Volume 08 Issue 04 Published, November 17, 2004 ISSN 1535-864X

Intel® Technology Journal

Toward The Proactive Enterprise

Successful Application of Service-Oriented Architecture Across the Enterprise and Beyond

A compiled version of all papers from this issue of the Intel Technology Journal can be found at:
http://developer.intel.com/technology/itj/index.htm

Successful Application of Service-Oriented Architecture Across the Enterprise and Beyond

George Brown, Information Services and Technology Group, Intel Corporation
Robert Carpenter, Information Services and Technology Group, Intel Corporation

Index words: SOA, SOE, Business Process Automation, BPM, BAM, SOA, federated enterprise, business process model, value chain, reference model

ABSTRACT

This paper is divided into two major sections. In the first section we look at evolving service-oriented architecture to the service-oriented enterprise, and in the second, we examine integrated process and technology reference models.

The literature abounds with articles about Service-Oriented Architecture (SOA) and particularly Web service-based SOA. Yet, to be successful within a real enterprise, there needs to be more than simple agreement over protocol, data packaging, service invocation, and discovery. While the latter are essential to SOA by providing a foundation for the creation of services, they do not describe how those services function within the overall enterprise environment. They do not describe, for example, how to deliver reliable, scalable enterprise business processes built upon SOA–i.e., the Service-Oriented Enterprise (SOE). The literature on this topic is remarkably sparse.

In the first section of this paper, we explore SOA from the perspective of an SOE where the primary service building blocks execute within a managed SOA eco-system. This managed SOA ecosystem relies upon orchestration, business rules, and process correlation to provide monitoring and management, and business process analytics and autonomics. This framework weaves SOA into the enterprise fabric providing a rich service framework within which to implement enterprise business processes. We begin with an overview of SOE, and then discuss a technology development project to create a Business Rules service. Several usage models are discussed.

In a globally connected economy, enhanced collaboration increases visibility of critical supply and demand changes and streamlines the flow of information that governs critical design and supply chain decisions. The main barrier to digital supply chain collaboration is complexity and high costs associated with extensibility of business processes throughout the network of suppliers and customers. Business process extensibility is the ability of an enterprise to conduct digital collaboration with its partners, by extending processes throughout the supply chain to increase visibility and streamline the information flow.

Enterprises must be enabled to expose their business processes through SOA while retaining and extending the capabilities normally found in tightly integrated vertical applications. We must comprehend fully those collaboration models and the architectural elements necessary to support them.

In the second section of this paper, we explore methods for modeling and implementing business processes within the value chain as an integrated system and introduce a set of principles and guidelines for defining a reference architecture that consistently and accurately represents the value chain as an SOA. We explain how this can be accomplished with a reference model for modeling, simulation, and benchmarking business processes and with a reference architecture that supports deployment and management of collaborative processes within the value chain.

Our research with Collaborative Product Development Associates introduced us to a Federation Enterprise Reference Architecture (FERA*), a reference architecture that extends SOE to the federated enterprise.

* Other brands and names are the property of their respective owners.

GENERAL INTRODUCTION

As enterprises embark on the path to e-business, particularly supply chain management, they are challenged to abandon traditional modes of thinking about their business processes and computing infrastructure and to embrace an entirely new paradigm. Processes that were traditionally scoped entirely within the boundaries of an individual enterprise now participate as elements in a collaborative process within a value chain across a federation of enterprises.

At the same time as enterprises are challenged to cope with this new paradigm, they must also come to grips with fundamental changes in their own computing infrastructure. Most notable among these is the emergence of Service-Oriented Architecture (SOA) and particularly the Web service version of that architecture.

SOA creates problems for an enterprise on two distinct levels. First, the enterprise must come to grips with the complexities of collaborative processes across a federation of enterprises. To do this, it must develop a set of principles and guidelines for defining reference architecture that consistently and accurately represent the value chain as an SOA. This can be accomplished with a conceptual architecture for implementing technology to support modeling, simulation, deployment, and management of collaborative processes within the value chain. This architecture does not specify the detailed semantics of business transactions between the collaborative partners; rather, it distills from the structure of collaboration the key capabilities that must necessarily be present to support them. It becomes the framework for expressing those detailed semantics.

Second, enterprises must come to grips with the fundamental changes in their computing environments. SOA is creating a not-so-quiet revolution in the nature of business computing. Enterprise Resource Planning (ERP) vendors and many others are beginning to incorporate SOA into their products, and enterprises are beginning to incorporate SOA into their computing infrastructure. These directions are positive, but SOA, by itself, provides neither a framework for implementing business processes within an enterprise nor a framework to expose those processes to collaborative participation within a federation of enterprises. These goals can be achieved through a managed SOA ecosystem where process is implemented and managed across the "sea of services."

In the first section of this paper, we first develop a blueprint for the successful application of SOA across the enterprise and beyond. Our approach is decidedly and unabashedly process centric. We begin with a discussion of SOA exploring its limitations. We then propose an architecture to counteract those limitations. This architecture reconstructs the process from the various pieces that execute in discrete process controllers and services and provides a framework for the control of those processes, through an analytics framework and a business rules engine. The result is a flexible, agile, managed SOA ecosystem that supports process analytics and process autonomics across a heterogeneous process control environment within the enterprise. This architecture takes SOA and evolves it to a Service-Oriented Enterprise (SOE) where enterprise processes can be implemented in a consistent and robust way. Second, we discuss the results of two technology development projects that investigate this architecture and present an example that demonstrates the architecture's integrative capabilities. The investigations validate the architecture's value in evolving SOA to SOE.

Third, we return to the complex question of collaborative processes across a federation of enterprises. This inquiry is motivated by the fact that the main barrier to digital supply chain collaboration is the complexity and high costs associated with extensibility of business processes throughout the network of suppliers and customers. Given that the enterprise has evolved to SOE, it still faces a bewildering array of interaction models that have led to "one off" solutions that are neither consistent nor extensible. It is not that the SOE model has failed but rather that its scope needs to be expanded to comprehend collaboration models and business semantics across a federation of enterprises.

What is needed is a set of open source, standardized, integrated frameworks or reference models that would dramatically improve an organization's ability to design, develop, and maintain their business processes. The perceived value of a set of integrated models is far greater than the individual frameworks alone. To explore this proposition, the state of the art is reviewed and then the Federated Enterprise Reference Architecture (FERA) is described in some detail. FERA is proposed as a framework for providing those frameworks and reference models. To validate the potential of FERA, we examine how specific collaboration models are supported by components of the architecture. The results are both positive and promising.

