

White Paper on Bell Canada ORS Business

Overview

The Inventor of Telephone in late 1800's and patented in the year 1876 by Alexander Graham Bell. Bell Canada was formed in 1880's and expanded to a telecom giant in over the century. To find more about Bell Canada please visit <http://www.bce.ca/en/aboutbce/history/index.php>

Now Bell Canada is one of the largest distributors of telecommunication products in North America. Bell Canada provides telecommunication solutions for consumers and business groups like Small Medium and Enterprise. Bell Canada offers Data, Voice, Convergence, Wireless, Cabling Products and Services to end customers.

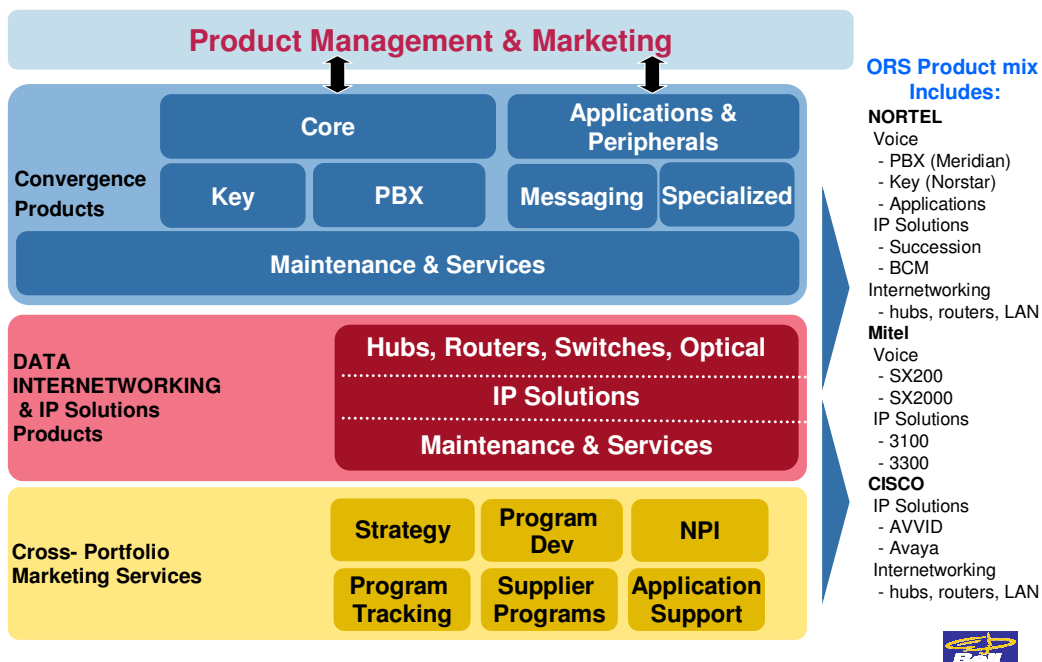
Background

Bell Canada's business focuses on Small Medium and Enterprise customers with a wide range of products, programs and service offerings. The industry solutions include

- Data Connectivity solutions
- Voice Connectivity solutions
- Pay-for-Service
- Messaging Services
- Global Solutions
- Information & Communication Technology (ICT) Solutions
- Industry Solutions
- Outright Sales and Solutions (ORS/OSS)

Bell Canada ORS business platform offers the following product hierarchy

ORS Sales Process



Business Problem:

Bell Canada's Out Right Sale (ORS) business face the challenge of changing product line carried out different business process and sale process with complex product configuration and pricing requirements. Keeping the product configurations up-to-date with associated service requirements seemed to be a greatest challenge the business had always faced.

Requirements:

Bell Canada wanted to simplify the business process by introducing a unified set of configuration, services, pricing, and workflow with integration services on the web (Intranet and Internet) with the following features

- Access to the Large Customer Data from SAP Systems
- Configuration Engine to manage the changing product lines
- Based on the configuration mapping the complex service calculations internally
- Managed Pricing and Discounting of the products, programs and services
- Smart Workflow to reduce the quoting life cycle
- Allow sales to process transactions in real time
- Notification and follow up integration services
- Document Management for the online ordering system
- Reporting and Tracking facilities
- Increased customer loyalty and revenue
- Building quick and correct solution the first time

Challenges:

