

---

# RosettaNet: Adoption Brings New Problems, New Solutions

Suresh Damodaran

Copyright © 2005 Author

## Abstract

The first phase of RosettaNet innovation and deployment was fuelled by the early challenges of achieving standards-based interoperability and making B2B integration work over the Internet. In the second phase, RosettaNet is working to reduce the cost of multi-enterprise collaboration to increase the depth of collaboration and to encourage small- and medium-sized enterprises to participate and thereby increase the breadth of multi-enterprise collaboration. This paper focuses on the XML-based technologies and methodologies that RosettaNet is using to address the principal challenges of the second phase, and shares some insights that may be useful for those facing the challenge of creating standards for information exchange within an enterprise or between enterprises.

## Table of Contents

1. Anybody Can Create a FOOXML .....	3
2. Introduction .....	3
3. Partner Interface Process (PIP <sup>®</sup> ) .....	3
3.1. What is a PIP? .....	3
3.2. Creation and Use .....	4
3.2.1. Create and Specialize .....	4
3.2.2. Implement .....	4
3.2.3. Execute .....	5
4. RosettaNet: Evolving Phases .....	5
5. PIP Implementation Cost Influenced by PIP Creation Techniques .....	7
5.1. Improving PIP Creation to Reduce Cost .....	8
5.1.1. XML Schema for Representation .....	8
5.1.2. Shared UML Models .....	9
5.1.3. Automated XML Schema Creation .....	9
5.2. Efficient Messaging for Collaboration .....	10
5.3. Convergence of Product Information Exchange and Business Information Exchange .....	10
5.4. Increasing the Breadth of Integration or Driving B2SME .....	11
5.4.1. Fencing Standards Mutation .....	11
5.4.2. Enabling the non-24X7 SME .....	12
6. Conclusion .....	12
Acknowledgements .....	12
Bibliography .....	13

# 1. Anybody Can Create a FOOXML...

FOOXML was created by a set of spirited XML enthusiasts from the Fancy Objects and Oddities (FOO) industry who believed they were going to change the rules by which B2B integration was played. FOOXML was created by the FOOIC (FOO Industry Consortium). It took only about one year to get key individuals from each of the big companies in FOO to sit around the same table (mostly virtually from their desks thousands of miles away) and create a *Purchase Order Request* and a *Purchase Order Confirmation* business document. Soon, they had Version 1.0 of the business document.

FOOIC was entrusted with maintenance of the business documents. In time, more and more people adopted the business document standard from the FOO industry, and requests for changing some of the code lists, and even some elements in the Schema poured in. More versions were created. The FOO industry also needed more business documents. FOOIC engaged some other individuals who were new to the process of creating business document standards to create the *Invoice* and *Advance Shipping Notice* Schema because the original individuals who created the *Purchase Order* and *Purchase Order Confirmation* documents were considered too valuable to waste time with the standards building. These new documents were also considered a big success. Some engineers in the IT division who were processing these documents noticed that these new business documents used a different structure for elements such as *ContactInformation* and *Address* than were used in the earlier business documents. Sometimes the same concept was represented using similar sounding but different terms, and often had different elements, cardinalities, and attributes. However, not many people paid much attention to the engineers. After all, what do engineers know about business!

## 2. Introduction

While the story about FOOXML is entirely fictional, I maintain that it is quite plausible. It is well established that anybody who spends some time learning DTD or XML Schema can create a business document and a “standard.” We have evidence of such standards in several industry specific organizations, and even in OASIS. But to ensure that versions of these documents are maintained, and that multiple, related business documents share the same syntax and semantics for the same business concepts, a serious engineering effort is required. This paper describes this effort as well as several surrounding issues that come with adoption of such a standard for business-to-business integration (B2B), using RosettaNet as an example. While I am mostly concerned with business documents and their use in this paper, I assert that this argument applies to any “standard” that expresses itself using XML Schema or DTDs.

## 3. Partner Interface Process (PIP<sup>®</sup>)

RosettaNet was founded in 1998 to develop a global language for e-business. To achieve this goal, RosettaNet develops specifications for business documents called Partner Interface Processes<sup>®</sup> (PIPs<sup>®</sup>) and details how to exchange business documents reliably and securely over the Internet.

### 3.1. What is a PIP?

RosettaNet developed the Partner Interface Process (PIP) specification to enable trading partners to use shared (also called public) business processes. The PIP standardizes public business process automation by standardizing business documents, the sequence of sending these documents, and the physical attributes of the messages that define the quality of service. A business document with its envelope is called an action message because it requires a business-level action by the recipient. Currently, PIPs are classified as one-action or two-action, based on the message exchange patterns of action messages. The two-action PIP has a pair of action messages, a request and a response. In contrast, the one-action PIP only has a notification message. Additional physical messages, such as Acknowledgements, and Errors are exchanged at a lower level corresponding to each action message, and are termed “signals.”