The goal of this paper is to describe an integrated process and technology framework where business processes are enabled both within the enterprise and within the federation of enterprises. At the core of the exposition is the business process, which must be supported by correlation, process analytics; and process autonomics within the enterprise, and must be supported by standardized, integrated frameworks or reference models within the federation of enterprises. Throughout, the SOA paradigm is maintained but more than simple services are exposed; the business processes themselves are exposed within and across enterprises. While the two sections of this paper may seem distinct, they are not. Our goal in these next two sections is to show how the evolution of SOA to SOE is nothing more than the architecture for the enablement of the process itself within the enterprise. The development of integrated frameworks and reference models across the federation of enterprises is nothing more than the architecture for the extension of process beyond the enterprise walls. Together they build an integrated process and technology framework to enable business processes for the e-corporation.

EVOLVING SERVICE-ORIENTED ARCHITECTURE TO THE SERVICE-ORIENTED ENTERPRISE

Our research explored SOA from two different perspectives.

1. In early 2004, we looked at the role of orchestration within SOA firstly from a practical, integrative perspective. The context for that investigation was the belief that orchestration engines, as process automation controllers, will function within a heterogeneous process control environment. This belief is driven by emerging capabilities in applications, ERP and others, and by the limited usability domains of individual product offerings.[1]
2. In the summer of 2004, we looked at the creation of generic services to support an SOE environment by using orchestration itself both as the process controller and as the methodology for exposing the service. A business rules engine was chosen as the object to expose, and several use cases were explored. The goal was to examine how a general-purpose service could both function within an SOE and support the SOE vision.

Interesting as these research activities were, they beg the question of what an SOE is. It is too early to give a definitive answer as the literature using this term has, by no means, reached consensus.[2]

For this reason, we focus on the more modest issue of the service taxonomy and architecture upon which an SOE will necessarily rely. For the purposes of this paper, a working definition of an SOE is as follows:

*A **Service-Oriented Enterprise** is an enterprise that implements and exposes its business processes through an SOA and that provides frameworks for managing its business processes across an SOA landscape.*

At the core of the SOE is the business process that should be elevated to an explicit object within the enterprise. It matters not whether the process in question is contained entirely within the enterprise or whether the process spans firewalls between "enterprises." As virtual enterprises grow in popularity, and as extended collaboration becomes a reality, internal and external processes will necessarily overlap and some issues around security and control will remain unresolved for now.

With process as the focus, it should come as no surprise that the traditional elements of SOA (discovery, protocol, data packaging, and service invocation), while assumed to exist within the computing infrastructure, are not the focus of SOE *per se*. These elements describe how to construct services and how to use services. They do not describe how sets of services support enterprise business processes or how atomic services function within an enterprise.

The central challenge facing the enterprise approaching SOA, and Web service-based SOA in particular, is how to implement business processes within an enterprise in such a way that the process is visible and manageable end-to-end. As the number of services available within the enterprise increase and as the services become increasingly intelligent, their execution pattern becomes increasingly difficult to define *a priori*[3].

The central contention of the SOE architecture is that more than mere inter-service infrastructure SOA is required to support SOE: a managed service ecosystem is required.

A Managed Service-Oriented Environment

Enterprises have come to expect a high degree of integrated capability from existing vertical applications. As a general rule, they provide a solution stack containing the following:

1. Infrastructure monitoring
2. Connectivity
3. Application monitoring and management
4. Application-level work tasks
5. Domain-specific logic
6. Program flow control
7. Presentation

Increasingly, enterprises have come to realize that their business processes are not strictly contained within the domain of a specific vertical business application but, rather, span multiple applications within and across enterprises. This, in turn, raises the question of how to provide the same level of capability for processes when they are no longer executed within a single, integrated vertical application.

At a simplistic level, one could regard the applications as services and "orchestrate" a process as a sequence of steps each of which is contained within a separate application. The sequencing could be done formally by an orchestration engine, by an existing job scheduler, or by a real-time event model. With this approach, one has a functioning process but there are drawbacks. The fact that each application defines and manages its own stack independently creates a complex integration problem to answer the seemingly simple question: what is the state of the process now? More complex questions such as, what is the probability that a given process instance will complete within the prescribed SLA, remain elusive. What is missing in this approach is an overall context for the execution of the "services", i.e., a managed SOA ecosystem.

Solving these problems requires a fresh approach, one that views the process execution framework holistically. Taking this fresh look reveals a huge opportunity for refactoring and harmonization across the application landscape.

Refactoring the Application Landscape

The refactoring process has two phases: first, the elements of the stack need to be mapped to generally available enterprise services; second, additional services and frameworks need to be identified that complete the managed SOA ecosystem and provide the foundation for enterprise process management. At the very least, the first task is relatively straightforward, although the specific implementation within a given enterprise will map to its specific service offerings.

Table 1: Application stack to Service Mapping

Stack Element →	Service
Infrastructure monitoring	Std. Infrastructure monitoring
Connectivity	EAI
Application monitoring and management	Correlation, monitoring and management [BAM]
Application-level work tasks	Individual services
Domain specific logic	Business Rules Engine
Program flow control	Process orchestration and human workflow services
Presentation	Unified delivery, portal

At the heart of this exercise is the substitution of reusable, enterprise services to replace individually coded instantiations of capabilities within vertical applications. It represents the disaggregation of the vertical application into services. Most such services are obviously reusable. Initially, *Application-level work tasks* → *Individual services* may appear single use, but many will, in fact, be found to support reuse. In any event, the identification of such logic into discrete services enhances the potential for refactoring at a later point. The standardization of these services within the overall SOA framework achieves some immediate advantage in simplifying the management of the environment.

In addition to the basic mapping exercise, the following capabilities are essential ingredients of a managed SOA:

1. Process management across autonomous services.
2. Enterprise services to support processes.
3. Process monitoring and management.
4. Process analytics.
5. Process autonomics.
6. Integration of execution layer with high-level business modeling.

