Reinventing the enterprise with communications-enabled applications

by William Becchina, Larry Ciccarelli, Claire Giotis, and John Kenny

More and more enterprises are seeking to transform themselves by embracing a business convergence model that aligns their investments in information technology (IT), telecommunications, and business applications in a way that enables them to drive the next generation of revenue growth and productivity. Nortel is helping to accelerate this transformation by empowering business applications with a wide range of network communications capabilities – such as contact center, interactive voice response, unified messaging, and unified communications – that will speed up collaboration and business processes, improve agility, and increase productivity. To help achieve this integration, Nortel is leveraging the Service-Oriented Architecture framework to enhance its portfolio by making it more capable, customizable, and responsive to customer needs. Nortel is also building a GUI-based Service Creation Environment tool that will enable services to be rapidly created and tailored to implement particular business processes.

Enterprises are facing major challenges in how they adapt to the constantly changing needs of customers, partners, suppliers, and other external entities while at the same time supporting their internal users, business information systems, and communications platforms.

Specifically, in today's world, enterprises need to automate business processes that span different systems, media, and enterprise boundaries, while providing real-time information to a broad user community (employees, customers, suppliers, and partners) and ensuring that the information is current and consistent across all systems. As well, enterprises increasingly require rapid deployment of secure and highly available services, along with the flexibility to extend services at a pace driven by their user communities. Many enterprises also face not only a challenging mix of new and legacy platforms, but also a rising demand for applications that incorporate the advanced capabilities inherent in telecommunications networks.

Our goal at Nortel is to empower the myriad of enterprise applications with Nortel's rich set of communications enablers – such as contact center, interactive voice response (IVR), unified messaging, and unified communications – while ensuring that the network is applications-aware and able to provide the security, quality of service, and other capabilities needed to flexibly support a wide variety of enterprise applications, each with its own set of requirements.

To expedite this business convergence in enterprise applications, Nortel is pulling together its broad base of technologies and product portfolios – spanning both enterprise (including customer premise solutions and applications) and carrier networks (including wireline and wireless) – both to deliver the required carrier-grade scalability, security, and reliability and to enhance the user experience by simplifying interaction in a rich applications and services environment.

In contact center applications, for example, Nortel's Expert Anywhere solution (which provides such communications capabilities as presence and call-routing rules) is used to extend the customer's reach beyond the contact center operator to any designated knowledge worker within the enterprise. This capability enables businesses to route customers to the most appropriate agent or knowledge worker regardless of location or function, replacing previous call-routing strategies that were based on agents' physical locations.

Nortel is also applying such speech technologies as natural language speech recognition, text-to-speech, and speaker identification to increase agent productivity by automating routine customer inquiries, as well as to improve security by removing agents from customer authentication tasks.

To offer these and other capabilities, a contact center must be able to support a wide range of new technologies and provide the tools to configure, monitor, report, and integrate tasks in accordance with the enterprise's business rules.

The Nortel Applications Center (NAC) suite provides an open design that permits seamless interworking of all Nortel's Contact Center applications, as well as open interfaces to all enterprise applications, such as customer relationship management (CRM) or workforce management (WFM). The NAC extends the proven capabilities of the Contact Center applications with self-service speech and Session Initiation Protocol (SIP) capabilities to

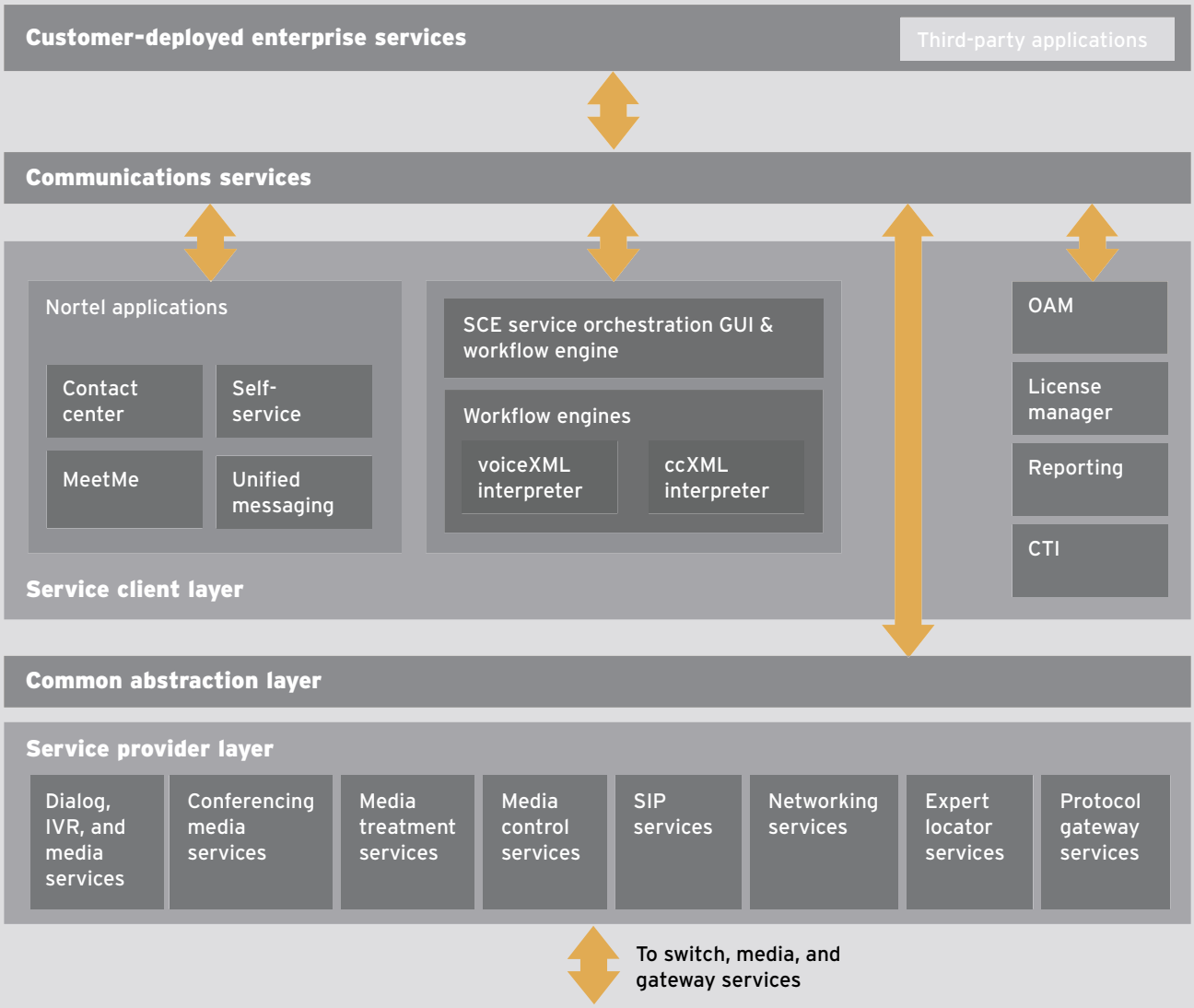
build a truly loosely coupled product that is uniquely adapted for a Service-Oriented Architecture (SOA)-based enterprise.