PIPs are also classified according to the business areas they serve. The business areas, called Clusters, are partner and product service, product information, order management, inventory management, marketing and information manage-

ment, service and support, and manufacturing. Each Cluster is further divided into Segments. For example, PIP 3A4 [PIP3A4] is a two-action PIP with two action messages: *PurchaseOrderRequest* and *PurchaseOrderConfirmation*. This PIP belongs to Cluster 3, Order Management, and Segment A, Quote and Order Entry, and is the fourth PIP in Segment A.

For more information about RosettaNet and to obtain details on the available PIPs, see the RosettaNet Web site at [www.rosettanel.org](http://www.rosettanel.org) [http://www.rosettanel.org/].

## 3.2. Creation and Use

To understand the current challenges related to developing and implementing PIPs, this paper presents a brief overview of the PIP life cycle. The PIP life cycle begins with creating and specializing the PIP, followed by implementation and deployment of the PIP, and ends with executing transactions that conform to the PIP. Table 1 summarizes the participants and the stages in which they are involved.

Stage	Participant	Key Concerns
Create	RosettaNet	Consistency, clear specification of constraints
Specialize	RosettaNet Member	Minimizing deviation from generic PIP
Implement & Deploy	Supply Chain Company, Solution Provider	Security, alignment of public and private business processes, partner onboarding
Execute	Supply Chain Company	Scalability, traceability

### PIP Life Cycle

We describe each of these stages in more detail below.

#### 3.2.1. Create and Specialize

PIPs are created through the collaboration of teams responsible for developing the business requirements and the DTD or XML Schema for the PIP. Members of the business requirements team come from multiple companies. This team creates detailed requirements documents that are used as input in creating a PIP. A separate technical team creates the DTD or XML Schema for the PIP. Usually, at least one member of the technical team is also a member of the business team to provide continuity. Special attention is paid in this stage to capture the business constraints and represent in the PIP. Reuse of elements in the existing PIPs is an important consideration at this stage. After the PIP is created, it is validated in the field by deploying and exchanging test or sometimes production PIP messages that conform to the new PIP. Any issues found are resolved, and a validated version is released.

Some users (usually RosettaNet members) of a specific RosettaNet PIP may also create a specialized variation of that PIP for themselves and their trading partners. The generic version and the specialized variations are deployed in the field. Special attention must be paid at this stage so that changes to specialize the PIP are made in a manner consistent with the way that PIPs are created. Without such consistency, the improvement made during the creation stage may not be reflected in the specialized versions. RosettaNet also accepts change requests for PIPs from its members and, on occasion, may issue new versions of PIPs.

#### 3.2.2. Implement

The next stage in the PIP life cycle is implementation. A typical scenario is that trading partners request that a company implement one or more RosettaNet PIPs. The company searches for a RosettaNet solution provider. A RosettaNet solution must have at least two key capabilities. First, the solution must provide an interoperable implementation of the messaging service, RosettaNet Implementation Framework (RNIF). Two versions of RNIF are deployed currently,

RNIF 1.1 [[RNIF11](#)] and RNIF 2.0 [[RNIF20](#)]. The second requirement is a translator for the PIP messages. In a production environment, this translation is done in two steps. An industrial-strength mapper is used to map the PIP DTD or XML Schema elements to and from internal data representations of these elements. The map is used to translate the PIP messages, as described in Section 3.2.3, *Execute*. In addition to these two key capabilities, a RosettaNet solution may have several additional features, ranging from adapters (to integrate with popular internal applications such as ERP, CRM, and CAD) to sophisticated track-and-trace capability for PIP messages. When the implementation is complete, the solution is deployed.

Some key concerns during implementation and deployment are:

Which PIPs to implement

Specialized interpretation requirements for each PIP implemented from the trading partner

Back-end integration needs for each implemented PIP

Firewall configuration, S/MIME certificate (for signing and encryption) procurement and distribution, setting up SSL, and other security concerns

Schedule for each trading partner's participation in PIP exchange

Exchange of connectivity profiles, including port numbers and URLs and agreement with partners on response time, expectations, and other Quality of Service parameters (this exchange is also referred to as the “on-boarding” process)

Testing.

Note that trading partners are involved during deployment or at any time during execution. The on-boarding process for a trading partner may take days, or even months, because of the interoperability issues involved and the security concerns of either party regarding exchanges over the Internet.

### 3.2.3. Execute

During execution, the messages created in conformance to PIP specifications are exchanged between trading partners (enterprises) using RNIF. The inbound and outbound maps created during the implementation stage for a given PIP are used to translate the PIP messages sent in and out of an enterprise. An important activity at this stage is to be able to debug problematic business processes by tracing specific business documents exchanged. While RosettaNet does not specify how this tracing should be done, RosettaNet solutions usually support such activity in varying degrees.

The distinctions among the create & specialize, implement & deploy, and execution stages are pronounced. RosettaNet controls only a part of the first stage, creation of the PIPs. RosettaNet partners specialize PIPs. Companies that implement RosettaNet PIPs control the implementation stage. RosettaNet solution providers aid these companies by providing RosettaNet solutions. IT departments or the management of the respective companies involved in the deployment and execution of PIP-based transactions control the execution stage. For a more detailed analysis of the efforts required in the implementation stage, see [[WSJSD](#)].