In building a complex order processing system on web based architecture have some of the following difficulties

- Performance
 - Bell Canada ORS has got nearly 4 million customer locations with associated or linked customer of 12 million in total. Also the data is SAP system, would require connectors extract data on the web
 - Product Catalog search capabilities and vendor product configuration mapping to product info base
- Configuration
 - Bell Canada supplies telecommunication products from various vendors, managing the product configurations each vendor inside Bell Canada system is too difficult to maintain
- Integration
 - One single web based system to interface with Billing, Supply Chain, Contract Management, RMA, Back Office systems would require huge amount of time in integration and maintaining
- Managing Services and Pricing
 - The complex selling system required functionalities like Data Mapping from External/Internal configuration tools to quoting system with accurate pricing logic. Mismatch in pricing between quoting system and billing system would result in billing dispute.
- Workflow Management
 - Reducing the quote to sold time frame by automation of manual workflows into system controlled workflow using email integration, tracking and notification services

Solution:

The professional services team was assigned to review the Bell Canada ORS business process and come up with a solution which will serve above 500 concurrent users to connect on the web and build quotes for the end customers. With the introduction of Unified Tool and Unified Database modules all the front end functionalities were grouped under Unified Tool (UT). When the quote becomes a successful sale the order is transferred over to Unified Database for back office processing and release to SAP for order billing and supply chain management.

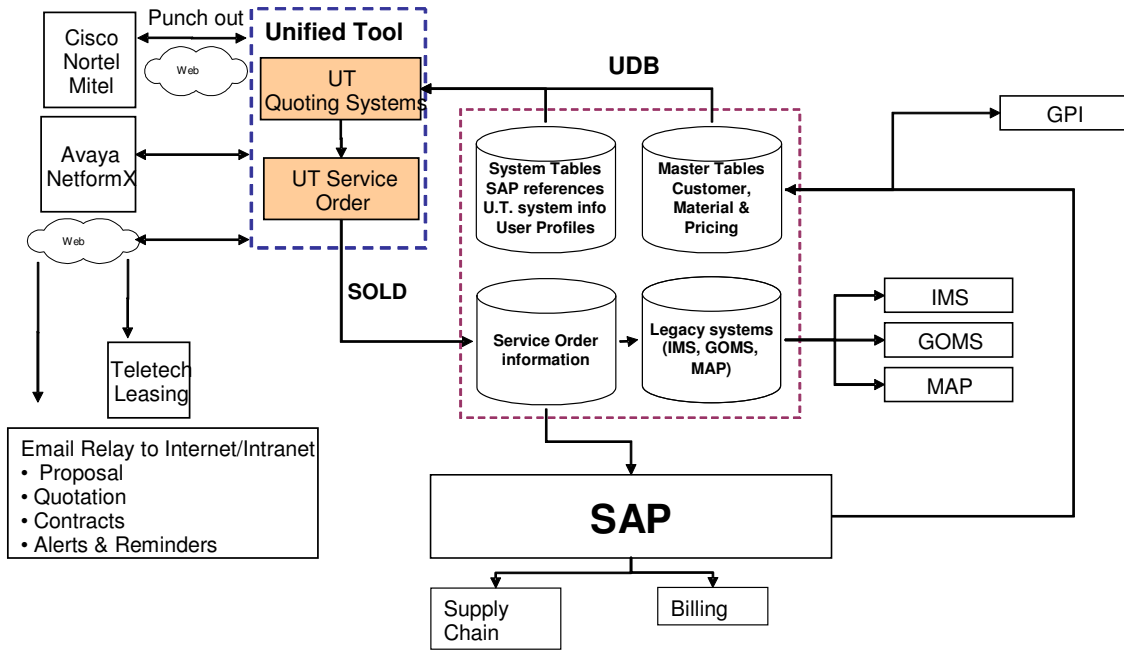
UT is a sales quoting system for engineered products that runs entirely on the intranet, enabling us to cut costs; increase efficiencies across front-end functions; and access current, accurate, and consistent data.