To extend the value of its contact center and other communications applications, Nortel is also moving rapidly to build a large and solid ecosystem of key industry players, including Microsoft, IBM, Hewlett-Packard, SAP, and Accenture, to develop collaborative technology solutions. Nortel's alliance with Microsoft, for example, will help enterprises transform their business communications by speeding the transition to voice over IP (VoIP) and unified communications (see page 16).

Flexible solution for applications

The key convergence point for network functions and business applications is the fast-emerging SOA methodology

Figure 1. Nortel Multimedia Applications Service-Oriented Architecture



This figure presents the architectural concepts of the Nortel Multimedia Applications portfolio – encapsulating Nortel's Contact Center portfolio, Unified Messaging, Media Application Server, and Communication Control Toolkit – and its logical alignment with Service-Oriented Architectures (SOAs). The architecture is organized in a layered format.

The uppermost layer, customer-deployed enterprise services, represents the enterprise's business processes and portals. This layer provides the integration point for the customer's IT infrastructure, the underlying communications services provided by Nortel (which will be presented as Web Services in a SOA environment), and third-party applications (such as Microsoft desktop applications). The customer's IT infrastructure may include customer portals (for example, web pages and email systems), general business applications (order entry and management systems, backend database integration, and so on), and perhaps specialized customer

(see page 20).

There is no doubt that the need for applications to share information and interoperate has always existed. Many potential solutions – from such early middleware solutions as Common Object Request Broker Architecture (CORBA) to recent enterprise application integrations that utilize XML messages sent via HTTP – have come and gone in the past with varying levels of success.

Unlike these past solutions, SOA clearly presents an opportunity for next-generation enterprise IT, communications, and business applications integration – both within the enterprise and beyond with customers, partners, and suppliers.

To take advantage of this development, Nortel is judiciously applying SOA to provide open interfaces to its portfolio capabilities (presented as Web Services), as well as in its products to

relationship management (CRM) systems. From a SOA perspective, this layer can be considered as the Enterprise Service Bus (ESB) middleware.

Below this layer, the communications services layer is where functions of the Nortel communications infrastructure can be exposed (made available) as a set of services.

Below the communications services layer is the service client layer. The service client layer is where pre-packaged Nortel applications – such as contact center, conferencing (MeetMe), self-service (including speech recognition), and unified messaging – as well as out-of-the-box customer features such as the Nortel Service Creation Environment (SCE) graphical user interface (GUI) exist.

The applications and features can be normalized and exposed as a simplified set of services to the communications services layer, hiding their significant underlying complexity. For example, the Nortel Contact Center Multimedia Manager "immediate and scheduled call back" Web Service provides a portal into a combination of skillset resolution, expert location, and automated click-to-call-back functions.

Complex orchestrations of communications services may also be created using Nortel's SCE GUI tool (see Figure 2) and exposed at the communications services layer. The SCE defines service orchestrations as logical workflows that leverage such workflow definition languages as voiceXML, an XML definition for creating voice dialogs, and CallControlXML (ccXML), an XML definition for complex call-flow management. These definitions are dynamically interpreted by workflow engines, which orchestrate the underlying service providers in real time to deliver the desired functionality.