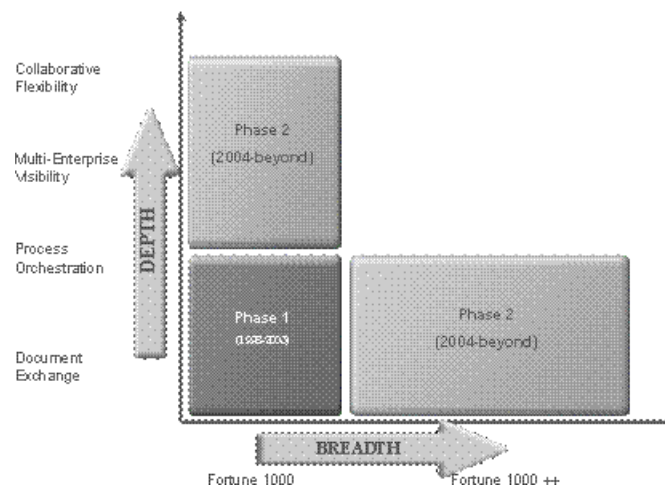
## 4. RosettaNet: Evolving Phases

In the first phase of adopting RosettaNet specifications—from 1998 to 2003—large and influential enterprises invested in direct connections with top-tier partners over the Internet. The focus in this phase was to design the business documents in XML with limited message exchange patterns (known as PIP choreography) and put them into production as quickly as possible. In fact, RosettaNet has always been perceived within the hi-tech industry as the global standards body that has better turn around times (roughly 18 months) for standards than the corresponding EDI standards bodies.

The primary driver for adoption in Phase 1 was the perceived cost advantages of using direct connections over the Internet to integrate with a trading partner instead of using a third-party intermediary [VAN1]. In addition, the use of direct connection with a trading partner permitted interaction with partners at any time of the day, instead of the once per day traditional routine of batch-oriented EDI transactions. While both of these reasons justified the use of RosettaNet for the major supply chain influencers, for many of their partners, at least initially, the reason for implementing RosettaNet was the need to keep the existing business with their important customer. This reason remains the same for some of the enterprises, because their trading volume is fairly low, and RosettaNet is just a cost of doing business.

The adoption of RosettaNet for interaction with partners over the Internet in Phase 1 has also produced other benefits. The information delivered by the PIP needs to be integrated with the back-end systems, and this encouraged a review of internal business processes, which resulted in fixes that improved efficiency. Errors introduced through multiple data entries also are considerably reduced because the RosettaNet solutions grew to provide additional internal integration capabilities. Some companies have also realized additional benefits by wider use of PIPs for newer business processes. For example, some of the supply chain planning activities (demand forecast) that formerly were manual are now implemented using RosettaNet PIPs, which considerably reduces errors and increases timely availability of information [INTSH].

The major achievements of Phase 1 were to establish RosettaNet as a reliable and secure method of exchanging business documents over the Internet, and to demonstrate that RosettaNet is a promising alternative to exchanging documents through established intermediaries known as Value Added Networks (VANs), especially with large exchange volumes. However, this argument is losing steam because the cost of VAN services has shown a downward trend. Therefore, the return on investment (ROI) of the RosettaNet implementation over the years is less than expected when predicated over the savings from VAN services.



**Figure 1. Evolution of RosettaNet**

Therefore, for continued adoption, RosettaNet needs to identify other business drivers and adapt to these new drivers. One such business driver is the use of RosettaNet to increase the depth of collaboration with trading partners. Since RosettaNet defines business documents to support newer business processes that replace manual processes, the use of these new business processes can drive collaboration between companies much deeper and tighter, increasing transparency and flexibility in the supply chain. Another business driver is the need to automate business exchanges between the large number of small and medium enterprises (SME) to reduce errors in documents such as purchase orders and invoices.

Thus, the evolution of RosettaNet will be characterized by depth of multi-enterprise collaboration and breadth of multi-enterprise collaboration. Figure 1 describes the depth and breadth of adoption of RosettaNet by Fortune 1000 companies during Phase 1 and projects the depth and breadth of adoption by Fortune 1000+ companies during Phase 2.

The vertical axis of Figure 1 describes the needs related to depth of integration that RosettaNet must address. A PIP transaction includes an exchange of business documents, within a PIP choreography. The PIP choreography is limited to either a single-action PIP or a two-action PIP. This PIP choreography provides a limited amount of process orchestration that spans a single PIP. The process orchestrations must evolve to include multiple PIPs to support end-to-end processes such as order-invoice-ship-pay (also known as order-to-cash process). Multi-enterprise supply chain visibility enables an enterprise to collect and analyze distributed supply chain events, such as delayed shipment; to generate specific recommendations, such as expedited shipment; and to reach predefined supply chain business goals, such as on-time delivery. Such visibility can considerably improve the efficiency of supply chain execution and planning. Beyond enabling multi-enterprise visibility, RosettaNet also needs to enable multi-enterprise collaboration scenarios of intensive collaboration among a few trading partners, which exhibit multi-enterprise collaboration and intense real-time interactions at a much higher degree than before. For example, customer involvement in product design, and dynamic logistics planning are two business processes that require intense real-time interactions between multiple trading partners.