UT now provides quoting solutions for all ORS business for Bell Canada in integration with SAP system

Refer to the following diagram for the end-to-end system map

ORS Systems & Tools

High level view of ORS end to end Systems



System Architecture:

The Entire ORS system is split into three major modules

- Unified Tool - Front End systems
- Unified Data Base - Back Office Systems
- SAP - Billing, Invoice, Supply Chain, Contract Management Systems

Unified Tool System Functionalities:

- Application:
 - Workflow and Access Control List
 - Each user flow has set of finite function in the quoting cycle with dynamic business driven access control
 - Guided solutions for users
 - Configuration:
 - Integrates with third party vendor sites for Product Configuration (**Cisco, Nortel, Mitel, NetformX, Avaya**)
 - Secured XML Data Transfer between Bell Canada ORS UT to vendor sites
 - Data Processing:
 - Based on user selections and Configuration data UT system would identify and generate appropriate service tasks for pricing
 - Generate additional user workflows for approvals where ever needed
 - Use of Rule Engine to build and maintain all the business rules
 - Pricing and Discounting
 - The products and services pricing and discounting module is controlled by the Pricer module.
 - Uses Rule Engine behind the application to maintain all the business rules for easy maintenance
 - Maintains the vendor and Bell Canada programs as part of the discounting module
 - Document Management and Notification Services
 - Proposal documents, Quotations, Forms and Legal Documents generation dynamically
 - Bar Coding Solution
 - Reports in Excel, PDF, HTML
 - Service Order Module
 - Complex data mapping module uses Enterprise Data Access Components
 - Data transfer to and RDBMS system
 - Data Transfer using SOAP
 - Integration to **UDB/SAP**
 - 500 users support

Unified Database Functionalities

- Application
 - ASP (Active Server Pages) and Java
 - Microsoft Platform with SQL Server 7.0 Database Servers
 - Integration with external SOAP Servers
- Modules
 - 200 users support
 - Help Desk tool
 - GPI – Product Info base
- Integration
 - UT and SAP
 - Legacy systems (**IMS, MAP, GOMS, Price Book...**)

SAP Functionalities

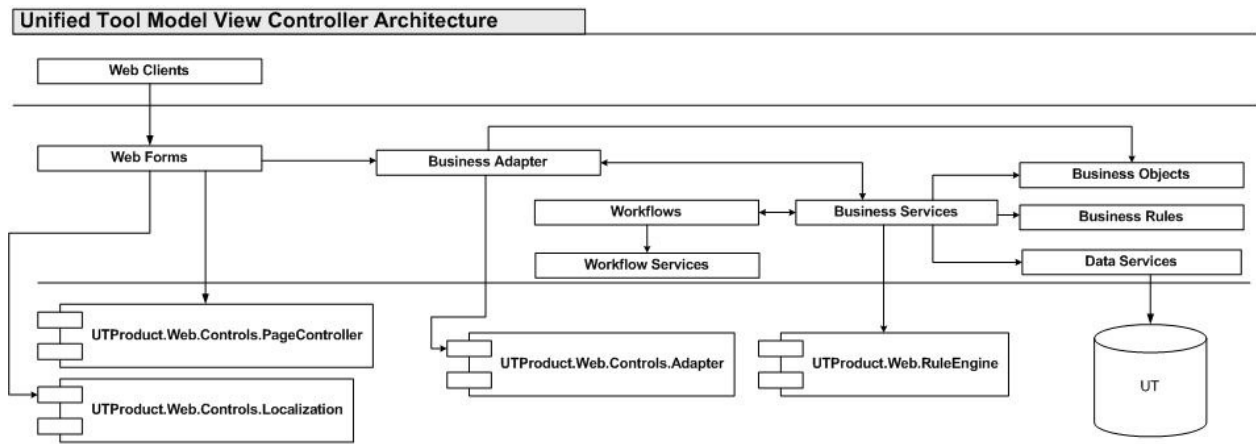
- Application
 - SAP version 4.6
- Modules
 - 800 users (BO, Finance, Operations, CAG...)
 - Sales & Distribution
 - Finance
 - Material Management
- Integration
 - **FSOS, BRIP, ARICS, EDRIS, GL, UDB/UT**

UT Application Architecture:

UT was built using the model view controller approach

- Web Based Interface (Thin Client)
- Model View Controller Implementation
 - Page Controller Component
- Multi-tier architecture for future expansion and modularity
 - Web Interface, Business Services, Object Services and Data Services
 - Separation of Business Logic from Presentation Layer
 - Separation of Data View from the Business Logic
- Componentized Application Development
 - Application Data Adapter
 - Rule Engine Component
 - Globalization Component
- Structured Object Modeling
- Microsoft .NET platform (SDK 3.0)
- Use of Microsoft Enterprise Libraries
- Deployment in a distributed load balancing environment
- SQL 2000 Server

