EDITORIAL

In IT, the vocabulary for Enterprise Reporting is one of the richest with many words (« Infocentre », Datawarehouse, Datamarts, Business Intelligence, Executive Dashboards) to designate the sophisticated reporting and decision-support tools. We trace back how reporting technology has evolved and explore the growing trend as vendors continue to consolidate their offerings to address Datawarehousing and Business Intelligence.

Clients today look for a rich client experience and expect similar interactions within a browser as with standard desktop applications. AJAX is being rapidly adopted to address these needs. We shade light on what is behind this buzzword as well as how Cambridge follows this track. Stay tuned for future articles where we will delve deeper into this area. Another emerging standard targeted to simplify enterprise web application development using the J2EE stack is the Seam stack from JBOSS. In our endeavour to determine the best frameworks and technologies to facilitate high quality development with lower effort, we share our point of view on the benefits offered by Seam.

Successful Projects need a strong collaboration of business process understanding, IT infrastructure and architecture and project management. Over the years, Cambridge Consultants have successfully bridged this gap between the Business and IT teams for many of our key clients across Switzerland. We present our fruitful experience with Helsana where we provided extensive support during the execution of their enterprise wide CRM project.

Cambridge has served its clients across Switzerland over the past decade. We have delivered many projects ranging across different vertical industries and spanning different technologies. Our consultants continue to make a positive impact on clients repeatedly providing support in a broad spectrum of engagements. We remain committed and would like to take the opportunity to thank our client representatives and their teams for their trust and continuous co-operation. We look forward to many more successful ventures in the future!
Over the years, many methodologies have been developed with the goal of providing executive teams with the relevant data to support their decision.

Why then, after many years spent developing and enhancing Datawarehouse/Business Intelligence (DW/BI) platforms, is it still often so difficult to produce, change or distribute a report?

Despite the never-ending sizeable investments and the managerial attention this topic attracts, current DW/BI platforms may not be delivering on their promises. How can this be explained?

In order to understand the limitations, two aspects need to be analyzed: how these DW/BI platforms have been constructed over the years and, which data management techniques they use to collect, clean and transform the transactional data into meaningful information. Data management techniques have largely evolved over the last twenty years. They started with the flat-file data storage technique. Although quite simple, this technique has proved to be flawed in the fact it created significant data redundancies among the files.

In response to this data redundancy issue and the associated increase of data storage requirements, companies rapidly adopted a more efficient technique to store and administer data: the relational databases. While addressing the shortcomings of flat-files, the new approach required a significant IT knowledge for users to create reports. In addition, the elimination of data redundancy had a major impact on performance which can lead to potential lock-ups of the entire DW/BI platform, preventing the rest of the organization from accessing it. OLAP engines, the third generation of data management, were therefore designed to solve these two issues. They provide slicing and dicing, filtering and sorting functionality that enable end users to manipulate easily large blocks of data, called cubes. They provide a very fast access to pre-computed data, hence enabling decision-makers to create on-the-fly analysis without asking for IT developments.

OLAP engines are such a radical departure from traditional databases that IT skills required to design, optimize and administer them are rare on the market. Furthermore, when it comes to adding, modifying or removing data, the OLAP engine is far less flexible than a relational database.

These three techniques coexist today in most companies and constitute the standard DW/BI three-layer architecture. The first layer aims at collecting data from transactional systems. The second layer is the relational database that stores the data and serves as the core DW/BI data repository. The third one is the OLAP engine which is populated with the data extracted and transformed from the second layer and accessed by end users for instant, on-the-fly data analysis requirements.

But today, even this standard DW/BI architecture is proving to be too limiting to effectively meet enterprise reporting requirements for three main reasons. First, the high costs required to design, implement and operate such platforms both in terms of human resources skilled in various data management techniques as well as hardware and software resources spanning the various technologies.

Second – and probably the weakest point of this architecture – is the complexity required by changes to OLAP cube structures. It usually requires the full recreation of the cube which can last for hours. Quite a constraint in today’s ever-changing environment!

Thirdly, we question the need itself for powerful OLAP tools. In our experience with DW/BI, most of the end user reporting requirements can be met using simple Web-based reporting engines. Only in rare cases (e.g. client segmentation analysis, or complex performance analysis), will an OLAP engine do better than a simple reporting engine.