The horizontal axis of Figure 1 describes RosettaNet-based integration, or community adoption over time. As the graph shows, although some large enterprises may have thousands of trading partners, only a small percentage of these partners are electronically integrated. RosettaNet has an opportunity to create new technologies and approaches to expand the adoption. The errors introduced by humans reading a form and typing into another form (known as the multiple-data-entry problem) are costly, and electronic integration is the only viable solution to this problem.

Phase 2 of RosettaNet is evolving based on two hypotheses. One hypothesis is that reducing the technical cost of collaboration, as well as allowing innovative use of information already being exchanged via PIPs, will better position RosettaNet to play a key role in efficiently increasing the depth of multi-enterprise collaboration. Another hypothesis is that reducing the cost of entry into the RosettaNet-integrated world through low-cost integration solutions affordable to small- and medium-sized enterprises (SME) will encourage widespread adoption. The remainder of this paper describes the efforts to validate these hypotheses in the business community.

## 5. PIP Implementation Cost Influenced by PIP Creation Techniques

About 50 percent of the PIP implementation effort is spent on semantic mapping of elements in the PIP to and from the back-end applications [WSJSD]. Therefore, the cost of integrating PIP action messages with the enterprise applications is high. As described in Section 3.1, every PIP action message requires an inbound and an outbound map. Anything that increases the complexity of creating these maps increases the human effort and cost required for the mapping, and may increase the deployment cost if deployment is delayed because of the complexity. A study of PIPs 3A5, 2A9, 2A12, 3A4, 3A7, 3C3, and 3C4 in 2001 [LEAN] identified the following factors as significantly adding to the cost of implementing a PIP:

- Ambiguity, incompleteness, and inconsistencies in PIP specifications

- Lack of reuse across PIP specifications

- Too many options in action message contents

The deficiencies listed above were identified in DTD-based PIPs, which, in addition to the DTD, require another document that defines additional constraints on the elements of the DTD. Thus, the deficiencies apply to both the DTDs and the corresponding message guidelines.

PIP 3A4, PurchaseOrderRequest Message Guidelines, provides an example of ambiguity. The PIP states the constraint for *GlobalTransportEventCode* as follows:

“Only one occurrence each of *GlobalTransportEventCode* equal to “Ship,” “Dock,” and “Pickup” Is allowed.



*GlobalTransportEventCode* occurs within *ProductSubLineItem*. There can be 0 or more *ProductSubLineItem*. It is unclear whether *GlobalTransportEventCode* applies to each *ProductSubLineItem* individually, or to all of them.

Incompleteness often accompanies PIPs that have numerous options for elements. Inadequate explanation of the reason for the options and the absence of cross-element dependencies present problems. For example, consider the statement “At least one occurrence of *GlobalProductIdentifier* or *PartnerProductIdentifier* is required.” Is it acceptable to have both?

When multiple ways are available to provide the same information, there is also the potential to have contradictory information provided by these multiple ways. For example, in PIP 3A4, *shipTo.PartnerDescription* is mandatory and can be provided at the *ProductLineItemLevel* or *ProductSubLineItemLevel*. A convention for override identifies the correct value of *shipTo.PartnerDescription* to apply at the *ProductSubLineItemLevel*. Such a convention does not negate the requirement to define values for mandatory elements at all levels, even if they are overridden. Another type of inconsistency occurs when the same element consists of different properties, or a different number of properties, across the same or different PIPs.

These problems require considerable human effort to resolve and increase the cost of implementing PIP-based, multi-enterprise collaboration. They often result because the teams that develop the business requirements and the teams that develop the DTDs do not collaborate effectively during development to ensure consistency. Sometimes the only things they share are the general process of creating PIPs and a business dictionary that defines common terms. Coordinating the effort between and across the teams during the development phase is the best way to avoid problems and reduce costs. The next section explores the various ways such coordination is achieved.

## 5.1. Improving PIP Creation to Reduce Cost

After the implementation issues discussed in the previous section were identified, RosettaNet revisited several aspects of PIP creation in an effort to improve the process and reduce costs. Some problems were process oriented; some were technical. The general theme has been to avoid handcrafting the end user XML Schemas as much as possible, and to promote sharing of information across multiple PIPs. The following practices are being implemented during the development phase to address concerns during implementation enumerated earlier:

Using XML Schema for representation of the PIPs,

Using UML models to capture business requirements,

Sharing UML models across PIPs, and

Automating XML Schema creation from the UML model.

Sections 5.1.1, 5.1.2, and 5.1.3 discuss these practices in more detail and describe some issues RosettaNet encountered when these new practices were being adopted (2003-2005).