Refer to the following diagram for the application architecture



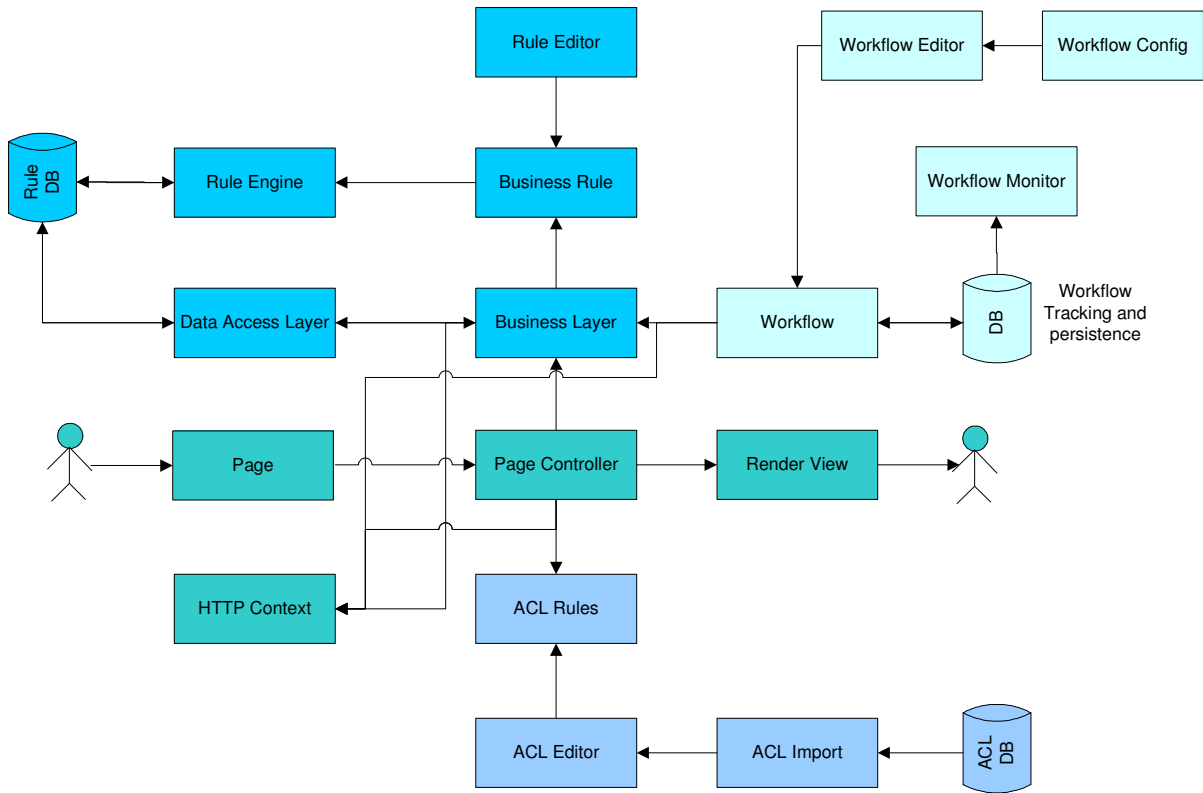
Product Components

The product hosts a set of components and libraries that is built over Microsoft's .NET framework. These components and libraries form an integrated framework layer. This framework exposes system level interfaces for applications to interact with. This layer facilitates the business components to interact with framework. The framework mainly targets the web-based application. The framework allows the developers to write, build and compile web applications. The framework comprises of tools and libraries, some of which are standalone and some, are integrated into the visual studio environment for the developers to build web applications.