Other supporting entities in the service client layer include:

- operations, administration, and maintenance (OAM), which may be used to configure service clients and service providers or which expose or retrieve configuration data to the communications services layer;
- license manager, which allows or prevents access in real time to applications and services based on purchased license options;
- reporting, which provides historical and real-time reporting for application and service usage; and
- computer telephony integration (CTI), which is the traditional approach to providing customer application integration to real-time communications. CTI continues to play a valid role where tight coupling to the underlying services is required and the loose coupling model of SOA may be inappropriate, such as in applications that rely on long-running sessions with significant asynchronous eventing.

The common abstraction layer normalizes the interfaces to the underlying service providers, supporting an element of service provider plug-and-play and enabling services to be independent of media (voice, video, instant messaging, etc.). It also coordinates and authenticates service client requests and service provider responses.

The service provider layer houses the individual service providers (capabilities, or enablers) that ultimately deliver the communications services. Nortel offers a wide array of providers, including:

- dialog, interactive voice response (IVR), and media services, such as the playing of prompts, speech recognition, and collecting of feedback;
- conferencing media services, including voice conferencing and instant message chatroom-like services;
- media treatment services, such as recorded announcements, tones, music, and email auto responses;
- media control services, including low-level media stream control, such as Media Resource Control Protocol (MRCP), Real-time Transport Protocol (RTP), and Real Time Streaming Protocol (RTSP);
- Session Initiation Protocol (SIP) services, including call control and user presence;
- networking services, including the ability to move customer contacts around the network and maintain associated contact data;
- expert locator services, including the ability to monitor the availability of experts in the enterprise and provide a gateway to route contacts to them; and
- protocol gateway services, including the ability to leverage remote providers.

Below the service provider layer, switch, media, and gateway services connect to the wider real-time communications infrastructure, which includes everything from Nortel PBXs and remote media servers to telephones.

enable rapid incorporation of additional functionality, such as presence and SIP-based voice or multimedia sessions.

With SOA, the component functions and services are loosely coupled, overcoming the problem of many of today's applications that are relatively static and tightly coupled (i.e., tied directly to products and network infrastructure). SOA enables applications to be built and deployed rapidly in a modular fashion by combining reusable functional components using a common Web Services interface and framework and orchestrating them into enterprise workflows that connect people and business processes.

Moreover, this loose coupling of services makes it easier to fully automate the delivery of requested services to users, as opposed to agent-assisted or other methods of interaction. Since services in a SOA framework are independent of each other, the choices available to the user can be presented in parallel as opposed to a hierarchical "tree" structure, empowering users themselves to select the exact means to accomplish the task at hand.

Certainly, there are challenges in integrating real-time and often asynchronous communications services into a SOA environment, but an advanced SOA that employs an event-driven architecture (described later in the article) will play a key part in solving this integration issue.

SOA provides the framework through which network capabilities (such as presence, proximity, and identity management) and advanced Nortel services (such as multimedia customer interactions, click-to-call, Expert Anywhere, presence-based call-backs, and natural-language-based self-services) can be tied together with business applications (such as those available from Microsoft, IBM, SAP, and Oracle). To help achieve this integration for Nortel's rich existing suite of functionality and services as well as new features, we are leveraging the SOA framework to enhance our portfolio to make it more capable, customizable, and responsive to customer needs (Figure 1).

Creating services

Defining and creating communications services within the SOA framework and orchestrating them to work together and with other Web Services as part of an end-to-end enterprise workflow is being addressed with Nortel's Service Creation Environment (SCE).

The SCE is a graphical user interface (GUI)-based tool that enables users to quickly architect, test, and implement effective custom communications services required to not only meet an enterprise's unique requirements but also differentiate it from the competition. The tool, which leverages such open standards as Voice Extensible Markup Language (VXML), will be used by Nortel Global Services, channel partners, and end customers.

> Our goal at Nortel is to empower the myriad of enterprise applications with Nortel's rich set of communications enablers – such as contact center, interactive voice response, unified messaging, and unified communications – while ensuring that the network is applications-aware and able to provide the security, quality of service, and other capabilities needed to flexibly support a wide variety of enterprise applications.

From a business transformation point of view, the SCE supports a key principle in migrating services to a SOA: leverage existing functionality where possible and introduce new functionality where necessary. (For more on the SOA transformation challenge, see page 38.)

The SCE enables rapid service creation through abstraction and service reuse. It abstracts the basic building blocks to a high level, incorporating as much functionality into as few building blocks as possible. Some of the building blocks are functionally equivalent to hundreds of blocks used in previous tools. As a result, when designers choose to use a block in the service flow, a high level of functionality can be achieved with relatively few clicks. This high level of abstraction also allows designers to quickly learn the appropriate use of each of the block types (Figure 2).

By default, all services in Nortel's SCE are reusable, which enables easy creation of higher-level composite services as well as the invocation of services from any SOA framework via Web Services. Composite services comprise calls to invoke other reusable services, as well as the custom logic to glue those other services together.

To construct composite services, the SCE will be delivered with a standard package of functions (such as queue-to-skillset, play prompt, collect input, and fetch data) as well as enterprise-specific functions (such as a corporate security function based on caller authentication or a call-routing engine based on natural language speech processing).

Using the SCE, a designer can quickly develop a flow chart that models not only the path a user will take to navigate a service but also the service interactions both with the user and with other services – a process known within SOA as service orchestration.