### 5.1.1. XML Schema for Representation

The expressive power of DTD is simply inadequate to represent the constraints on the elements of the PIP action messages, as described in the previous section. XML Schema is far superior to DTD in capturing most of the constraints unambiguously and completely because of its superior expressive power. For example, while DTD has 10 data types, XML Schema has over 44 built-in types, and allows creating your own data types. XML Schema, however, has some inadequacies. One is the inability to express cross-element dependencies. For example, “*businessName* (which occurs on Lines: 20, 55, 124, 190, 227, 302, 332, 376, 421, 471, and 501 of PurchaseOrderRequest [PIP3A4]) has the following constraint: “Mandatory if *GlobalBusinessIdentifier* is not used.” The constraint on *businessName* clearly depends on the occurrence of *GlobalBusinessIdentifier*.

RosettaNet uses Schematron [STRON] to express cross-element dependencies within XML Schema. After considerable analysis of the existing cross-element dependencies in PIPs, some templates utilizing the value or existence of elements



have been created. These templates are helpful in avoiding handcrafting new Schematron expressions when engineers create PIPs. RosettaNet PIPs created after the adoption of these changes in the representation of the business documents no longer have Message Guidelines. An analysis of the constraints related to cross-element dependencies is forthcoming.

New and interesting issues have also emerged, with XML Schema-based (Modular) PIPs. Those implementers who are accustomed to using Message Guidelines with DTD based PIPs need a spreadsheet that will serve the same purpose as the Message Guidelines with XML Schema based PIPs, even though the spreadsheets are not normative. RosettaNet provides Message Descriptions in spreadsheet format now, even with XML Schema-based PIPs. The good news is that the Message Description generation process is also automated, requiring little human intervention.

One ongoing challenge with the use of XML Schema for PIP representation occurs when it is necessary to create new PIP Schema and new derived types from existing Schema and types. The challenge is to physically change as few existing modular Schemas as possible, because any change to an existing Schema requires creating a new version for the changed Schema. One solution to this problem is to use substitution groups to create derived types. However, substitution groups are cumbersome and require considerable forethought on where changes can be expected. Another approach is to consider XML Schema reuse as a corollary of UML class reuse. In this case, it is possible to create a derived class in UML without changing the base class, even though the corresponding XML Schema representation of the base class would change slightly.

### 5.1.2. Shared UML Models

Today, UML models are used to capture the business requirements from the business team (also known as the RosettaNet Milestone program team). Previously, the RosettaNet business dictionary contained definitions of terms used in the Message Guidelines. With the use of UML models, the business dictionary was expanded to include definitions of the UML models as well. Some of the UML classes are shared across all the PIPs. These classes are called Universal Structures. Other UML classes that are shared among a few PIPs usually deal with a single business domain, such as order management. These UML classes are called Domain Structures. The DTD-based PIPs have been refactored with considerable effort to remove ambiguity and to bring consistency to the process of creating these structures.

Inheritance and aggregation are used in the UML models to achieve consistency and to reduce the size of the XML messages. A style guide for UML representation is used to ensure that engineers follow the same conventions when they create the UML models. The shared UML models also need an easy method of storage and retrieval. RosettaNet utilizes the content management facilities of ebXML Registry to store and retrieve the shared UML models and the corresponding XML Schema.

The use of shared UML models also provides an opportunity to create information models that span an end-to-end process relating multiple PIPs. For example, an information model for order-to-cash may include Domain Structures from the following PIPs: PIP 3A4, PIP 3A7, PIP 3A8, PIP 3A9, PIP 3B2, PIP 4B2, PIP 3C3, and PIP 3C6. These PIPs represent a range of domains from order management and logistics to remittance. Use of the information model allows an entirely different view of the PIPs. PIP action messages are transformed from representation of individual business documents to vehicles for exchanging information represented in the information model implemented by each partner. Development of this view is a significant step towards innovative ways to exchange the information represented in the information model.

### 5.1.3. Automated XML Schema Creation

A key advancement in the process of creating PIPs is the use of an automated production line to convert the UML models to XML Schema that can be used for validation and release by the RosettaNet membership community and RosettaNet, respectively. An automated production line is necessary to prevent human-generated errors in the final PIP XML Schema. Creation of this production line required standardizing various aspects of the PIP creation process so that the conversion from UML to XML Schema could be automated. The following list contains a sampling of the standardization documents created by the RosettaNet Architecture Office and RosettaNet Engineering:

*Namespace Specification and Management*, Dec 2003

*XML Design Guidelines*, Dec 2003

*PIP UML Design and Modeling Guidelines*, Feb 2004

Standardizing every aspect of PIP creation forced RosettaNet to think through thorny issues of what released components should be versioned, how to represent the versioning, and how to keep traceability through the PIP creation process so that a problem in XML Schema can be traced back to the original UML model. Such a PIP made using XML Schemas with multiple physical Schemas in separate name spaces is called a “Modular PIP” in contrast to a DTD-based, “Monolithic” PIP, because a DTD-based PIP is defined in a single DTD, in a single file. The storage and retrieval of PIP Schema components and UML models are automated using a content management system based on ebXML Registry.

A challenging issue when creating Business Documents is, “How can a Business Document be designed to make it easy to change while maintaining conformity to the original Business Document?” XML Schema does help in describing data constraints a lot more precisely than DTD, though specializing XML Schema has become a challenge. The specialization problem, and how RosettaNet is addressing this challenge are described in Section 5.4.1.