In light of the above, is there a new architecture that could reduce the overall technical complexity of the “best practice” DW/BI platform and adapt quicker to ever-changing reporting requests?
Based on our latest experience in the DW/BI area, the answer is yes thanks to two emerging trends: integrated product suites and in-memory OLAP engines.

Major software vendors now cover the whole spectrum of the DW/BI area and propose product suites that cover the entire reporting chain from the transactional data extraction process up to the creation of the OLAP cubes.

On the OLAP technology front, a new technique is now emerging which provides the flexibility of a typical relational database engine and the analysis power of the OLAP. Contrary to regular OLAP engines which create physical data cubes accessed once in a while and stored independently from the relational database, these OLAP engines create in-memory data cubes on-the-fly on demand when the user really asks for OLAP functionality (e.g. slicing, dicing).

From a conceptual standpoint, this new technique can easily be implemented on a typical DW/BI platform as it fits within the same three-layer concept presented earlier.

This novel approach simplifies the IT administration chores by eliminating the complex daily updates and distribution of the physical OLAP cubes throughout the whole company. Moreover, this concept of “virtual” OLAP brings the long-awaited flexibility that was only dreamed of in the current standard architecture: while an OLAP engine practically limits the number of analysis dimensions to 5 or 6 per OLAP cube, this new platform can build “virtual” cubes of more than 20 dimensions, more than will ever be needed.

This new architecture copes better with the diversity of end user reporting needs and lends itself to a smooth transition from the current standard architecture. This could even be the recommended approach when the existing DW/BI platform is an aggregation of multiple technologies.

The standard key success factors of a successful DW/BI initiative still remain valid:

1. Any DW/BI project must start with a scoping phase that collects user reporting requirements and the overall context. Part of this phase is the clarification of multidimensional analysis needs for which OLAP services will be required.

2. Companies need to develop metrics to measure the success of reporting projects and a Return-On-Investment framework as for any other IT project.

3. Finally, the management must be ready to embark on an iterative process. An appropriate change management strategy must be defined and applied in order to leverage the flexibility of the new DW/BI architecture.

In conclusion, the DW/BI area is undergoing fundamental changes. The timely delivery of comprehensive reports is more and more a key differentiator in today’s flood of information. In this context, the use of OLAP-centric tools must be reviewed and the technical architecture adapted. Even if perfection is not here yet, reporting can now get better and more importantly… timely!

---

**Architecture Corner**

Some of you may be laying out basic interoperability, evolvability and manageability principles for future Service-Oriented Architecture (SOA) developments and reference models within your organization. Interoperability deals with the definition of identifiers, formats and protocols (IFaPs). As an architect, you enjoy analogies and may be interested in knowing that in April, we celebrated the 50 years of the 20-ton packet network also known as intermodal containerized shipping. This successful set of container standards allows the packetization of (sometimes very large) goods at such low costs that the location of the country of origin no longer matters. When defining your interoperability principles, consider simple pervasive concepts for your IFaPs that will prevail for the way new, old and current enterprise components will talk to each other within your architecture.
If you currently browse through IT magazines and technical web pages you will have recognized that the use of the word “AJAX” increased a lot. Are they talking about the Netherlands football team, you might wonder? Or do they refer to the floor cleaner called AJAX? No, they do not – AJAX is a web technology offering “Rich Client feeling” in a browser based environment. Rich Client feeling means the usage of high interactively desktop like front ends with a sophisticated communication model between client and server. The underlying communication model keeps the web pages stable in the browser instead of reloading (so called “flickering”) after every user interaction. To put it in a nutshell, AJAX offers the possibility to create complex web front ends which mirror standard application feeling used commonly throughout organizations today. An interesting example of this is the recently launched Google Maps which is AJAX-based.

Note that AJAX is not a new invention: it rose already in 1997 due to the introduction of so called remote scripting. Invisible pages were used in order to cover the communication stack between client and server (picture 1).

As traditionally witnessed in IT, technologies need a certain amount of time in order to gain visibility to the market and be adopted by the end user.

This particular innovation has been started in 2005 when Jesse James Garret introduced in an essay the word AJAX, which stands for “Asynchronous Java And XML”. Since that time there has been a lot of movement within the Web User Interface space, independently from AJAX itself. New technologies and frameworks emerged and based on these technologies new possibilities opened up to developers and end users. But (un)fortunately we are still at the beginning of the wave - a single default set of patterns for the development of AJAX applications has not yet been established, which makes it sometimes extremely difficult for customers investigating new or legacy Web frameworks to decide which AJAX path they want to follow. Choosing the right technology for the implementation of a browser-based application has become much more complex since there are more than 100(!) frameworks available.

