Toward Business Transaction Management in Smart Service Networks

Willem-Jan van den Heuvel and Mike P. Papazoglou • European Research Institute in Service Science, Tilburg University

This article proposes a novel business transaction model for smarter service networks, which will leverage pooling real-time information from software services and sensor networks to more effectively manage fine-grained tenets of service based applications, such as critical business data, local and aggregated quality of service, and associated key performance indicators (KPIs).

The face of the global digital economy is rapidly becoming a global, services-centric economy that realizes economies of scale through networked enterprises transacting and cocreating value on digital infrastructures with a global reach. The global digital economy is fuelled by new distributed computing technologies, often provided via the Internet, that enable cheap provisioning, scalability, and seamless connection to anyone, anywhere, impacting how we live our daily lives, conduct business, and shape our societies.

Services economies’ explosive growth, coupled with the evolution of powerful, digital communication networks – which we tend to associate with the Internet – help transform service companies and their clients alike from regional businesses to globally integrated, value-creating service networks. In particular, service networks are open, complex, and fluid socioeconomic systems of organizations and processes that break away from classical knowledge and power hierarchies to accommodate the coproduction of new knowledge and services through organic peer-to-peer interactions.

Resources in service systems can include people, software systems, computing devices and infrastructures, organizations, and shared information, such as business rules, regulations, measures, and methods. Resources are connected internally and externally through value propositions. Generally, service systems cocreate value for three levels of society: individuals, such as employees, consumers, and public officials; enterprises, such as businesses and supply chains, public entities, or user communities; and national, continental, and global populations.

Smart Service Networks

Leveraging service networks with novel distributed computing, such as sensor network technologies, gives rise to the concept of smart service networks (SSNs). These networks are designed to increase value for their clients as well as partners participating in the network; they interweave smart physical devices, tagged objects, and sensors with software services that live on the Internet. We might argue that, in this way, SSNs support value-adding, end-to-end processes that can monitor their own performance by sensing and interacting with the physical world, and repair, upgrade, or replace themselves proactively.

For example, Wal-Mart has adopted the SSN concept to co-invent new, more profitable, and greener ways to do business, optimizing resources and exploiting knowledge from thousands of its worldwide suppliers and third-party logistics companies pooling not only real-time information from Internet services but also from sensors, mobile devices, and RFID tags. This lets Wal-Mart significantly reduce its environment-
tal footprint and address issues such as fishery depletion, climate change, and pollution.

Figure 1 illustrates the concept of SSNs as we envision it and shows end-to-end processes that might be connected on a global scale. These processes rely on services that providers deliver and clients consume. Both clients and providers are connected with the service network through their own service-based applications (SBAs). Figure 1 shows that, for instance, an SSN for the car-manufacturing industry could comprise globally available, end-to-end processes managing customer orders, billing, payment collection, purchasing, and inventory management. In particular, it displays a business transaction — labeled “purchasing” — defined over two end-to-end processes (“order management” and “inventory management”) that are internal to the network. The business transaction stipulates and enforces the conditions under which a service client (such as a car manufacturer) can acquire products (for instance, spare parts) from a supplier and specifies service providers’ responsibilities in terms of pricing, part delivery time, transportation costs, and so on.

Figure 1 also illustrates another important characteristic of smart service systems: they’re layered on an IT infrastructure — typically, a cloud — that enables network partners to seamlessly integrate their software services with back-end systems and wireless sensor networks. In this way, virtual software services can collect information about the physical environment. For example, the inventory services can receive real-time notification once a ship with RFID-tagged containers has arrived, personnel have unloaded those containers, and trucks instrumented with wireless sensors are approaching the warehouse.

**Business Transactions**

We claim that transacting services in SSNs demand unique approaches with regard to how service engineers conjures up, develops, and manages business transactions. In an SSN environment, *business-aware transactions*, or business transactions, are fluid and complex, involve multiple parties spanning many organizations, and can have a long duration. Particularly, we might best define them as automated, long-running propositions involving negotiations, commitments, contracts, shipping and logistics, tracking, varied payment instruments, and exception handling.