Another significant area of standardization is PIP choreography. ebXML BPSS (Business Process Specification Schema) is used to standardize the exchange sequence of PIP action messages, and to specify the quality of service attributes related to the physical exchange of the messages. Previously, UML and text were used to describe the same information. Use of BPSS permitted “dropping in” the BPSS-compliant PIP choreography description in XML into an implementation, thus avoiding human keying in of these values. In practice, however, individual partners rarely change the choreography, whereas they change the business document embedded in the PIP action message frequently.

## 5.2. Efficient Messaging for Collaboration

As RosettaNet PIPs came to be used for complex collaboration (for example, to convey demand forecasts), the PIP message sizes grew to an average of 300MB or more. RosettaNet is working to provide solutions to the problem of the steady increase in the size of PIPs. One way to reduce the message size is by using compression. RosettaNet has released a Technical Advisory for S/MIME compression of the messages when they are sent over RNIF 2.0 [CMPTA]. An additional issue on availability of RNIF during peak PIP transaction rates also has been resolved [HIGH].

Another approach to reduce message size is to eliminate redundancy of content within messages. RosettaNet is currently exploring ways to reduce the message sizes using a referencing mechanism to enable references to information that has been already sent. Although not in production yet, RosettaNet has created prototypes of PIPs with embedded references, called “Referenced” PIPs. The ability to precisely point to information within a message will also considerably help in precise error notification. A Logistics Proof-of-Concept that incorporates Referenced PIPs along with EPC (Electronic Product Code, a standardized version of RFID) has been demonstrated live [UCON05].

## 5.3. Convergence of Product Information Exchange and Business Information Exchange

In the hi-tech industry, products are quite complex, and their characteristics change several times a year. The need to exchange product data with partners electronically in an efficient manner is urgent. The RosettaNet Technical Dictionary (RNTD) defines the commonly used types of product components (for example, resistor, DRAM), and their properties. For example, PIP 2A10, currently used to exchange Material Composition information to support the European RoHS directive [ROHS], is developed from the product information exchange business requirements. This PIP has a simple message structure with limited constraints expressed in the message itself (because of its generic nature), and the contents of the message must be processed with an extra step to decipher the types and constraints that must be applied to validate the message. These PIP messages have very little hierarchical information embedded in them. However, their lack of structure and their flexibility are beneficial when the types of information exchanged change frequently, and community agreement processes for PIP structure become a bottleneck rather than an aid to interoperability.

However, some users are frustrated because they are required to specify values for at least some of the unused yet mandatory elements in the PIPs. The large number of elements results because the PIP structure includes most of the business needs of the original developers of the PIP, and consequently may be a byproduct of design by committee.

Since the RNTD contains all possible types of products, only subsets of product types of RNTD are used for specific business processes. For example, PIP 2A10 to support Material Composition Reporting requires only a subset of RNTD to communicate hazardous material information. There are two approaches to efficiently communicating product information using RNTD. One approach is to create a new PIP to communicate every new subset of RNTD that will be required to communicate product information relevant to a specific business process, as in PIP 2A10. However, it is not always efficient to create a specific PIP for each such purpose. Rather, a generic PIP may be able to support multiple business processes that require RNTD. Using the generic PIP is the alternate approach. The more generic the PIP is, the fewer the constraints defined for the PIP; however, another Schema that describes the constraints corresponding to each PIP is necessary. Ironically, although generic PIPs do away with defining PIPs specific to individual business processes, “constraint definitions” to satisfy each individual business process become necessary. The only perceived advantage to RosettaNet users in this case will be the ability to change constraint definitions (also known as IDA, of Information Definition Agreement), which may involve a relatively easier RosettaNet community process than changing a PIP definition.

The majority of the PIPs created by RosettaNet and used by the membership have complex message structures and express quite a number of constraints in the messages themselves. PIPs that exchange product information have considerably simpler message structures different from other PIPs, as explained earlier. Within RosettaNet, the Engineering Information Management (EIM) program is developing a convergence plan for these different types of message structures. A key observation that has helped in the convergence plan is that even though product information exchanges need to be flexible, based on experience, it is possible to impose more structural constraints on PIPs, as discussed earlier. It is hoped that XML Schema-based agreements will bring the PIPs used to exchange product information closer to the structure of other RosettaNet PIPs. Meanwhile, the dictionary architecture has been improved considerably over RNTD under the RosettaNet Dictionary Architecture (RDA) initiative that concluded in 2005.

## 5.4. Increasing the Breadth of Integration or Driving B2SME

Extending RosettaNet integration to small- and medium-sized enterprises (SME) also requires considerable standardization. A significant challenge to B2SME integration is the affordability of the solution to the SMEs. Internet connectivity also presents a challenge to SMEs; few are expected to have 24x7 connectivity. Creating low-cost solutions for SME integration requires even more standardization, some of which requires specializing the standardizations discussed earlier. These specialized approaches to standardization are described below.