Together, this leads to a conceptual architecture:

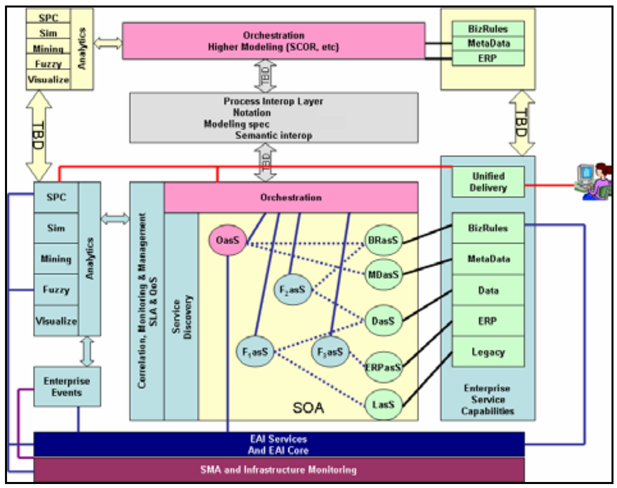

Figure 1: SOE conceptual architecture

The salient features are as follows:

1. Enterprise service capabilities represent (a) legacy applications and data that are wrapped and exposed as services within the basic SOA, [e.g., Data→DasS] and (b) new services, such as business rules. Authentication and authorization would be additional examples.

2. Within the managed SOA domain, services, orchestration, service discovery, correlation, and management provide the bare bones of a managed service environment.

3. The analytics piece relates correlation and management back to the execution environment through business rules to achieve autonomics.

4. Finally, principally through orchestration, business rules and data, higher-level process modeling is tied to the basic execution environment through a process interoperability layer[4].

Together this architecture provides a framework for creating and managing processes within a managed SOA ecosystem.

Correlation: Problem and Opportunity

It is a principle of managed SOA that processes are, where possible, under the control of an orchestration engine. The engine controls the flow of a process and invokes the individual services. For a given process, there may be one or many orchestration engines. These engines, $\{P_k\}$, may be arranged in several models:

1. Single instance invocation
2. Hierarchical
3. Peer-to-Peer. This is often managed by reducing the peer-to-peer model to a quasi-hierarchical hub and spoke model.

In practice, even relatively straightforward processes involve multiple process controllers. This is, in part, due to incorporation of process capabilities into Business-to-Business (B2B), ERP, and Extract, Transform, and Load (ET&L) products. An example will illustrate.

1. A company's manufacturing capacity is in constrained state according to its business rules.
2. A sales order is received over the standard B2Bi environment (B2Bi is P_1).
3. Because it is in constraint, the order is diverted to an analyst for review in a human workflow process (HWS is P_2).
4. The analyst requires a data refresh and cube rebuilds involving a host of data sources. This is managed by a process-based ET&L tool [ET&L is P_3].
5. The analyst concludes the HWP that notifies stakeholders, and executes against the ERP system, which may well have its own processes to execute before placing the order [ERP is P_4].

This is just the input side of the process. It has an external business rules engine in #1 and process controllers $\{P_1, P_2, P_3, P_4\}$ in #2-5. When scenarios like this are extended across enterprises, the number of process controllers will, quite naturally, increase.

Enterprises demand answers to questions such as (1) What is the state of the process now? (2) Where is the order? (3) What are the processing statistics for orders of this type? (4) How do we analyze what has happened and design a better process? (5) How could we introduce SPC or autonomics into order handling? (6) What can we discover about this process?

The problem is that the entire process, $\mathcal{P}$, maps not to a process controller, which could in principle manage flows under its exclusive control, but to a set of them $\{P_k\}$ executing in an order determined not by an *a priori* model, but by logical outcomes based upon the data being processed. Recovering information about $\mathcal{P}$ is difficult, at best.

The solution to the problem has two parts. First, the parts of $\mathcal{P}$ executing in the different $\{P_k\}$ must be associated. This is the correlation problem. Second, information about $\mathcal{P}$ must be extracted from the correlated pieces in the $\{P_k\}$. This will feed the analytics and autonomics engines.

Consider the design pattern *Correlation Identifier*. It has been used to provide a method for associating a response with a request[5]. It is also being adopted and extended in the Web Services Business Process Execution Language[6] [WS-BPEL] where "correlation sets" are defined for similar purposes.[7] For our purposes, tthese standard-use cases, associating a response with a request, are too narrow in scope.

What is needed is an extension of the notion of a correlation identifier[8] or correlation set to meet the needs of process management within a managed SOA. Consider a correlation identifier that has the following properties:

1. It is an extensible xml object which is extended whenever
    a. receipt of process invocation, i.e., $\rightarrow P_k$
    b. optionally during management by P_k
    c. at a process handoff $P_k \rightarrow P_j$
    d. processing ends at P_k
2. Its extensions [leaves] contain
    a. process information, sequence number within a path in $\mathcal{P}$, invoking system, invoked system [optionally tracing the call sequence to underlying services providing the "work" of the process]
    b. process context information, including but not limited to (i) a mapping to physical infrastructure, (ii) process partners, etc.,

and (iii) a key for mapping the activities of $\mathcal{P}$ in P_k against P_k's operational repository.[9]

c. process value adornments, e.g., value of sales order, etc., as name-value pairs.[10]

At each stage of $\mathcal{P}$, C_ID is extended with the state of the process. The next step is to collect the correlation identifiers into an analytics environment. Standard Enterprise Application Integration (EAI) such as Web Services (WS) can mediate the collection. Such a model leads to an architecture as shown in Figure 2.

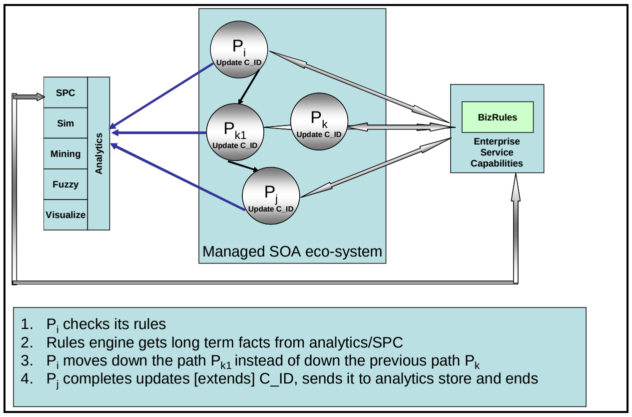

Figure 2: Correlation architecture

Individual process controllers implementing a business process $\mathcal{P}\{P_i, P_k, P_{k1}, P_j\}$ update the analytics environment with the correlation identifier, C_ID, as described above, with basic data and extended data as specified by the enterprise business rules. The first job of the analytics environment is to reconstruct a navigable process from the collection of C_ID. Fortunately, most will be simple linear chains or trees [directed, acyclic nets]. Since at $\rightarrow P_i$, P_i receives C_ID and at $P_i \rightarrow \{P_j, P_k, P_l ..\}$ [11] it passes it on in an extended version of C_ID, the overall reconciliation of the collection of C_ID over a path is not too difficult.