Service orchestration

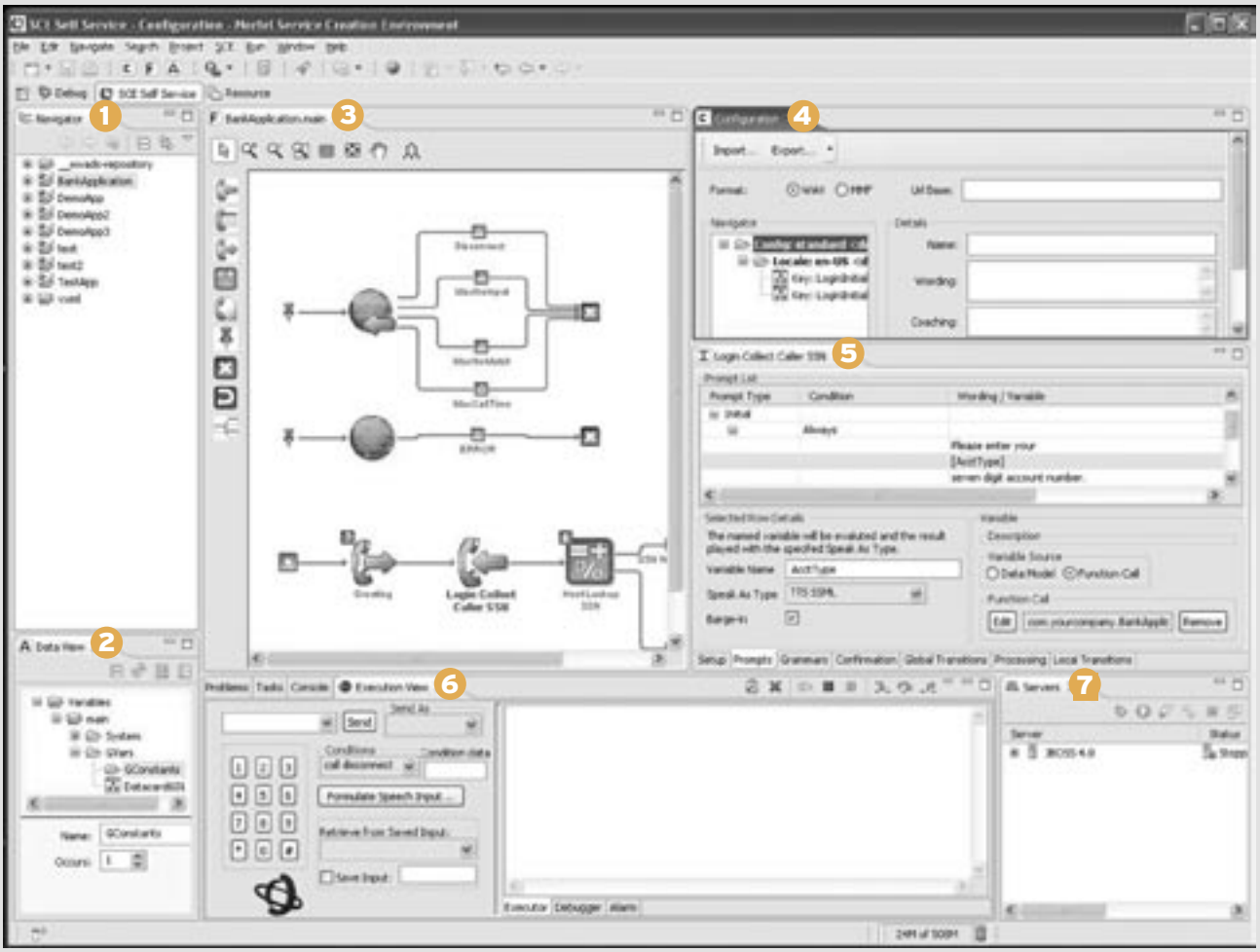
SOA makes use of loosely coupled Web Services to meet the requirements of business processes, and these Web Services can be orchestrated into workflows that define the order in which two or more Web Services are executed. These workflows represent business logic, and business logic that is intended to be reused should be encap-

sulated as a Web Service. The Business Process Execution Language (BPEL) is commonly used to define and represent business processes.

For example, service orchestration could be used to initiate a communications-enabled business process, such as an approvals process, which can become unwieldy when more than two people are involved. Suppose one member of the approvals process rejects the proposal under review. This event triggers the orchestration engine to begin a process that notifies each member of the rejection, including any information available, and also informs them that a conference call will be established to resolve the issue. The orchestration engine then monitors the availability of each member and when it locates a time when all are available, it automatically calls each member and places them into the conference. Additional capabilities, such as delayed calling or enhanced presence, could also be envisioned that would allow participants to prioritize their calls, setting rules to accept some high-priority calls while making themselves unavailable for other calls.

Basic SOA employs an interactive request/reply model where a Web Service publishes callable interfaces that can be used to build higher-level or composite services. In this model, the composite service sends requests to and waits for

Figure 2. Nortel's Service Creation Environment



Nortel's Service Creation Environment (SCE) is a graphical user interface (GUI)-based tool that enables a user to quickly build and test an effective communications service.

Developed as an extension to the Eclipse integrated development environment, the Nortel SCE augments the powerful toolset and flexibility of the open-source Eclipse platform by providing a set of views and editors for building a communications service.

As shown in this screen-capture example, the views/editors include a resource navigator (1) and data view (2) that give the user a broad view of the services files and data; the workflow editor (3), configuration editor (4), and block editor (5) that allow the user to edit details of the service; and the execution view (6) and servers view (7) that facilitate service deployment and testing.