### 5.4.1. Fencing Standards Mutation

One source of the cost to SMEs is the same as that for depth of integration—the implementation and maintenance cost resulting from the variations in syntax and semantics of PIP action message structures. Major users of RosettaNet typically provide an implementation guide with the DTD-based PIPs they implement. The implementation guide clarifies the Message Guidelines issued by RosettaNet further, and modifies the PIP DTDs themselves, for example to include more code list values. RosettaNet previously did not establish rules to describe how one supply chain company can modify a PIP via the implementation guidelines specific to that company or by actually changing the DTD. The absence of rules creates a large number of subtly varying message semantics for the same PIP, which makes implementation difficult for small companies and distributors who need to trade with multiple partners.

The RosettaNet Automation Enablement (RAE) program addresses this issue by specifying rules on how any PIP can be changed. When a RosettaNet member changes a PIP, it is designated as a TPIR-PIP (Trading Partner Implementation Requirements-PIP). For example, PIP 3A4 can have multiple TPIR-PIPs issued by different RosettaNet members. While new maps (one for inbound, one for outbound) are required for each TPIR-PIP, the TPIR-PIP guidelines control the variations and the effort required to create these new maps. To support SMEs, a new presentation format called

TPIR-PF (Trading Partner Implementation Requirements-Presentation Format) has also been introduced for such TPIR-PIPs. This format has the benefit of maintaining the same presentation style across all the PIPs, and even for the same PIP required by multiple partners. SMEs who see only these forms have no need to use a map if the information in the PIP messages is not integrated with other back-end applications. TPIR-PFs may also be specified by the RosettaNet community member who creates the corresponding TPIR-PIP.

To help make the TPIR-PIPs and TPIR-PFs available easily and in an automated fashion, RosettaNet is considering offering a public repository service. This repository is being specified using both UDDI and ebXML Registry interfaces.

### 5.4.2. Enabling the non-24X7 SME

Bringing B2SME to a community of SMEs that are not connected to the Internet all the time (24X7) has its own challenges. This challenge is not unique to RosettaNet, and has manifested with EDI-based transactions as well. The traditional solution in EDI-based transactions is the use of an intermediary, known as a Value Added Network (VAN), which provides mailboxes to the SMEs for depositing their documents when they happen to be connected to the network. VANs traditionally operate on a private network, though most do provide on-ramping facilities via the Internet. A similar solution can be deployed over the Internet to enable SMEs that desire to be connected via RosettaNet, by either providing connectivity to existing VANs or by creating an Internet-only intermediary service. A specification that defines the rules such an intermediary must comply with while transporting RosettaNet messages may be helpful. Currently, RNIF only specifies a point-to-point messaging service.

The Multiple Messaging Specification (MMS) is being developed to enable transporting PIPs over multiple messaging services, specifically in compliance with EDIINT AS2, ebXML Messaging Service, and WS-I Basic Profile. Although this effort may ease the requirement to use RNIF 1.1 and RNIF 2.0 for exchanging PIPs, it will also increase the number of interoperable messaging services that support PIP exchanges. When the number of implementation options increases, the cost of solutions increases for both the solution providers and the customers. Therefore, the future adoption and support of these new messaging services will largely depend on how widely RosettaNet gets adopted.

Another opportunity for reducing cost is to standardize the configuration of RNIF implementations. Considerable effort is expended today to configure an RNIF implementation at a trading partner so that it would interoperate with an RNIF implementation at another trading partner. ebXML Collaboration Protocol and Profile and Agreement (ebXML CPPA) is a suitable standard for this purpose. RosettaNet has an opportunity to further reduce cost of on-boarding trading partners by standardizing on either ebXML CPPA, or a similar standard, and applying the standard uniformly across all of the messaging services being considered for the Multiple Messaging Specification program.

## 6. Conclusion

RosettaNet prides itself in being business-centric and utilizing existing technology in innovative ways to benefit its members. This paper describes the evolution of RosettaNet from a technology that makes use of the Internet to reduce VAN charges, to technology that brings depth and breadth to electronic integration. This 7-year transformation has been slow by the standards of the Internet age. However, considering that the fundamental ways information is exchanged using traditional EDI-based technologies has changed little over the past 30 years, the pace of transformation in RosettaNet is quicker. It is also important to consider that the transformations described in this paper are driven by the endeavors and feedback of RosettaNet members who use its technology.