Once the process is reconstructed in the analytics environment, it can be subjected to standard process execution analysis[12], statistical process control, simulation with appropriate tools, data mining, fuzzy logic and heuristics, and process visualization.[13]

With this framework, correlation solves the initial problem of associating the segments of $\mathcal{P}$ executing in the different $\{P_k\}$ and creates the opportunity for a broad spectrum of process analytics opportunities.

Process Autonomics and Business Rules

With the correlation identifier strategy defined, we can now tackle the second part: how to link information about $\mathcal{P}$ to an autonomics framework.

Figure 2 illustrates that framework by providing integration of the managed SOA ecosystem to the analytics environment through a business rules engine.[14] By designing processes to be rules dependent, the control loop can be completed to enable autonomics.

Most rules engines evaluate rules against a set of facts and those facts come in two forms: instance facts are submitted to the rules engine when the rule is to be evaluated, and environmental facts are longer term and persistent within the rules environment. The integration occurs by taking the results of the analytics, (e.g., SPC computations, infrastructure alerts, etc.), and propagating them to the rules engine as events over the standard EAI bus. The rules engine stores them as environmental facts. Then, when a process executes, it submits its instance facts to the rules engine where, based upon the logic of the rules, they are combined with environmental facts to render a decision back to the process. Control can be exercised over the process through the analytics framework and through the rules engine. A simple example based upon infrastructure alerting will illustrate the principle.

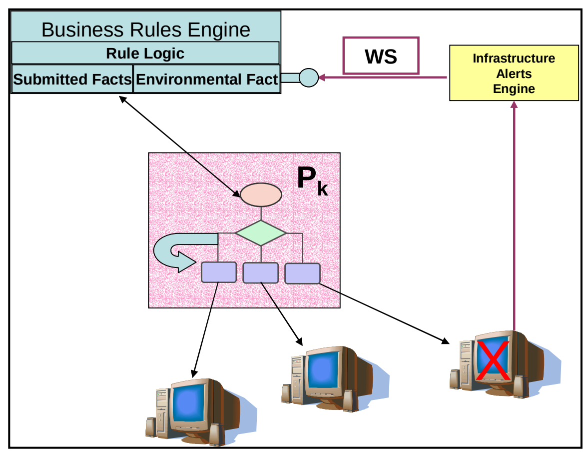

Figure 3: Infrastructure rules integration model

In this model, infrastructure events are sent to the infrastructure events engine (operations console) where an alert is sent via Web services to the rules engine. When a service goes down, the rules engine receives the event and stores it together with the expected time to availability as an environmental fact. When a process, $\mathcal{P}$, starts under control of P_k, P_k calls the rules engine to determine if the infrastructure is ready. The submitted process is checked against its dependencies and the non-availability stored in the environmental fact base. The rule computes the likelihood of completion within the SLA based upon the estimated availability time. P_k is instructed to proceed, resubmit in x minutes, or abort as appropriate. Additional analytics-based and infrastructure-based process control scenarios are easily developed within this architecture.[15]

In the technology development activities ending in the summer of 2004, we evaluated a business rules engine in this architecture against the following scenarios:

1. Provide switches in other processes.
2. Provide thresholds in other processes.
3. Alternative source parameterization.
4. Add content to processes based upon business domain logic.
5. Demonstrate rules as an integration point.
6. Demonstrate rules-based reflective programming [variant of #4].
7. Demonstrate forward chaining for complex automation.

Each of these allowed us to provide a level of autonomic process control from a rules environment and allowed us to explore different sourcing options for the environmental facts. An additional benefit was that with rules-based processes, the process could be kept more generic driving the potential for reuse. When measured against the critical success indicators of the project,[16] the rules-based integration approach proved successful. The framework provided the richness necessary to move SOA toward SOE.

Challenges and Solutions

The challenge to move SOA toward SOE is met in large part by the architecture described here. The service taxonomy and architecture provide a framework that enables enterprises to expose their business processes through SOA while retaining and extending the capabilities normally found in tightly integrated vertical applications.

The principal challenge will be to evangelize the correlation identifier as used in this paper and to standardize its elements. While the architecture is not strictly dependent upon it, there will need to be efforts in the supplier ecosystem to drive acceptance of the model and to begin to develop an open standard for implementation. The advantages of doing so are enormous for the company seeking to move past SOA to SOE.

An additional challenge is to spur work on the process interoperability layer to define the standards and mechanisms needed to create transparency between models describing collaboration scenarios within a federation of enterprises, and models executing processes within an enterprise under this architecture.[17]

To enable enterprises to expose their business processes through SOA while retaining and extending the capabilities normally found in tightly integrated vertical applications, more than mere inter-service infrastructure SOA is required: a managed service ecosystem is required. This ecosystem leads to a natural division: enterprise services, managed SOA environment, analytics framework, and integration to higher-order modeling.

The challenge of the heterogeneity of the process control environment in turn drives the necessity of correlating the process segments of $\wp$ executing in disparate controllers { P_j , P_k , P_l ..}. This is resolved through the correlation identifier, C_ID, extended by each P_k and managed in the analytics environment. C_ID provides basic process execution data, processes context and values for analysis, and provides the pointers back to the operational repositories of the $P_{k,}$ which allow for detailed tracing and trouble shooting.

The analytics environment together with the infrastructure management environment then provide input to the rules engine that serves as the integration point to the execution environment. This completes the autonomics control loop. A basic framework for exposing processes is achieved and the first steps from SOA to SOE are realized.

The significance of this research and technology development is that it solves the key challenge posed by emerging product capabilities. It provides a product independent framework for the correlation of processes across a heterogeneous process environment. The correlation identifier together with business rules then provides the foundation of process analytics and process autonomics moving the enterprise from the "sea of services" of an SOA architecture to a managed SOA ecosystem. The enterprise moves from SOA to SOE.

Equally importantly, this architecture creates a managed process execution framework for the enterprise that enables it to participate within a federation of enterprises precisely because the execution of externally triggered processes are managed and visible end-to-end within the enterprise. It provides the control necessary to guarantee reliable participation in collaboration scenarios across a federation of enterprises. The next step is to comprehend fully those interaction models and the architectural elements necessary to support them. The next section does just that.