The SOA transformation challenge

When deciding whether to adopt a Service-Oriented Architecture (SOA) approach to transform their IT and unified communications infrastructures, enterprises are generally faced with two initial considerations: first, to determine if SOA is the right strategic fit for their businesses; and second, how best to implement it.

The loose coupling of SOA promises a number of significant advantages, including IT development rationalization and component reuse, adaptability and flexibility to support rapid change, and a defined methodology to support integration of disparate technologies.

Although SOA would appear to have potential benefits for any enterprise, every case is unique. SOA is likely to merit consideration if the business faces some of the following challenges:

- IT and unified communications infrastructures are an integral part of business processes;
- business processes must continually adapt to meet market demands, driving constant change to the underlying IT and communications infrastructures;
- the business is consolidating post-acquisition IT and communications infrastructures that have disparate systems and technologies;
- business processes are adaptations or expansions of a set of core processes, which fits well with the SOA hierarchical approach to service creation and deployment;
- multi-channel customer portals (email, voice, instant messaging, etc.) are supported, but the core business service that uses them is generally the same;
- the business relies on external business-to-business (B2B) applications to complete its business processes, or would operate more efficiently if certain aspects of the business processes could be outsourced; and
- the business supports automated self-service portals, customer contact queuing and resolution, and unified communications that potentially could leverage customer relationship management software.

SOA implementation strategy

If the determination is made that SOA is right for the business, then SOA must be implemented using a well-thought-out strategy. A top-down approach, starting with a business process/IT alignment strategy is a prerequisite, and it should filter down through the various business, process, and infrastructure layers of a SOA.

This strategy starts by defining what services are required, consolidating and refining them into a set of core reusable and extensible services, and then considering how these services can be supported with such elements as boundary access, governance, security, discovery and registry, service definition, and potentially the creation of service providers (capabilities and enablers).

As the enterprise matures, implementation of a SOA will progress through five key stages.

Stage 1: XML and SOAP normalization wrappers

A reasonable starting point is to move to a common means of requesting and exchanging data or services across the disparate IT infrastructure. While SOA is a technology-agnostic concept, this stage recognizes that SOAP (which originally stood for Simple Object Access Protocol) and Extensible Markup Language (XML) are considered its *de facto* building blocks.

Leveraging these technologies enables the enterprise to wrap existing interfaces and create advantage by normalizing how data is exchanged and services are requested. The relative success of simply wrapping existing interfaces with SOAP and XML to expose services depends heavily on the nature of the legacy interface (for example, how granular, client-friendly, and secure it is).

The service consumer will also have no knowledge of the capabilities of the underlying service provider – sometimes referred to as the non-functional description of the service. Use of these newly exposed services will likely require significant manual intervention and external considerations for such functions as audit, security/authentication, throughput control, and version control for service capability updates.

This approach has been used within the Nortel Multimedia Applications space by the Contact Center Manager Administration web client in order to expose operations, administration, and maintenance (OAM) data. The same approach is being applied by Nortel's Multimedia Applications Professional Services team, which is taking existing formally supported interfaces, such as the Nortel Communication Control Toolkit application programming interfaces (APIs), and wrapping them with SOAP/XML.

Stage 2: Service definition, creation, and publication

At this next stage, an enterprise needs to take a fresh look at its underlying IT infrastructure. Service definition requires greater consideration, since services must become easier to locate and consume by target service requestors. The enterprise will also need to address how underlying service providers are abstracted and operate, which may require implementation of new service providers. At this point, organizing and publishing available services becomes important and the concepts of Web Services Description Language (WSDL) and simple service registries take effect. Service providers may implement a level of runtime governance, such as basic authentication and throughput-throttling

support. Given access to a central store for WSDL, service updates become easier to manage.

An example of this approach within the Nortel Multimedia Applications space are the "request scheduled/immediate call-back" Web Services (effectively "click-to-call" services) exposed by the Contact Center Multimedia Manager server. These services are released to customers with WSDL and reference web page implementations.

Stage 3: Composite services and service orchestration

As more services are developed, service orchestration may be applied to combine services to create new composite services, with flow control among the underlying services potentially carried out by a coordinating workflow engine (several of which exist within the Nortel Multimedia Applications portfolio).

Service orchestration is an important step in leveraging services to create flexibility and rapid response in a dynamic environment. This aim is supported if the ability to create or modify the service flow can be significantly simplified by providing a workflow creation graphical user interface (GUI), like that supported in the Nortel Multimedia Applications Service Creation Environment (SCE).

Stage 4: Advanced SOA services

The next step will likely require formal services middleware – an Enterprise Service Bus – to support the expanding SOA IT infrastructure. As an increasing number of services need to be coordinated between enterprise-wide service requestors and service clients, formal consideration must be given to the concepts of governance, advanced security, and quality of service control. As well, registries may become more complex, supporting publish/subscribe/notify intranet Universal Description, Discovery, and Integration (UDDI) servers. As well, coordination of services moves further up the SOA layers, leveraging Web-Services-centric Business Process Execution Language (BPEL)-compliant workflows, creating a hierarchy of workflow engines that extends from the customer portal all the way to the enterprise desktop.

Stage 5: Public service consumption

The final stage in SOA deployment occurs when the business subscribes to or publishes (exposes) its own services to and from the Internet or other businesses.

replies from subordinate lower-level Web Services. Typically, communication is via a remote procedure call (RPC) parameter list, and the service must be available when it is called.