## Acknowledgements

RosettaNet progresses thanks to individuals provided on loan to RosettaNet as well as a few on RosettaNet payroll. I have been fortunate to be part of this large community of passionate and dedicated individuals, spread all over the world. It is difficult to name only a few, but I would be amiss not to name at least a few of my former colleagues in

RosettaNet who influenced and contributed to the RosettaNet Integration Architecture in various capacities (in alphabetical order): John Cartwright (on loan from Intel), Derek Coleman (on loan from Hewlett-Packard), Hussam El-Leithy (RosettaNet), Alfred Elkerbout (on loan from Philips Electronics), John Mackin (RosettaNet Japan), Kenji Nagahashi (on loan from Fujitsu), Lavina Nagar (volunteer), Neelakantan Kartha (on loan from Sterling Commerce), Kamarul Zaman Abdul Rashid (on loan from Intel), Mark Schenecker (on loan from E2Open), Nikola Stojanovic (RosettaNet), Paul Tearnen (RosettaNet), and Frank Yang (RosettaNet). RosettaNet Architecture Advisory Committee (AAC) provided a good forum to discuss architecture issues. Thanks to my colleagues in RosettaNet AAC, and its co-chairs, Jean-Claude Monney (ST Micro Electronics), and Paul Tearnen (RosettaNet). The PIP Engineering team in Los Angeles, USA, and in Penang, Malaysia spent considerable time building the XML Schemas according to the various guidelines. Thanks to Angela Harris, Mike Houdeshell, Paul Tearnen, and Nikola Stojanovic for reviewing earlier versions. Thanks to Maggie Harris for considerably improving the readability of this paper.

## Bibliography

- [RNIF20] *RosettaNet Implementation Framework: Core Specification, V02.00.01*, RosettaNet, March 2002. <http://www.rosettanel.org>
- [RNIF11] *RosettaNet Implementation Framework Specification, V1.1*, RosettaNet, Nov1999, <http://www.rosettanel.org>
- [INTSH] Intel and Shinko use RosettaNet standards to build forecast-to-cash procurement process, RosettaNet Case Study, 2003. <http://www.rosettanel.org>
- [W3CSD] *B2B Integration over the Internet with XML – RosettaNet Successes and Challenges*, Suresh Damodaran, WWW2004, May 17-22, 2004, <http://www2004.org/proceedings/docs/2p188.pdf>
- [SCTRN] Schematron: An XML structure validation language using patterns in trees, <http://www.ascc.net/xml/resource/Schematron/Schematron.html>
- [WSJSD] Automating B2B Integration with XML: The RosettaNet Approach, Suresh Damodaran and Neelakantan Kartha, XML Journal, <http://xml.sys-con.com/read/45068.htm>
- [PIP3A4] PIP3A4: Request Purchase Order, Version 2.01, RosettaNet, Aug. 2002. <http://www.rosettanel.org>.
- [VAN1] Vendor Managed Inventory Case Study: Working Together, U-Connect Conference, July 8, 2005. [http://www.uc-council.org/uconnect/RosettaNet/session\\_descriptions.cfm](http://www.uc-council.org/uconnect/RosettaNet/session_descriptions.cfm)
- [LEAN] Legacy PIPs Analysis and New Architectural Principles, V1.03, 24 Aug 2001, *RosettaNet Internal Document*.
- [UCON05] Logistics Proof of Concept- Evolving the Architecture to Enhance Logistics Visibility, U-Connect Conference, July 7, 2005. [http://www.uc-council.org/uconnect/RosettaNet/session\\_descriptions.cfm](http://www.uc-council.org/uconnect/RosettaNet/session_descriptions.cfm)
- [CMPTA] SMIME Compression of RosettaNet Payload: RosettaNet Implementation Framework V02.00.01, July 2003. <http://www.rosettanel.org>
- [STRON] Schematron: An XML structure validation language using patterns in trees. <http://www.ascc.net/xml/resource/schematron/schematron.html>
- [ROHS] Directive 2002/95/EC of the European Parliament and of the council of 27 January 2003 on the restriction of the use of certain hazardous substances in electrical and electronic equipment. Official Journal of the European Union, L 27/19, February 13, 2003. [http://europa.eu.int/eur-lex/pri/en/oj/dat/2003/l\\_037/l\\_03720030213en00190023.pdf](http://europa.eu.int/eur-lex/pri/en/oj/dat/2003/l_037/l_03720030213en00190023.pdf).
- [HIGH] Developing High Availability Features to Address Failure Recovery of the RNIF, OASIS Symposium, April 27, 2004. [http://www.oasis-open.org/events/symposium/pre\\_program.php](http://www.oasis-open.org/events/symposium/pre_program.php)

## Biography

### Suresh Damodaran

[Sterling Commerce](http://www.sterlingcommerce.com/) [<http://www.sterlingcommerce.com/>]

750 West John Carpenter Fwy

Irving

Texas

United States of America

[suresh.damodaran@rosettanet.org](mailto:suresh.damodaran@rosettanet.org)

Suresh Damodaran is Technology Director with Sterling Commerce ([www.sterlingcommerce.com](http://www.sterlingcommerce.com/) [<http://www.sterlingcommerce.com/>]), a subsidiary of SBC Communications. From September 2002 to September 2004, Suresh was the Chief Technologist of RosettaNet ([www.rosettanet.org](http://www.rosettanet.org) [<http://www.rosettanet.org/>]), on loan from Sterling Commerce. Currently, Suresh serves on the RosettaNet Architecture Advisory Committee. Suresh co-authored *ebXML: Concepts and Application*, published by Wiley. Suresh holds a B.Tech.(Hons) in Electronics and Electrical Communication Engineering from the Indian Institute of Technology at Kharagpur, and a Ph.D. in Computer Science from the University of Louisiana.