**Picture 1:** AJAX communication model
Instead of „talking“ to the server directly the web page uses another hidden web page in order to synchronize data. Thus the upper page is kept stable in the browser!
As a consulting organization, we are periodically confronted with this challenge and have invested some efforts within the last months evaluating key solutions which we find most viable from both functional and non-functional standpoints. We believe that it is important to embrace the entire presentation stack for AJAX applications and anticipate the use of related technologies both for legacy and new applications.

While AJAX adds high impact to your web applications, it also adds value to the corresponding development process as well: AJAX applications can be developed much faster due to the use of frameworks and predefined libraries. Think back to the old “Visual-Basic times” and now imagine a development process for web applications which is similar to this: front ends can be created without the need of writing HTML code and the developer focuses completely on the Java business logic. This simplification of the development process has been adopted by our teams which adopt a prototype driven development process.

Considering the potential of AJAX-based web applications to compete with traditional desktop-based applications used today in all organizations, we touch at one of the key features of Web 2.0 and the offering of Software as a Service (SaaS). While past initiatives providing CRM or ERP solutions by ASP players have been plagued with usability issues related to the basic use of Web technologies, AJAX and Rich Internet Applications reshape the landscape and place Web 2.0/SaaS players in a favourable position. Make sure you are prepared well for this evolutionary approach of web technologies because earlier or later there will apply one saying: the browser will be your computer!

**Picture 2:** Resource Management application with lists and trees.

---

**Cambridge Survey**

Looking for IT Value Protection? Is it about **IT Governance**?

We have an on-going survey in progress, looking for this answer and focusing on:

- The expected benefits of an ITG initiative
- The ITG products and frameworks market today
- The usual ITG organizations, in your business and globally
- The ITG budget as a percentage of the IT budget: yes but how much?
- The maturity of the IT investments decision process.

By the end of the autumn, we will be please to share with you the result of this survey and the approaches we propose to protect the value of IT, within the four **IT Governance** domains:

- Portfolio Management (Business Services Alignment)
- Enterprise Architecture (Business Services Architecture)
- IT Service Delivery (Business Services Planning)
- IT Service Support (Business Services Execution)
Cambridge as link between Business and IT
Customer care at Helsana – an area-spanning CRM Project

Overview

Helsana, Switzerland’s largest health insurance provider, follows a customer-oriented approach and offers its services to both private and corporate customers. The annual total of premiums generated amounts to some 5 billion Swiss francs with the total number of insurees surging beyond 1.8 million for the first time in 2006. www.helsana.ch

Challenge

Helsana needs top-rate instruments to maintain and further expand its market position.

Customer care was identified as the main area of concern; considerable scope for improvement was identified, for instance, in terms of consistent and automated processes in the call center and access to vital customer information. Most everyday tasks were carried out manually, which resulted in poor consistency and slow processes.

The challenge Helsana was facing consisted in the implementation of improved processes combining enhanced dataset quality and optimised process cycle times.

In this scenario, the application landscape is the backbone ensuring safe and stable operations, especially during the year change, the key period in the health insurance business. The stability and safety thus created has positive repercussions on customer and user satisfaction.

Solution

Teams from the private and corporate customer business, supported by IT (under the leadership of Ulrich Schmid, member of the Helsana Executive Board), jointly pursued the ‘Vision 9’ project whose goal it was to create a consistent platform for all users and to improve customer relation processes. Vision Consulting AG (www.vision.ch) was chosen as partner company for the technical implementation and provider of the CRM software VisionEnterprise, whereas Cambridge Technology Partners backed the project in terms of project planning, reporting, task and risk management, cost control, internal and external project communication and acted as interface between IT and the individual business areas. The interface-related tasks entailed the analysis and area-spanning reconciliation of requirements, translation of requirements into technical specifications and eventually the coordination of the different IT departments for the technical implementation.

For the associates, a graphical user interface containing all relevant data was created, enabling them to lead customers through the conversation from call reception through to the details of service provision. This favours a single-minded approach and allows staff to gather all the information required.