Advanced SOA complements the basic request/reply interactive model by adding event-driven architecture (EDA).With EDA, Web Services post, receive, and publish event messages and/or subscribe to them. Communication is between peer Web Services that exchange messages and/or documents. The sending service sends a message, and in the case of EDA, it does not need to know whether the receiving service is there, and it is therefore more loosely coupled. In the EDA model, the Web Services and higher-level composite service have independent transactions, unlike the interactive model where the scope of a transaction covers both. These independent transactions also allow for looser coupling.

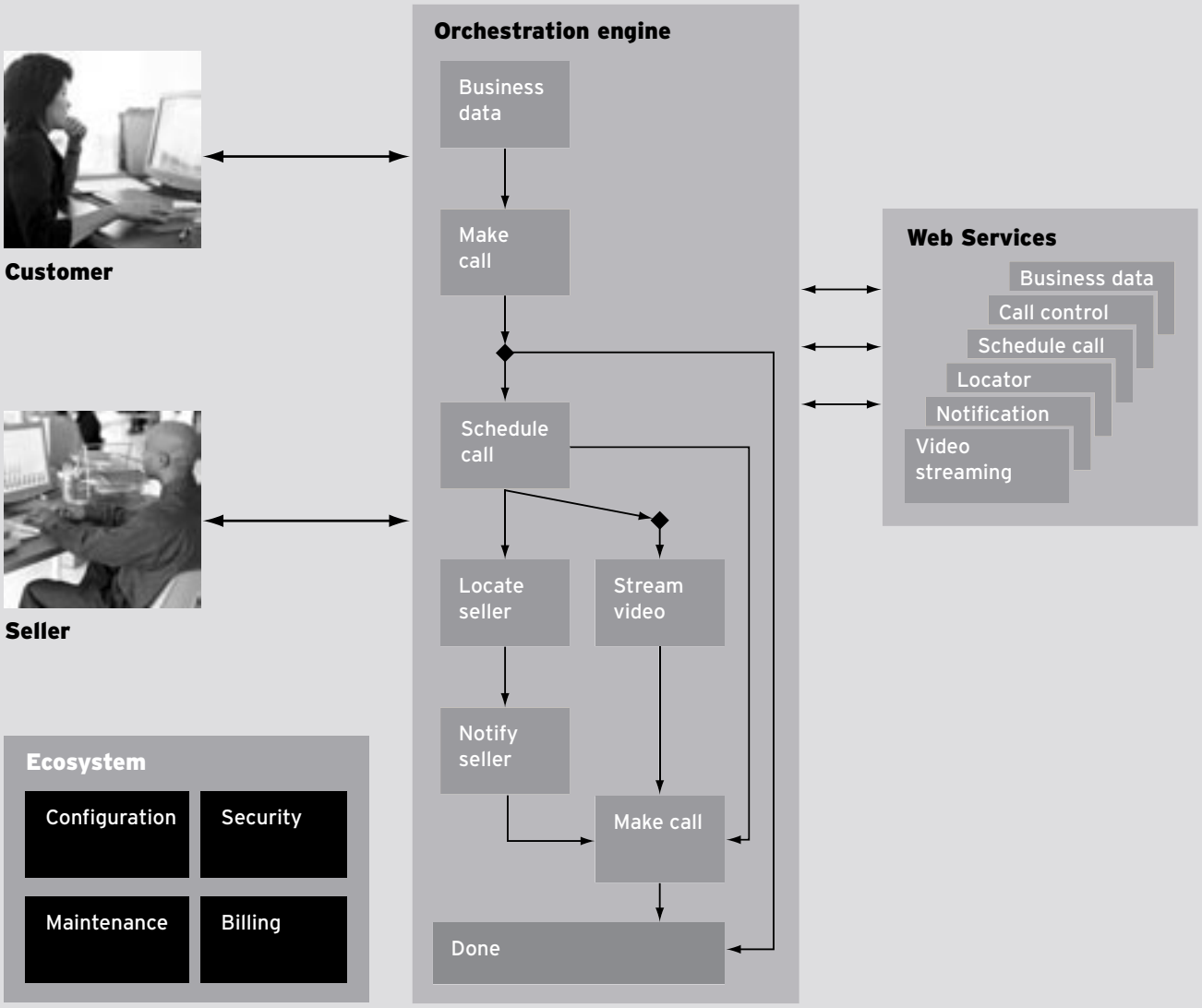
To support both request/reply interactions and EDA event notifications, Enterprise Service Bus (ESB) middleware is being developed. An ESB is a standards-based software infrastructure that allows services, such as Nortel communications services, to be shared across the enterprise to support various business processes. In a SOA framework, an ESB simplifies the integration and reuse of service components by making it easy to dynamically connect, mediate, and control these components and their interactions.

Using an ESB, communications services will be able to identify themselves as available for use, and all users can take advantage of these capabilities by incorporating them directly into a business process. Eventually, the ESB will extend outside the enterprise to include the entire enterprise ecosystem of partners, suppliers, and customers.

Nortel is investigating potential tool vendors that can provide this ESB functionality. To assess the value of these third-party tools, we are defining and prototyping various use scenarios.