The framework can be broadly classified into the following

- Page Controller Framework
- Adaptor Framework
- Rule Engine Framework
- Localization Framework

Architecture



Page Controller Framework

The page controller framework offers the following service

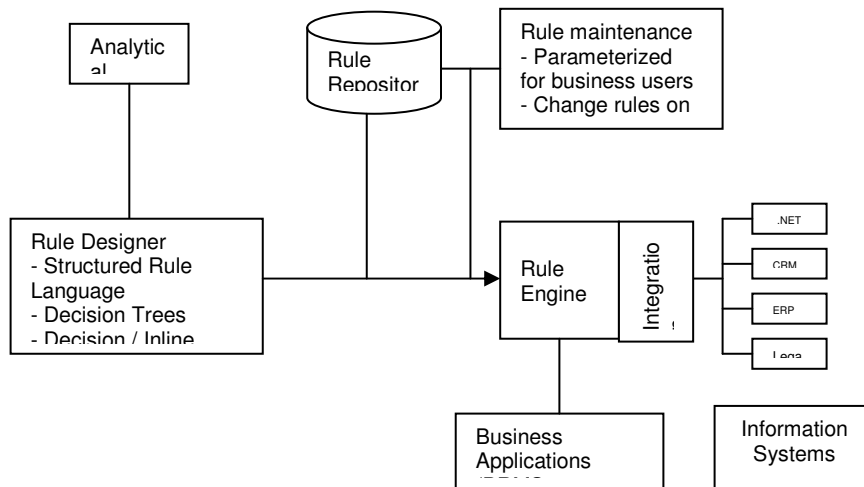
- Get list of all workflows
- Start a workflow
- Get current workflow state
- Get the page state
- Get Roles associated with each state
- Pass the workflow events from the users to the current running workflow
- Helps in page navigation by getting the current workflow state. The page controller makes decision based on the state of the workflow
- Queries host service to get status information's of each workflow instance
- Page controller creates Custom identities based on the application user. The identity holds user's roles. This identity is sent to the workflow host service which in turn sends it to appropriate running workflow instance
- Page controller looks up into the Http Context for user authentication information before executing the state.
- The page controller will make the error messages available to the developers.

Adaptor Framework

The Adaptor framework offers the following service

- Sits between the Page controller and the business layer
- The 'behavior' of a control is handled by the adaptor controller which uses the object datasource model
- Bind properties and method of the ASP.NET controls to the middle tier
- Passing data from the view to the adaptor within the functions as parameters, objects etc.
- The view may also be able to perform editing operations on its collection of objects, essentially UPDATE, INSERT, DELETE
- Two way control binding capabilities
- Custom classes to handle run time controls
- Avoid declarative syntax binding

Rule Engine Framework



The Rule Engine framework offers the following service

- **Rule Engine:** It controls and directs the selection and execution of the rules within applications and processes
- **Rule Repository:** Provide version control and contains a complete set of prior and current rules. The record of each rule contains information's about author, dependencies/interactions with other rule applications
- **Rule Designer / Rule Editor:** This is a rule design environment, where rule authors creates entities and define the rulesets. Within each rulesets the users makes use of structured rule languages and rule definitions, conditional statements, decision and inline tables
- **Integrated Framework:** They are a set of API's which external applications will use of while interacting with the rule engine. The framework enables different application built on different platforms to consume these API's to execute their business rule. The framework feature are

- Applications can fire all the rules within an rule app
- Application can explicitly fire individual rulesets within a rule app
- The rule applications are auto loaded once they are invoked from the application
- A rule session is automatically created for every rule app. The state of these rule sessions is persisted in memory. The state management of these sessions are taken care by the framework
- User has the choice to enable the state persistence mechanism
- Notification's generated by the rule engine in form of information, error or warning are made available within the application
- Constraints fired due to violations of rules within the rule app are made available to the external application
- Errors from the rule engine are made available to the external application
- Query value list can be fetched from the rule app.
- The framework needs to be made aware of the all the rule apps. This is achieved using configuration driven files

Localization Framework

The Localization framework offers the following service

- The framework accommodates custom controls that have been derived from base web controls and additional custom properties have been implemented to support localization
- Localization can be achieved by using the "Key" property for each of the controls either in the declarative syntax or in the code behind for run time controls
- Configuration driven resource files
- Capabilities to toggle between multiple languages at runtime without restarting the application
- Thread safe Hash tables used by resource manager to store resource information
- Faster access to the resource information as they are readily available from memory.
- Does not use reflection / File I/O operations during every request or during rendering of the controls
- Developers do not need to invoke any localization API's manually

Development Platform

- Microsoft Windows XP
- Microsoft IIS Web Server 6.0
- Microsoft Visual Studio 2005
- SQL Server 2000
- ASPOSE Excel PDF Components
- Fly Tree View Controls