The basic process has remained unchanged since the introduction of the solution. As soon as the customer is identified, an entry mask containing all master data will appear on the customer relations manager’s screen, thus providing him with all the customer information from the very beginning. In a further step, the customer relations manager will discern the customer’s concerns and trigger a lead workflow.

Over time, the solution has developed to become more than just a run-of-the-mill CRM solution. In fact, it facilitates access to information for all departments dealing with customers’ concerns (the claims-settlement department among others) and integrates a whole range of additional distribution channels like the internet (e.g. requests for offers from external price-comparison portals such as comparis.ch), the outbound call center and the link-up with subsidiaries.

Today, the system provides up-to-date information to some 2,000 employees and supports all customer-related processes.
**Result**

The team has mastered the challenges of the project posed by time pressure due to year-end operations and market cultivation as well as by interests and dependencies vis-à-vis other projects at group level and has reached the following milestones in a phase-oriented project approach:

**Phase 1:** staggered introduction of electronically supported basic customer care processes (beginning of 2005)

**Phase 2:** introduction of process automation, staggered expansion of existing processes and link-up with a further organisational unit (beginning of 2006)

**Year change as a touchstone for enhanced productivity**

In the business year 2004, the first year after the introduction of the solution, a total of 1.5 million workflows were managed, more than 5 million activities triggered and 250,000 offers submitted to new customers. As per year-end 2004, Helsana had acquired some 170,000 new customers, an achievement unthinkable without the introduction of the new, integrated CRM platform.

In the business year 2005, the consistent introduction of process automation enabled Helsana to conclude many of the administrative year-end tasks at a significantly earlier stage than in previous years.

“**CTP has made a major contribution to the successful introduction of our CRM system thanks to target-oriented risk management, solid project management and a practical understanding of CRM and technology that was conveyed.**

What I see as key was the ability of CTP to approach all the various stakeholders on the business and IT sides individually and get them on board.”

Ueli Schmid, Leiter Informationsmanagement

---

**Seamless Integration**

Since its launch as Java standard for the implementation of user interfaces in web applications, JavaServer Faces (JSF) has gathered increasing popularity and has come close to outranking the current industry standard Struts as number one amongst MVC frameworks. JSF boasts extendable component architecture, generates client-specific markup and enjoys growing support in IDEs.

As is often the case with standards, their specification cannot cater to all needs of the developers community. It is therefore not surprising that open source or commercial frameworks fill in those gaps. Seam, which was developed by JBoss and whose version 1.0 was recently launched, is one of them.

JSF is a remarkably flexible and extendable standard; Seam avails of that to correct some of the shortcomings and provide support to developers.

The most striking novelty is the direct integration with EJB 3.0. So far, UI components had to be tied to so-called backing beans, an additional layer between the UI and the service layer. Through simple annotations, Seam makes it possible to use stateful or stateless session beans directly as backing beans, which can help save significant glue code, replaces the conventional declarations of backing beans in XML by annotations and harmonises the component models of JSF and EJB 3.0. Backing beans, so far implemented as simple Java objects, now feature transactional capabilities when their methods are combined with the Seam approach.
The second significant innovation is the introduction of new contexts which regulate the life cycle and visibility of data. On top of the contexts known thus far, such as the request or session contexts, a conversation context, a business process context and an event context will be introduced, with the conversation context bearing the greatest potential for frequent use. The conversation context models a work unit from the user’s point of view, for instance the reservation of a ticket. Seam thereby takes care of potential stumbling blocks like back buttons, multi-window support and de-/allocation of resources at the beginning and end of the conversation. With the current JSF model, this would have to be solved via backing beans in the session context which, in larger applications, constitutes a potential source of memory leaks or errors.

The business process context supports long-running processes with multiple users with an integration option directly with BPM by JBoss. In this way, page flows can be controlled directly from BPM and actions can be carried out directly from page flow transitions.

Some other small Seam innovations completing JSF deserve to be mentioned. In JSF, for instance, navigation on the actual sites containing the search results is not permitted (bookmarks). Via annotations, Seam allows for the simple definition of corresponding request parameters; what is more, a tag library in Seam allows easy integration in the user interface.

**Conclusion:**

The innovations outlined above make Seam an attractive extension of JSF. The open JSF specification, straightforward integration via annotations and decent documentation significantly facilitate the first working steps for developers. Further JBoss technologies such as jBPM, Embeddable EJB Container or Hibernate/JPA promise an entirely integrated architecture stack and should definitely be kept in view.