INTEGRATED PROCESS AND TECHNOLOGY FRAMEWORK

In a globally connected economy, enhanced collaboration increases visibility of critical supply and demand changes and streamlines the flow of information that governs critical design and supply chain decisions. As a result, companies can better coordinate their internal product development plans and their design and engineering effort, and they can bring more new products to market faster while improving on-time delivery and reducing inventories and costs. However, information technology solutions currently used to support supply chain collaboration have at best only scratched the surface of the total opportunity.

The main barrier to digital supply chain collaboration is complexity and high costs associated with extensibility of business processes throughout the network of suppliers and customers. Business process extensibility is the ability of an enterprise to conduct digital collaboration with its partners, by extending processes throughout the supply chain to increase visibility and streamline the information flow.

Global economic pressures mandate that manufacturing companies find ways of faster and more intelligent collaboration throughout the supply chain. And while there are processes that can be integrated using relatively simple forms of collaboration, the vast majority of the most critical value-added collaborative processes are non-linear, non-deterministic, and multi-disciplinary in nature.

Integrated business process management and information technology architecture is critically important in the increasingly complex global operations. The standardized business process is a critically important factor for sustainable differentiation in global business as it provides baseline metrics for continuous improvement in understandable terms, and it contributes to the formulation of consensus across the participating companies in the value chain. An integrated approach is to utilize process reference models along with reference architectures representing a variety of technology deployment components serving loosely coupled collaborative processes that can be reconciled through direct mapping from the process models, defining business semantics in plain, non-technical language.

Such a reference architecture, described here, is based on the federated approach, where loosely coupled federated systems connect to each other through gateways based on common business semantics and open standards. This approach utilizes Web services for system integration and workflow orchestration and open standard business semantics for describing the federation and its working principles. Without predetermining and imposing any specific workflow and without entailing any additional investment on the part of the participants, this approach can penetrate as many tiers of the supply chain as required.

Service-oriented architectures generally presume a collection of services that enable the business much as pre-SOA infrastructure has. A case in point is the Service Reference Model for FEA, the Federal Enterprise Architecture as shown in Figure 4.

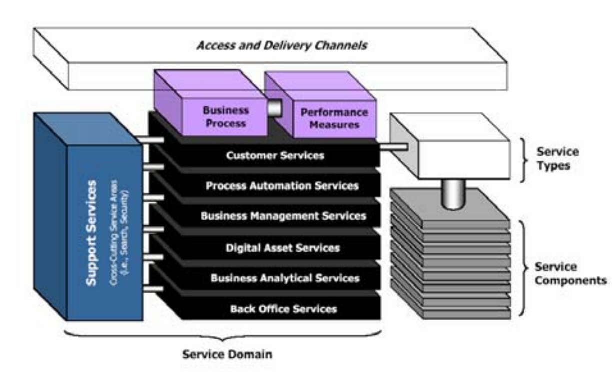

Figure 4: Service Reference Model for FEA

What is missing from these generic architectural patterns is any recognition of how business semantics determines the services that are defined. Moreover, there are no patterns to define how those service-based processes are instantiated. Lacking such patterns diminishes the distinct advantages of the well-formed architecture for the SOE as discussed earlier.

FERA[18] has potential to be the collaboration framework in a future state architecture that enables enterprises to expose their business processes across a federation of enterprises through SOA. To validate the potential of FERA, we examine how specific collaboration models are supported by components of the reference architecture. We look at how certain collaborations in process scenarios from different communities of businesses found in our value chain are enabled by FERA.

Defining of Business Semantics

It is a well-documented fact that business and IT managers rank business process improvement as their number one priority. Companies that are process-focused have well-defined processes that are both vertically and horizontally aligned, are well-governed, produce costeffective and reliable outcomes, and are able to adapt more quickly to market changes.

The corporate focus now shifts away from the speedy transformation of materials through the supply chains to understanding what drives product value chains. In turn, the fundamental organizational model may shift to promote better transparency and synchronization of the extended value chain. For example, centralized management may split into federated business units to reduce the scope in terms of the breadth of product coverage in each unit, and to tie product metrics more tightly to such criteria as time to market, acceptance, and profitability. For another, outsourcing of product design and manufacturing may not only enhance focus, but may also share the responsibility across a larger resource pool. Even the delivery model may shift from an emphasis on product manufacturing to service or solutions. In the process, corporations look for control and balance of the three major business payoffs that arise from total product value-chain management: time to market, market acceptance, and profitability.

Intel's IT research agenda in Business Process Systems has focused on the use of reference models to represent process scenarios from different communities of business with the intent of generating reusable templates that can accelerate the integration of trading partner business systems/processes into the supply network. This work started with application of the Supply Chain Operations Reference (SCOR) model and expanded to successful application of enterprise-wide reference models. In order to consolidate these business process reference models into a usable value chain reference model, it became clear that the reference models must be coupled with a collaboration framework that facilitates implementation of process scenarios across the value chain.

A set of open source, standardized, integrated frameworks or reference models would dramatically improve an organization's ability to design, develop, and maintain their business processes, and the perceived value of a set of integrated models is far greater than the individual frameworks alone. A general consensus has developed among partners developing essential collaboration models for product design for supply chain that the long-term value proposition is to focus on a Value Chain Operations Reference model (VCOR).[19]

Defining business semantics in terms of the common vocabulary of VCOR aggregates business applications and business processes to a higher level of abstraction. In this way, value chain integration enables coordination across departmental, organizational, and enterprise boundaries from an overall business-level perspective.

The benefit is that it facilitates service-composed processes and, thereby, brings service-oriented relevance to a complex IT landscape in which ongoing, flexible adaptation is necessary.

Figure 5: Value Chain Operations Reference model, VCOR

Mapping of Business Process to Core Collaboration Capabilities

To classify information exchange patterns that can take place in a federated enterprise framework, one has to consider two dominant characteristics: how is information shared and how is the business context of collaboration administered and preserved. Because the information content and context are decoupled, a combination of the two process characteristics determines classes of collaborative patterns in a federated enterprise.

Information sharing has three patterns: people to people, systems to systems, and people to systems (and vice versa). In people to people collaboration obviously information gets exchanged using a vocabulary and semantics that enable all parties to properly interpret the content and context of collaboration. In systems to systems, information exchange semantics are left to systems to interpret and process. Business context administration has two patterns: centralized and distributed. Where a centralized pattern exists, a common business logic supports the entire scope of collaboration; where a distributed pattern exists, a separate business logic gets involved, thus some shared business semantics need to reconcile the context of collaboration between semantics specific to each participating domain.

There has been considerable work done in defining and implementing e-Workforce solutions to facilitate synchronous and asynchronous collaboration in distributed environments for Classes 1, 2, and 3 as shown in Table 2.