Testing

- Internal Testing
 - Unit, Functionality Testing
 - Stress Testing, Load Testing
 - Traditional code testing – black box, white box, boundary value
 - System/Application Flow Testing
- QA and Business Analyst Testing
 - Technical Design reviews

- Code Inspection
- User interface testing
- System testing – Functionality, stress, installation and operation
- Regression testing
- Testing Tools
 - Ants-Memory profiler was used to identify the size of objects. The tools helped in identifying how many objects were alive and how many objects had been destroyed.
 - Ants-Performance profiler was used to identify bottlenecks in the user code. It helped in measuring the time taken for each of the individual function calls.
- Performance and Application tuning
 - Performance Measurements
 - Measuring the Web Applications request, response and throughput time
 - Measuring the web request hits/ seconds and application level errors
 - SQL Performance Monitoring during the different phases of testing cycle
 - Monitoring and measuring network bandwidth and network latency
 - Monitoring and measuring Rule Engine load and processing of rules
 - Measurement of Load balancing Components
 - Following sections were optimized based on the performance measurements
 - Code Fixes (Code Review)
 - Web Server (Creation of Application Pools)
 - Database (Stored procedures, Generating Normalized Data Views, Data schemas, etc.,)

Deployment Platform

- Microsoft Windows 2003 server
- Microsoft IIS Web server 6.0
- Microsoft .NET SDK 3.0
- Scale Out Load Balancing Software
- SSL 2.0
- SQL Server 2000

Application Support

- Impacts of **technical** and **process** changes
- Organizational and market requirements changes

Infrastructure

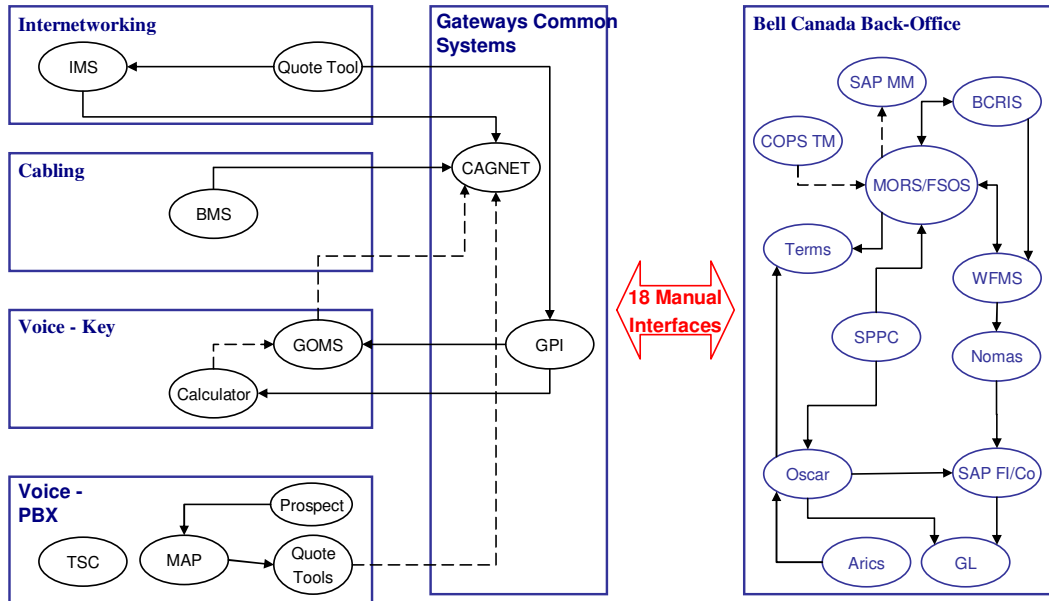
- On going server/application management to meet the demand of ORS business

Conclusion:

A comparison of the system architecture between the past and current is shown below. In the past all the ORS systems were highly disintegrated with huge amount of manual process.

ORS Systems & Tools

Previous Systems Architecture



Current Bell Canada ORS systems are now functioning as an end-end automated system. Since all systems are connected and efficiently built using the latest technologies and components the quote to order cycle had been brought down from days to hours.

Refer to the following diagram for the end system integrations

ORS Systems & Tools

Current Systems Architecture