Currently, SBAs concentrate on composing software services into processes relying on standards such as the Business Process Execution Language (BPEL) and WS-Choreography and then defining traditional long- or short-running transactions on top of them using standards such as WS-Transactions. However, SBAs can’t explicitly correlate critical business activities and events, quality-of-service (QoS) requirements, and application-significant business data — such as delivery dates, shipment deadlines, and pricing — in one process with related activities, events, QoS, and business data in other processes in an end-to-end process constellation. This implies that critical business information and procedures are deeply buried in SBA implementation code, which severely hinders maintenance and adaptation, both of which are essential for SSNs. Such hardwiring means that any change or update to the application-management logic already fabricated within an application requires programmatic changes to the SBA itself. This renders the potential reuse, customization, and monitoring of application management capabilities impossible. It also introduces intrinsic discontinuities between end-to-end processes because information flows shared between SBAs might be disrupted. For instance, a possible decoupling of payment information (in payment and invoicing processes) from ordering and delivery information about goods and services (in order management and shipment processes) increases risks, could vio-
late data integrity and contractual agreements, and might introduce discrepancies between the various information sources that underlie these processes. Fixing this problem requires expensive and time-consuming manual reconciliation. The principal activities required to sustain SBAs that collectively enact end-to-end processes include the collection, management, analysis, and interpretation of various business activities and data to make more intelligent and effective transaction-related decisions.

SBAs that collectively enact end-to-end processes in SSNs typically realize well-defined processes such as processing, shipping, and tracking payments, determining new product offerings, granting or extending credit, and managing market risk. Although such standard processes might drive transactional applications between SSN partners, they’re completely external to current Web services transaction mechanisms and are only expressed as part of application logic.

Business transactions will be able to manage fine-grained SBA tenets, such as critical business data, events, operations, local and aggregated QoS, and associated key performance indicators (KPIs). This will guarantee a continuous and cohesive information flow, correlation of end-to-end process properties, and termination and accuracy of interacting business processes driven by application control (integration) logic. In addition, in contrast to conventional service systems, transactions will pool real-time information from sensors and meters to achieve such correlations.

These concepts give rise to a multimodal transaction model that enables reliable smart transactions spanning from front-end SBAs to back-end system-level transaction support and beyond to organizations that belong to the SSN. The smart business transaction model then becomes the framework for expressing detailed operational business semantics. Conventional approaches to business transactions, such as Open Electronic Data Interchange (EDI), Unified Modeling Method (UMM), and electronic business XML (ebXML), focus only on the documents exchanged between partners, rather than coupling their application interfaces, which inevitably differ.

**Motivating Scenario**

Figure 2 depicts the flow of information between the interacting nodes (business partners) in a heavily simplified SSN involving three parties: a car manufacturer, part suppliers, and logistics service providers. A service-level agreement (SLA) governs these service interactions. The figure depicts the sequential ordering of the interaction events between the business partners in terms of message exchanges, assuming that each partner has effectively deployed an SBA that’s compatible with the message exchanges and associated SLA. As the figure shows, the business-aware transaction “Purchase part” (top) packages message exchanges 1, 2, 6, 7, and 8, which is governed by an SLA between the car manufacturer and the supplier (see the top right in the figure). The entire scenario comprises three business transactions, but due to space considerations, we’ve only included one SLA.

In general, the SLA stipulates a set of policy constraints, including temporal and spatial constraints, penalties, business rules, and regulations that participating SBAs must fulfill during transaction execution. It also prescribes recovery strategies for the business transaction in case any message exchanges fail—for instance, if the supplier sends an invoice message after unintentionally having failed to successfully issue a forecast, either the supplier must compensate the business transaction “Purchase part” via some means (such as issuing another forecast), or the entire transaction fails. In addition, the SLA drives the business-aware transaction with other agreements on the transaction’s KPIs, which enables the network partners to monitor and measure the performance of associated SBAs and
end-to-end processes in the network.

In our motivating scenario, the SLA unambiguously renders penalties in cases of non- or partial delivery, temporal constraints including delivery time and shipment duration, and KPIs, such as average order amount, discount-leverage, and cost per invoice.