Table 2: Synchronous and asynchronous collaboration in distributed environments

	People-to-People	People-to-Systems	Systems-to-Systems
Centralized (single authority)	Meeting	FERA Class 1: Bulletin boards and web meetings	Tight integration
Distributed (multiple authorities)	FERA Class 2: Personal interaction supported by collaborative software	FERA Class 4: Collaborative business process management	FERA Class 3: Publish and subscribe system-to-system

High-level collaboration service architectures exist that provide support for generic collaboration semantics through core collaboration business process services built on the collaboration services stack. For the pattern of collaborative business process management, Class 4, there is a need to bridge business processes with the core collaboration capabilities.

A significant lesson from this work is that business can be represented by business processes defined in terms of value chain reference models (e.g., VCOR), which can be used for process modeling, gap analysis, simulation, benchmarking, and consensus building. Secondly, an architectural representation can be used that maps process models to components of the conceptual architecture and resources used for accurate, fast, and flexible implementations of the process models in a federation. The two independent but reconciled process representations facilitate the mapping of business processes to core collaboration capabilities.

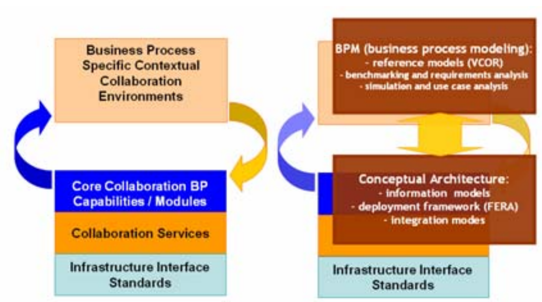

Figure 6: Integrated process and technology framework

Basic Components of FERA

FERA is a generic system architecture that describes all classes of deployment of the federated enterprise approach. FERA categorizes types of collaborative processes and patterns of technology deployment used to support those processes.

Being a common abstraction model representing a variety of technology deployment patterns of loosely coupled collaborative processes, FERA can be mapped into equivalent process models representing the same processes by using business semantics. Thus, in FERA an integrated process and technology framework can be established to speed up assessment of gaps, requirements, and deployment of the solutions used for supply chain collaboration.

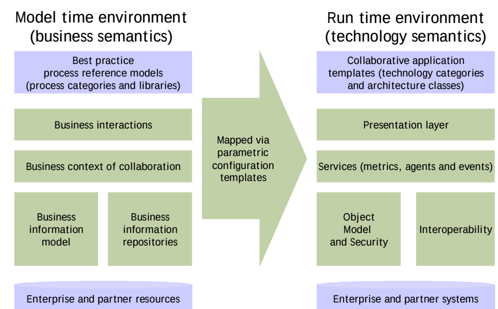

Figure 7: The usual representations of semantics

The two representations in Figure 7 are different views of the same process and need a mapping from the level of business semantics to that of technical infrastructure.

On the business semantics level, interactions between resources, their context of collaboration, information model, and repositories of information are all defined using business process modeling. This enables development of different models and their evaluation in a language that can speed up consensus building and provide for benchmarking and common representation of best practice reference models.

On the technical infrastructure level, the figure shows how technology components are used to deploy and support the processes defined in business process modeling semantics. A clear distinction must be made between the two sets of semantics. This distinction allows for an evaluation of the alternatives that can be used to support the same process, even if the deployment patterns need to be different in order to accommodate different levels of readiness of participants.

In general, FERA requires that participants can freely browse business models, evaluate them, and assess their own usage scenarios independently of each other, yet understand the dependencies between the collaborating parties. After that, different participants can elect the deployment model that corresponds to the business process they decided to participate in, supporting it with their own capabilities and system-level readiness. [20]

There are seven basic components of FERA, and a specific configuration of these components can be used to support different classes of processes, where certain deployment patterns can support more processes and vice versa: one process can be supported by a specific combination of different deployment patterns.

Figure 8 presents the basic components of FERA and we explain each component in the list that follows.

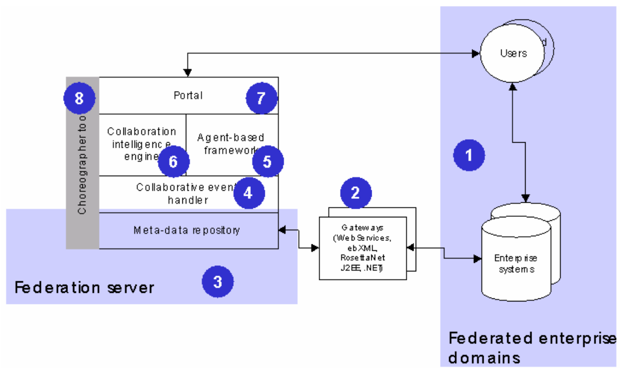

Figure 8: Basic FERA components

1. *Federated enterprise domains:* These are all systems that host the business logic, functions, and data for specialists and all users that need to collaborate.
2. *Gateways:* These components expose data contained in proprietary formats and local databases of the federated systems and users, and they establish open formats for data exchanges with federation servers. In FERA, gateways and federated systems have to conform to the set of standards commonly called SOA.
3. *Federation servers:* These open meta-schema servers publish and subscribe to and from gateways and portals. The servers support the persistence of collaborative sessions and preserve the context of the collaborative workflow. They also manage participants' profiles, and control access to data, security, and sign-on.
4. *Collaborative event handlers:* These rule-based engines control the start and finish and maintain the continuity of collaborative sessions, and they manage the status and change control of the collaborative workflow.
5. *Agent-based frameworks:* The component agents enable automation, pre-processing, and post-processing of data exchange steps in the collaborative workflow that is controlled by event handlers and maintained by federation servers.
6. *Collaboration intelligence engines*: These data collection, analysis, and reporting components provide insights into the effectiveness of collaborative processes to all participants.
7. *Portals*: These components enable Web-based user interfaces for participants to link directly to sessions on the federated servers.
8. *Choreographer tools*: These tools, used by federated instance administrators, are administrative tools that manage profiles, security, access, and sign-on protocols, and they link to federated system gateways and configurations of the meta-schema for the federated server. They are also used for choreography of business scenarios administered through contexts of collaboration and their version control.

Because of the autonomy of the enterprise domains, each needs an interoperability gateway. This gateway in FERA needs to be based on an open standard that will enable a reliable and scalable communications interface with the Federation Server in both directions. It should specify common functions that hide standard-specific terminology, constructs, and entities that can confuse end-users who do not need to know low-level standard specific details. Collaborative contexts are defined, preserved, and made persistent on one or more interconnected federation servers.

