, informative, integrated, distinctly customizable, and better-looking reports. For example, I designed the architecture in such a way that a developer on any module can call the reporting module with a specific set of objects configured to run a report with specific parameters. In this way we can include a "Print..." or "Report on this data..." button on almost any screen. It gives the consumer full control of which parameters to pass to the reporting module, and lets the reporting module (specifically the ActiveReports component) to either print, export, or simply display the report.

We are also taking this opportunity to re-work the way we analyze and present data. We are utilizing the benefits of the graphics package built into ActiveReports, as well as the more secure data query method we developed and have been using for several years now in other areas of our product.

I would not include "cost saving" as part of what we're getting out of this project; it's more about delivering better reports, simplifying the process to add custom reports, and investing time and money in improving this whole area of our product. The hope is that it will generate revenue by:

- positioning the product in a better light with respect to reporting
- streamlining our customers' experiences with running and getting useful intelligence from reports
- allowing customizers to create their own content and deploy those reports in our product
- enabling customers' operations to cut costs by having better intelligence

“ActiveReports enables us to deliver more tailored, informative, integrated, distinctly customizable, and better-looking reports.”

Thomas Kashin, Team Lead, Client Group,
Database and Reporting Team,
Interactive Intelligence

Looking Ahead

We're looking forward to delivering a partial release of our ActiveReports solution by the end of the year.

We'd like to eventually deliver our complete solution using ActiveReports, and a complete ad-hoc BI system using GrapeCity products. We haven't had the time or resources to research or develop the full ad-hoc version yet. We'd like to use the metadata infrastructure which we've spent time developing to drive that product as well. This would include a graphical analysis tool using drag-and-drop technology to bring dimensions in and out of the model, dashboarding and console technology, etc. If something existed that has the same complete API as ActiveReports, we'd seriously consider evaluating and perhaps licensing it.