The transaction management system should monitor and enforce several conditions in the SLA:

- Has the supplier acknowledged the order?
- Has the supplier and logistics service provider committed to a shipment date?
- Will the supplier start manufacturing on time?
- Will the supplier be able to get the order shipped on time?
- Does this order meet on-time delivery goals and other KPIs?
- If the order the logistics provider shipped doesn't arrive on time, how should we proceed?
- Does it affect other partners if the logistics service provider can't deliver the order? How do we compensate for this problem?

Traditional transaction monitoring mechanisms can monitor only system-related events and situations. However, it’s important to understand that business-aware transactions correlate application-level events with the supporting infrastructure. In this way, business transaction management is leveraged by tracking business transactions’ progress and dealing with process anomalies or breakdowns before they occur, while at the same time understanding repercussions at the SBA and support infrastructure levels. For example, if the manufacturer requests a change in the order, can we accept that change in connection with an agreed-on clause in the SLA regarding on-time delivery? More importantly, we can ascertain whether processes are still working to plan, if there are any bottlenecks and where they appear, if a process performance improvement or deterioration occurred, and so on.

Events in business transactions might stem from other virtual (software) services or wireless sensor networks. For example, a part bin signals the need for new parts by emitting a “notify of forecast” message to the part supplier (triggering an associated software service) before it actually runs out of stock. Another example pertains to a warehouse picker that — while at the output portal — receives a notification from an incoming part on which bin to deliver it to.

Marrying the business transaction model, which explicitly links business events to system-level events and messages in underlying SBAs, with real-time information gathered from wireless sensor networks and RFID tags will thus help deliver “smarter” business transactions.

### The Road Toward Smarter Business Transactions

The concept of networked enterprises has been around for a long time, but the decrease in wireless sensor and mobile device costs, together with the increase in enabling service and cloud technologies and global standards, will soon turn them into a reality. Indeed, this is already happening as we witness the birth of SSNs for energy, water, and traffic management, education, healthcare, and government.

The recent advent of Web service technologies and open standards such as the Web Services Description Language, BPEL, WS-Policy, and WS-Transaction has helped evolve our thinking about how distributed applications in service networks can connect and work together. However, none of these core Web service specifications and standards by themselves describe how we can connect individual services to create dependable, mission-critical solutions with scalability and complexity to guarantee the absolute completion and accuracy of interacting business processes — qualities that are critical for building and effectively managing service networks.

To remedy this situation, we’ve argued for a novel business transaction model in this article that caters to business transactions’ very fluid and dynamic nature, and with which business transactions can dynamically retract activities, try out alternatives, and allow optional activities to fail without influencing the successful completion of the associated end-to-end process.

The transaction model we discussed will lead to a comprehensive set of concepts and several standard primitives and conventions that service developers can use to realize complex SBAs in SSNs involving transactional end-to-end processes. This approach will mimic business operational semantics and won’t depend on underlying technical protocols and implementations. Mission-critical composite applications will differ from the smaller-scale composite applications built using BPEL extensions by being smarter and capable of dealing with complex, multiparty transactions in service networks. Such applications will understand not only system-level transactions but also how business-level transactions are progressing in terms of mutual commitments.
Web-Scale Workflow

into an initial version of the business transaction model. We presented a brief summary of this research work in this column. Interested readers can find more details elsewhere (see www.s-cube-network.eu/results/deliverables/wp-jra-2.1).

Acknowledgments

The research leading to these results has received funding from the European Community’s Seventh Framework Program under the Network of Excellence S-Cube, grant agreement 215483.

References


Michael P. Papazoglou is a professor at Tilburg University, where he is the executive director of the European Research Institute in Service Science and the scientific director of the European Network of Excellence in Software Services and Systems (S-Cube). His research interests are in service-oriented computing, Web services, large-scale data sharing, business processes, and federated and distributed information systems. He has published 20 books, monographs and international conference proceedings, and well over 150 journal and conference papers. Contact him at //preferred email?//

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.