If the Federation Server comes with a registry-repository (reg/rep) component, the participants must register themselves in the Registry and Repository section, which governs security, access rights, and some other basic profile maintenance-related requirements.

The FERA built-in collaborative services–Collaborative Event Handler, Collaborative Metrics, Intelligence Engine, and an Agent Framework–enable collaborative process coordination and configuration. Agents, events, and metrics lookups are managed as built-in services. These three components are the heart of collaborative process choreography and enable complex collaborative patterns to take place in a non-deterministic business process design. Internal communication and collaboration between the Federation Server and the built-in services are conducted by Web services. This is a typical SOA.

Agents, metrics and events enable complex collaboration processes. Each FERA participant has its own created and configured agent. Agents interpret business process collaboration semantics that should be modeled and specified using standard business process modeling specification(s). The specification candidates for business collaborations (private and public) are ebXML Business Process Specification Schema (BPSS), Business Process Execution Language for Web Services (BPEL4WS), Business Process Modeling Language (BPML), etc. Although none of these has reached the level of sophistication required by FERA Class 4 process (discussed later), alternative approaches based on UML semantics have yielded sufficient results in many applications.

The Collaborative Intelligence Engine logs the progress of collaborations and collects the information used for detailed collaboration analysis (collaboration paths, overall system responses, system errors, etc. [21]).

The Portal enables user access to the Federation Server, contains user-defined views, filters and macros, enables invoking of services from a Web browser, connects other applications that are not announced in the federation into a coherent environment for data viewing and exchange (e-mail, Web meetings, etc.).

FERA Process Patterns

FERA describes a variety of collaborative solutions used in practice to support complex processes spanning organizational domains, business entities, and information technology systems that are based on three common principles: *autonomy, partiality and independence; open content and context semantics; and continuous operation.*

- *Autonomy, partiality, and independence.* Each participant in the collaborative process retains full autonomy of the internal workflow and ownership of her data. FERA does not require any change in local business logic or processes. FERA does not prescribe the collaborative process; the process is self-discoverable by all participants who can elect to join a subset of capabilities that fits their business needs and/or system readiness. Any participant can enter or exit the federation at any time; the processes supported by FERA will be able to continue.
- *Open content and context semantics*. To connect all participants and to maintain persistence of the collaborative workflow state, FERA requires that the federation and its working principles be defined using an open standard semantics that is equally accessible to all participants. A two-part description is required by FERA: context and content, and they both need to utilize an open standard semantics (e.g., for context BPEL, UMM, for content ISO 10303, PDX, etc.)
- *Continuous operation*. FERA does not need to move the data between participating systems in order to support the collaboration process. It only needs to enable information sharing through common definitions and/or references to the data changes supported by context and content definitions at the Federation Server, while preserving the context within which the information is being shared. FERA needs to support a variety of configurations of the same process to be continuously changed and executed within the same context thus following preferences, internal constraints, and capabilities of each participant.

These principles when translated to technology requirements mean that four basic requirements have to be met by all seven FERA components:

- *Service-based integration*. APIs between FERA components internally as well as communications with the external computing resources need to be supported by Web services. This enables a consistent method of searching, finding, locating, configuring, and executing all FERA APIs.
- *Dynamic configuration without coding*. FERA patterns of collaboration are generic, described using open standard semantics and therefore reusable and configurable to support a wide variety of processes on any FERA-compliant platform. FERA components are described using configurable templates that do not require coding, so a rule-based mapping between business process models (e.g., business modeling semantics like UML) to FERA deployment semantics can be achieved. FERA patterns can assist in classifying and declaring collaborative capabilities of different participants, but cannot prescribe the process in advance.
- *Reusability of components and patterns*. The context of collaboration needs to be defined using open standard semantics, while the ontology of contents needs to be consistent across all shared meta-data definitions. This enables the processes to be configured as they fit the business needs and reconfigured when the needs change. The process choreography itself evolves as the process is being executed by utilizing a combination of events, metrics, and agent services.
- *Meta-schema decoupled from business logic*. In FERA processes, context and content are modeled

separately and cannot constrain each other through direct relationship dependencies. This effectively means that relationships between entities defining the content of collaboration, as well as relationships between entities defining the context of collaboration, need to be declared in their respective meta-schemas as objects too.

SUMMARY

Closing the gap between process and technology resolves one major hurdle: the overwhelming complexity of the processes themselves in optimizing collaboration across the design and supply chain. Processes present far more intricacies than most companies recognize. All too often, IT organizations apply simplifying assumptions regarding requirements that inevitably lead to the wrong technical implementation, because ultimately, far broader ranges of interactions need to be supported. Either the rigidity only serves processes too trivial to matter, or else the lack of flexibility drives out the intelligence of effective decision making. Extreme care must be applied in considering the full spectrum of process interactions to understand their character completely before implementing technology, since that will determine the deployment framework.

This paper explored requirements of an SOE that executes within a managed SOA ecosystem, relying upon orchestration, business rules, and process correlation to provide monitoring and management, and business process analytics and autonomics. A framework was presented that weaves SOA into the enterprise fabric providing a rich service framework within which to implement enterprise business processes.

The SOA has great potential to enable more rapid deployment of service-composed processes but a gap may arise because technology does not speak a process language. Service descriptions do not communicate clearly to people regarding their roles, responsibilities, priorities, and milestones. Organizations tend to normalize processes and apply narrow definitions to activities involving broader requirements and deeper complexity. Process variations across the value chain break the systems that are based on those normalization efforts.

Value chain processes must be mapped to a well-defined service framework, based on common standards, which enable the sharing of content-rich information in a timely manner while maintaining an accurate business context. In order to establish faster and more intelligent collaboration throughout the value chain, a process framework must be explicitly defined in order to be integrated into the extended enterprise based on common objectives, on clearly establishing expectations and roles for all individuals, and on setting a reference baseline for continuous improvement.

By fully comprehending those collaborative processes within the value chain and the mapping to a managed SOA ecosystem comprising SOE, the enterprise is enabled to participate in a federation of enterprises, through the managed process execution framework, with a correlation of processes, process analytics, and process autonomics based on meaningful business semantics across a federation.

ACKNOWLEDGMENTS

We thank Vasco Drecun, partner and PLM research director at Collaborative Product Development Associates, LLC. Vasco is a strategic partner in the research on FERA and a contributor to that section of the paper.