To implement an event-driven be-

Figure 3. Web Services orchestration



Enterprise business processes are implemented by orchestrating a rich variety of Web Services into workflows that proceed through three fundamental interactions: trigger, invoke, and associate.

In this example, a customer is viewing product information on a web page and requires further assistance from the seller. The customer **triggers** the business process by clicking on a web page to speak to the seller.

The orchestration engine receives this request and **invokes** various Web Services (shown on the right in the figure), beginning with the business data Web Service to retrieve customer data (e.g., contact information) and then the call control Web Service to establish a voice call between the customer and seller.

If the call cannot be established (e.g., the seller is not present), the customer can schedule a call for a later time when both the customer and seller are present by having the orchestration engine invoke the schedule call Web Service.

While orchestration continues to execute, the schedule call Web Service monitors the presence of both the customer and seller. In the meantime, the orchestration engine invokes the locator Web Service to locate the seller, and the notification Web Service to notify the seller of a scheduled call. At the same time, the customer has the option of selecting to view a product-related video, and the orchestration engine invokes the video streaming Web Service.

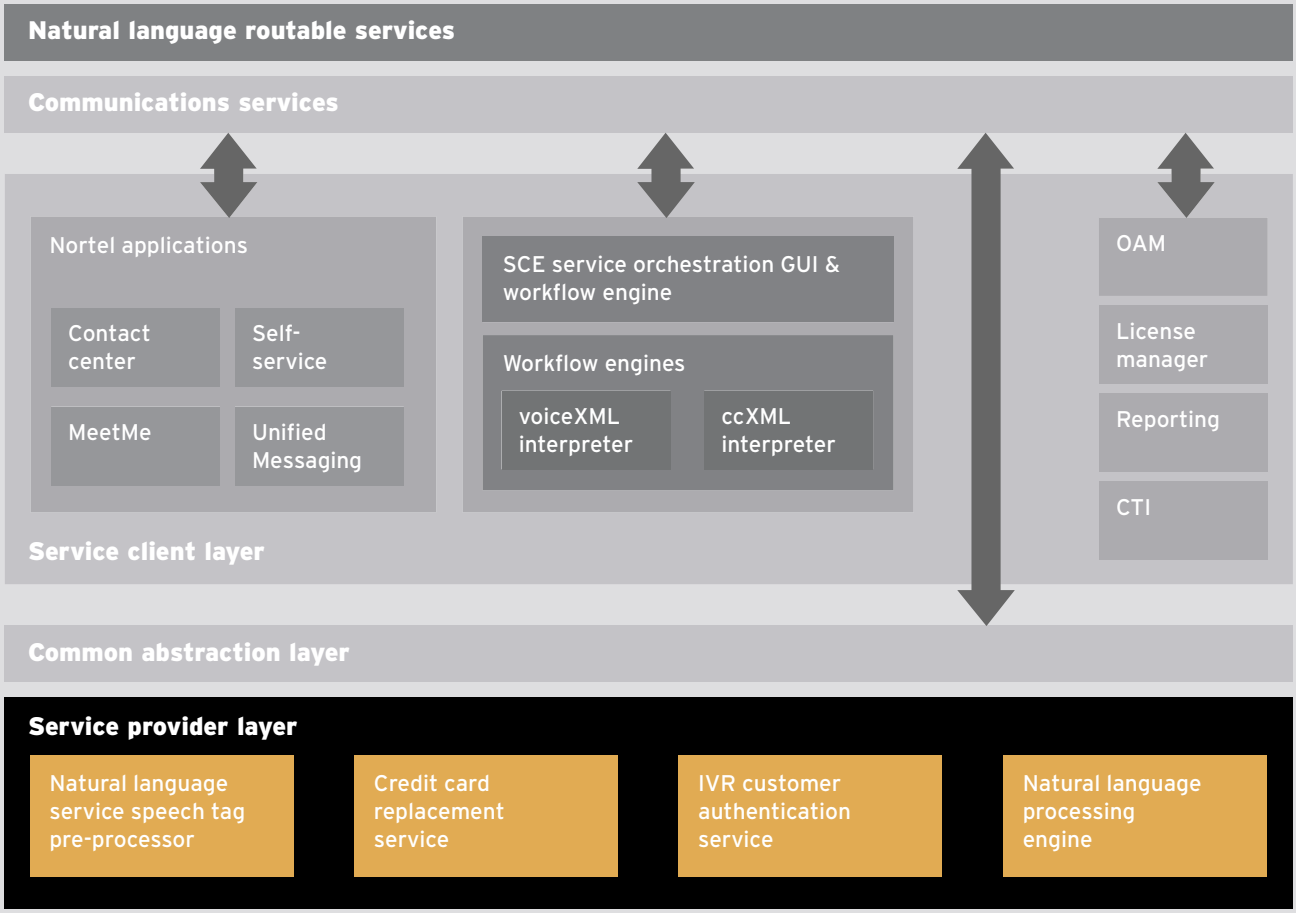
When the customer and seller are both available, the schedule call Web Service notifies the orchestration engine by **associating** with the pre-existing workflow so that the orchestration engine can then proceed to establish the call.

While all these interactions are taking place, the ecosystem is running in the background to ensure that the system is appropriately configured, security requirements are met, and performance/billing-related information is collected.

havior model, traditional applications must become more dynamic. The current generation of applications is too static, inflexible, and cumbersome. Although each application typically provides a scripting capability to permit some level of enterprise adaptation, these scripting capabilities are limited and typically employ proprietary configuration methods.

