Computational Integration propels institutions forward in today's knowledge economy

Enterprise leaders who wish to compete in today's fast-paced knowledge economy must engage the productive capacity of their workforce and fully exploit the innovative potential of their record systems. eLoomix, LLC, has developed and licensed a technology that does just that. Called Computational Integration (CI), this technology has proven to increase employee productivity, increase record system utility, increase data quality and consistency, and decrease the cost of maintaining and utilizing electronic records.

Well-designed record systems support the capture and management of data such that records are complete, accurate, persistent, and interoperable with systems that make use of that data. For example, academic medical centers and other biomedical research institutions across the United States have invested vast amounts of money and labor to build and maintain large repositories of medical records and basic and clinical research data. This is a significant accomplishment and, building upon these and similar large and established data resources, we can take the operational efficiency of these entities to the next level by capturing the processes that individuals have created to render their institutional data useful to their work.

The true cost of records systems

We tend to count system costs in terms of the amount it costs to either purchase a system or to develop and maintain a system. However, there are other costs, including the cost of unproductive use of the system, the cost of working around the system, and the opportunity cost to employees and institutions of busy, but unproductive, labor hours. A more holistic way of looking at the total cost of a system is to include the unproductive system use, workarounds, and lost opportunities as development costs that are borne by other budgets.

Cost of unproductive system use

Manual data entry provides a classic example of the high cost of unproductive system use. When engineers are designing and building large, monolithic record systems, they do look at possible inputs to the system, but the number and variety of potential data input sources makes it too expensive to build a special import program for each one, so engineers build general data-capture devices, such as web data entry forms for human use and APIs for programmatic use. In terms of development costs, data entry forms are an inexpensive way to solve the data-capture problem. However, when employees must dedicate more than half of their day to the task of manual data entry, manually porting data from one system into another, the "cheap" data entry form suddenly looks a lot more expensive. When you take into account the inevitable mistakes, the low job satisfaction, and the high turnover associated with such tasks, the "cheap" data entry form costs even more.

This is where CI comes in. CI captures and automates the processes people manually execute vis-a-vis their institutional data. If people are manually entering data from a disconnected system into the main record system, we build a programmatic bridge that automatically loads the data. In addition, these bridges can perform algorithmic checks on the data to support regulatory compliance and quality control efforts. Likewise, if people are required to record the events of the day, we build programs that pre-load as much existing data as possible, and the employee enters only the data that is truly new. These data bridges and loaders significantly reduce the number of hours engaged with the record system, and they increase the quality and consistency of the data capture by preventing human error. CI uses the record system APIs to bypass manual data entry forms, thus making data capture less expensive.

Cost of system workarounds

Another hidden cost of record systems is workarounds, which often involve people manually constructing views of data that are either helpful to their decision-making, helpful to planning and executing their day's work, or helpful to communicate something meaningful to their colleagues. It isn't reasonable to expect monolithic system engineers to anticipate all the possible uses of captured data. The most common approach to solving the future data use problem is to build a web interface for querying the data and to plug in a standard enterprise reporting application, such as Crystal Reports. The problems are many. People may not have the necessary expertise to construct queries that join multiple tables. The queries take a long time. People may not have access to the standard reporting application and may not have the expertise to use it. The reports may not be customizable to the extent needed and, finally, if crucial,

meaningful data resides outside the system, there's no way to include it in the reports. As a result, people resort to a mixture of queries and spreadsheets and even paper re-constructions to obtain what they need for their work.

Once again, CI captures and automates the manual processes that people are executing vis-a-vis their institutional data. This powerful technology has no difficulty correctly assembling data sets that span multiple sources. The source data does not have to be contained in the record system to be available to CI. CI can pull in any data, from any available electronic source, and correctly stitch it together provided there's something to match on. In addition, this powerful technology has no difficulty with finely tuned report formats, where the layout of the page itself is meaningful to the report's user.

Opportunity cost of busy, but unproductive labor

A third hidden systems cost is the institutional opportunity cost of employee time spent on manual data capture and system workarounds. Manual data entry and system work-arounds cost the employee mental and emotional energy, and this energy is used to do essentially the same thing, solve the same problem, over and over, day after day. It's costly to the employee, but it's even more costly to the business, who cannot benefit from this mental and emotional energy in the form of creative solutions for new problems. An enterprise in today's rapidly evolving knowledge economy cannot afford this. If an enterprise is to be a leader in today's world, it must have the creative power of its employees at full capacity, and CI unleashes this power.

CI unlocks another frontier in enterprise growth and maturity: data-driven operations, where data that have been collecting over the past few decades are used to scientifically answer operational questions about best practices. Questions like: "What is our error rate?" or "Does this practice actually produce the expected result?" CI makes it relatively easy to build complex data sets and perform statistical analyses on the assembled data. CI thus increases the utility of the captured data in addition to reducing the cost of its capture and use.

How Computational Integration succeeds

CI succeeds because development happens quickly and organically and in direct consultation with the individuals who are performing these data operations. CI does not attempt to build enterprise systems. Instead, it directly integrates employees' institutional knowledge as software. This enables the employees' organizational knowledge to remain useful over time, subjects it to testing and improvement, and makes it transmissible to others in the same domain.

To accomplish this, CI uses a microservices architecture, which means it builds small applications, called microservices, that solve discrete problems. Due to this architecture, it can evolve applications as needed without affecting any of the others. The developer and the user work hand-in-hand to iteratively develop applications that fit the users' needs as precisely as if the users were to do the task themselves. This automation also gives the users a chance to make improvements to their processes that would be otherwise be out of reach.

The development is fast because CI uses a functional programming language and because it builds these microservices using a platform of functions that already capture the users' domain. For example, there are functions to establish connections with data sources, functions to manipulate tables, functions to retrieve, check and verify the data, etc. This platform of functions is analogous to a dictionary of words. Microservice applications are analogous to essays written using those words. You can create an infinite variety of essays with a relatively fixed and small dictionary. For the specific domain of biomedicine, a platform of basic functions is already complete.

Conclusion

For large, complex entities to compete in today's knowledge economy, they must have technology solutions that propel the entity toward its goals. If employees are largely engaged in unproductive manual workflows, then technology is holding the enterprise back. If employees are using the technology to innovate and solve the problems deemed most urgent by management, then the technology is doing its job. CI, with its organic development approach, microservices architecture, and functional language platform is a forward-propelling technology. It is fast enough and flexible enough to keep up with the multiple competing demands of the innovative and visionary people, who are the beating heart of any operation.