ENDNOTES AND REFERENCES

---

[1] A secondary component of the investigation was to analyze the adequacy of BPEL as a language, where its well-understood limitations with regard to advanced synchronization scenarios were documented. BPEL constructs were developed to fill the gaps, but questions remained about both the language itself and the direction of the governing body, at least to the extent that BPEL is intended to support extended collaboration scenarios. On the vendor side, the inchoate state of thinking was confirmed through a series of meetings leading to the conclusion that further investigations remain necessary. Refer to Bartosz Kiepuszewski, "Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows," *Ph.D. Dissertation* 2002, pp. 62-91. See also Petia Wohed, Wil M.P. van der Aalst, Marlon Dumas, Arthur H.M. ter Hofstede, "Pattern Based Analysis of BPEL4WS," *Technical Report* FIT-TR-2002-04, QUT. See also Petia Wohed, Wil M.P. van der Aalst, Marlon Dumas, Arthur H.M. ter Hofstede, "Pattern Based Analysis of BPML (and WSCI)," *FIT Technical Report,* FIT-TR-2002-05.

[2] See for example, "Envisioning the Service-Oriented Enterprise," *Zapthink*, 2003. See also, Sleeper, "Towards the Service-Oriented Enterprise," "service-oriented enterprise reflects a change in the human, business process, and organizational governance factors that shape how IT interacts with the business" at

http://www.webpronews.com/enterprise/enterpriseonline/wpn-13-20030903TowardstheServiceOrientedEnterprise.html*

An additional aspect of SOE is Service-Oriented Infrastructure [SOI], typically seen in terms of GRID and the virtualization of the infrastructure layer. The relation between SOI and SOE is a topic of current research.

[3] With n services in the enterprise, there are n*(n-1) potential two-way interactions. Tracking the interactions through realistic business processes may produce graphs that are neither linear nor acyclic. Services without an execution framework quickly become unmanageable. This problem is made even more complex for dynamic processes where the process execution path is data dependent and not static.

[4] The process interoperability layer provides the bridge between the execution environment described in this section and the important business semantics and interaction models that are the focus of the second section of this paper. The two parts work hand in hand to provide a top-to-bottom blueprint for the SOE applicable to an enterprise and to a federation of enterprises.

[5] See for example, Gregor Hohpe and Bobby Woolf, *Enterprise Integration Patterns*, Addison-Wesley, Boston, MA, p. 163 (2004).

[6] See Working Draft 01, 08 September 2004 at http://www.oasis-open.org/committees/download.php/9094/wsbpel-specification-draft-Sept-08-2004.html#s.Extensions.sect4*

[7] op cit §10, 10.1 and 10.2. See also §14.4 [Extensions–Correlation].

[8] The correlation identifier will be denoted C_ID.

[9] This provides a method to explore the run-time repositories of the individual P_k relative to a specific process instance $\mathcal{P}$.

[10] Suitably defined, the "value adornments" become the backbone of business activity monitoring within the framework.

[11] Even within a tree, a process controller P_i may call more than one successor.

[12] For example, if time stamp data are collected along the way, process QoS and SLA measurements can be taken, trended, and comprehended.

[13] In the second section of this paper, we explore a Federated Enterprise Reference Architecture in the context of an Integrated Process and Technology Framework. From this section's point of view the federation server, while possessing vital capabilities for managing business semantics and collaboration context, embodies another process controller, P_k. It can participate in correlation, based upon a unique process ID, at the moment of service invocation, resulting in a rich integrated analytics capability across the entire stack.

[14] The general benefits of a business rules engine *vis-a vis* traditional application-level coded logic are that business rules (1) change quickly; (2) promote agility; (3) enable the business people to control/change business processes; (4) reduce tight coupling between business behavior and code base; (5) leave the released code base stable yet enable dynamic environment; and (6) as proved by the research here, can form the basis of SPC and reflective programming control over the execution environment.

[15] An interesting example is created by use of automated data collection, such as RFID, feeding an operational or analytics environment that, in turn, sets environmental facts in the rules base via events. Transactional or aggregated information can then drive logistics processes in real or near real time based upon a full set of business rules and their associated inferential patterns.

[16] The critical success indicators of the project related to the business rules engine itself were as follows: modify business behavior without modifying the underlying code base; move business domain logic out of code to minimize impact to the release to production process; drive agility by giving business owners control over business processes; support globalization by explicitly partitioning business rules from code; drive code reuse by reducing dependence on domain specific logic in code; make business rules visible and consistently documented; and provide a framework for versioning and control of business rules.

[17] In the TD effort, higher-level process models describing collaboration scenarios were exported into BPML and then translated by hand into BPEL for execution. The process worked but an automated solution assuring semantic integrity is needed.

[18] FERA was introduced by Collaborative Product Development Associates, LLC, through sponsored research on product lifecycle management and assessing collaboration processes for product design for supply chains developed with enterprise reference models. FERA represents a benchmark for establishing the level of maturity in supporting collaboration within a federation.

[19] The Value Chain Operations Reference (VCOR) model is being developed by the Value Chain Group, an independent consortium dedicated to the consistent

integration of constituent frameworks to form a common language for value chain optimization.

[20] The first part of this article, "Evolving SOA to SOE," describes the instantiation of a rich service framework to support FERA within an enterprise domain.

[21] Supporting unique process IDs at the level of FERA through the execution environment, such as that described in the first section of this paper, would enable correlation between enterprise systems and provide extended analytics across the entire federation.

AUTHORS' BIOGRAPHIES

George W. Brown joined Intel in 1994 and is currently a senior program manager within the ISTG Research group. He focues specifically on methods and tools to ensure Intel reaches its goals in supply chain management by identifying opportunities to apply information technology in innovative ways to solve business problems and improve Intel business processes. He is also the past chairman of the Supply Chain Council and has represented Intel in external research and benchmarking activities as chair of the SCC Research Stategy Committee. His e-mail is george.w.brown at intel.com.

Robert E. Carpenter joined Intel in 1997 after a 20-year career as an attorney, retiring as district attorney of Greene County, NY in 1996. For many years, he taught mathematics at Bard College and later law at Albany Law School. After joining Intel, he focused on the development of architectural frameworks and most recently has been responsible for the definition of process automation architecture across the business landscape. His current interests include process algebras and the application of formal representation theory to virtualized process and infrastructure. His e-mail is robert.e.carpenter at intel.com.

THIS PAGE INTENTIONALLY LEFT BLANK

For further information visit:

developer.intel.com/technology/itj/index.htm