The current generation of applications, however, contains a very rich set of communications services (such as presence, proximity, and identity management) waiting to be unleashed. To integrate (orchestrate) these communications services into enterprise workflows, our analysis has determined that there are three fundamental interaction methods: trigger, associate, and invoke (Figure 3).

Figure 4. Natural language processing



In this example of a Service-Oriented Architecture (SOA)-based contact center environment, services are invoked through a natural language self-service entry point application that enables callers to respond naturally to an open query, such as "welcome, how can I help you?"

The upper layers in the diagram are the same as those in Figure 1 on page 34, except for the top layer, where natural language routable services are an instance of customer-deployed enterprise services.

When a new service, such as a credit card replacement service, is introduced into the common abstraction layer, it publishes attributes such as speech discovery tags, which are utterances or keywords that a caller would use to access the service. Sample speech discovery tags might include <card>, <lost>, <replacement>, <new>, <billing>, <balance>, and <payment address>.

The natural language service speech tag pre-processor subscribes to speech-enabled services registrations and builds language grammars (a collection of acceptable utterances) based on the speech discovery tags.

At runtime, a new caller who utters a phrase, such as "I lost my wallet and I need a replacement card," will be directed to the credit card replacement service, which will in turn invoke the generic interactive voice response (IVR) customer authentication service and will continue to process the request with a speech-driven user interface. Both services will invoke the natural language processing engine, which accepts speech input and returns text-based data, generally in the form of key/value pairs (such as <card>/<replacement> or <card>/<new>).

Nortel's Media Application Server powers IP multimedia conferencing solutions

by Steve Whynot

The demand for native Internet Protocol (IP) conferencing continues to grow as enterprises converge their networks, infrastructures, and applications. This convergence trend is also driving the creation of new communications-enabled applications built on this IP conferencing infrastructure – applications that can improve real-time communications, business workflows, and group productivity.

Nortel's customer-premise, standards-based, Session Initiation Protocol (SIP) Multimedia Conferencing solutions address this IP conferencing market, and are providing enterprises with a faster return on investment and lower total cost of ownership compared to other hosted conferencing alternatives.

The enabling technology for these solutions is packaged in the Media Application Server (MAS) – an advanced multimedia-processing software platform that provides extensive support for conferencing and interactive voice/video response features. Currently part of Nortel's Multimedia Communication Server portfolio and SIP Contact Center solution, it will also serve as the media server for new applications. The MAS fully capitalizes on Moore's Law, with increases in processor performance resulting in higher channel capacity. Scalability, redundancy, and fault tolerance are achieved by simply stacking and teaming ordinary commercial off-the-shelf (COTS) servers.

Nortel designed the MAS from the ground up specifically to share host-based processors and host-based resources with applications, without restricting the application software or the execution environment. For example, it is common for multimodal applications to be written for and hosted on J2EE or .NET application servers. These servers can be deployed on the same physical hardware and operating system as the MAS, without impacting real-time performance or quality of service. This capability not only reduces deployment costs, but also simplifies application packaging.

The MAS architecture

The MAS is built around a unique architecture designed to meet real-time software constraints while using COTS hardware and operating systems. The MAS architecture separates functional areas of its software framework into distinct components. As a result, real-time components can be safely elevated in processing priority without inadvertently elevating the priority of less delay-sensitive software. In addition, these critical components can be implemented using a more complex asynchronous execution model, which ensures that time-sensitive tasks run to completion within established real-time boundaries.

To improve host-based processing performance, the MAS contains two distinct real-time media processing engines – one designed for conferencing, the other for interactive voice response (IVR) applications. Each engine is optimized for maximum performance by specifically targeting the traffic class, delay tolerance, and application of the media it serves and processes. This dual media-processing architecture avoids design compromises in the areas of scalability and quality of service.

Conferencing performance on the MAS is further optimized by unique mixing algorithms that address two different IP conferencing use cases.

• The standard mixing algorithm addresses traditional conference requirements and is designed to scale from a few channels to thousands. This algorithm employs industry-leading packet loss and timing skew concealment, a dynamic jitter buffer with compaction, a voice activity detector, gain control, and echo suppression. Nortel's Multimedia Conferencing product uses this algorithm to support a wide variety of SIP/PSTN gateways, hard and soft phones, and multimedia devices with varying audio/video codecs, bit rates, network conditions, and conference sizes.

• The conversation-space mixing algorithm is designed and optimized for anchoring a small number of channels for the purpose of applying advanced services. This algorithm streamlines audio- and video-traffic processing, maximizing scalability and minimizing audio/video delay. Nortel's SIP Contact Center product uses this algorithm to efficiently anchor and control customer calls for the duration of their sessions. Its use enables the seamless addition and subtraction of agents, supervisors, and IVR resources without requiring mid-call media stream modifications.

Upcoming releases of the MAS will contain additional enhancements, including support for signaling and media encryption, new audio and video codecs, voiceXML and CallControlXML interpreters, Web Services APIs, and new media control protocols. These technology enhancements will enable native connectivity to Microsoft's Office Communications Server 2007, as well as to other third-party SIP-based networks, and will power the next generation of Nortel's multimedia applications.

Steve Whynot *is Conferencing and MAS Architect, Multimedia Applications.*