Abstracting Business Modeling Patterns from RosettaNet

Pankaj R. Telang and Munindar P. Singh

North Carolina State University, Raleigh, NC 27695-8206, USA
prtelang@ncsu.edu, singh@ncsu.edu

Abstract. RosettaNet is a leading industry effort that creates standards for business interactions. The RosettaNet standard defines over 100 Partner Interface Processes (PIPs). The PIPs specify the business interactions at a syntactic level, but they fail to capture the business semantics of those interactions. This paper builds on a commitment-based approach for business modeling. It presents an approach that defines commitment-based business patterns abstracted from RosettaNet PIPs. This paper validates the patterns by using them to model a well-known real-life use case of a cross-organizational business interaction.

1 Introduction

The competition in the global economy compels organizations to provide high-quality products and services at an attractive cost. It forces organizations to identify contextual processes and activities, and to outsource them to other organizations that specialize in executing those processes. A few examples of such processes are human resources management, workplace management, payroll, call center, and IT infrastructure management. By outsourcing such processes, an organization reduces operational expenses, and at the same time gains access to specialized resources for the contextual work. The outsourcing results in a network of organizations. To conduct business, these organizations engage in a complex set of service interactions.

RosettaNet [6] is a leading industry effort that creates standards for business interactions. The RosettaNet consortium consists of over 500 organizations of various sizes, and from various industry sectors. These organizations use elements of the RosettaNet standard, named Partner Interface Processes (PIPs), to transact business that is worth billions of dollars. A PIP specifies two-party interactions for some specific business purpose. For example, a buyer requests a quote from a seller using PIP 3A1, and a seller requests financing from a financing processor, on behalf of the buyer, using PIP 3C2. A PIP specification contains a document that informally describes the purpose of the PIP, any assumptions, and the outcome. Additionally, a PIP specification provides an XML DTD or XML Schema that corresponds to the structure of the messages that the participants in the PIP would exchange.

RosettaNet PIPs specify the business interactions at a syntactic level, but they fail to capture the business meaning of those interactions. For example, in PIP 3A1, a buyer sends a request for quote to a seller, and the seller responds either with a quote, or a referral. RosettaNet leaves important details unspecified. If the seller responds with a

quote, does the seller commit to the buyer to sell the goods at the quoted price? Does the buyer commit to the seller to buy the goods at the quoted price if the buyer acknowledges the quote? The PIP leaves such questions regarding the business meanings of the interactions to human interpretation.

In our previous work [10], we present a high-level *business metamodel* to capture the way in which cross-organizational service engagements are carried out. This paper uses the business metamodel to abstract business *patterns* from the RosettaNet PIPs. Note that these patterns are not operational patterns like the workflow patterns [7]. Instead they are high-level patterns that specify the essence of the business. They provide flexibility in business execution.

There are two key motivations in abstracting these patterns. First, since the patterns are at a business level, business analysts can easily compose them to develop a desired business model. Second, a model composed from these patterns serves as a formal specification that can be used to verify an operational model defined in any technical standard. Organizations frequently migrate their business process implementations to newer technologies to benefit from the improvements those technologies offer. In such technology migration, the high-level formal specification verifies the correctness of the new implementation.

Contributions The main contribution of this paper is an approach for abstracting business patterns from RosettaNet PIPs. It applies the patterns to a real-life business scenario. The paper highlights the shortcomings of the existing PIPs, and identifies new PIPs that are necessary to model complex service engagements.

Organization Section 2 provides a background on the RosettaNet PIP standard and introduces our business metamodel. Section 3 presents business patterns for a subset of the RosettaNet PIPs. Section 4 applies the patterns to model an aerospace scenario. Section 5 concludes the paper with a discussion of the related work and some future directions.

2 Background

We now review some key background on RosettaNet and on our approach.

2.1 RosettaNet

The RosettaNet standard specifies over 100 PIPs for various business processes in the eCommerce supply chain. The standard classifies the PIPs using clusters and segments. A cluster represents a major business process of the supply chain. A cluster is subdivided into segments that represent subprocesses of the cluster's business process. Each segment contains many PIP specifications. For example, Cluster 3 represents the Order Management process, and Segment A of that cluster represents the subprocess Quote and Order Entry. Segment A contains PIPs such as 3A1, 3A3, and 3A4. The standard

employs three views to specify a PIP, the Business Operational View (BOV), the Functional Service View (FSV), and the Implementation Framework View (IFV).

The BOV informally describes the business process that the PIP implements. It specifies the two partner roles, and the business activities that the roles execute. A business process flow diagram shows the business activities, and the flow of business messages between the roles. For each activity, BOV specifies performance controls such as the need to acknowledge receipt, nonrepudiability of the receipt, and the timeout for the acknowledgment.

For a PIP, the FSV derives from the BOV, and specifies the RosettaNet services. It specifies the message exchange sequence in a business transaction dialog, and for each message, it specifies message exchange controls. These controls include the time within which an acknowledgment is required, the time within which a response to an action is required, and whether authorization is required for an action.

The IFV specifies the message formats as XML DTDs or XML Schemas. For each message, it specifies the communication requirements such as the need for a digital signature, and secured transport.

2.2 Business Metamodel

This discussion is extracted from our previous work [10]. A business model specifies how business is conducted. We concern ourselves with business models that involve two or more participants. The business *participants*, abstracted as *roles*, participate in a *business relationship*. The participants create, manipulate, and satisfy *commitments* in a relationship. They execute *tasks* for each other that enable them to achieve their respective *goals*.

Three distinct phases characterize business execution. First, in the *agreement* phase, participants enter into an agreement, and *create* commitments toward each other. Second, in the *assembly* phase, the participants *delegate* or *assign* commitments to others. A participant may delegate a commitment that requires the execution of a task which is not a core competency of that participant, or due to some other economic motivation. Third, in the *enactment* phase, participants execute tasks to *satisfy* their commitments.

Figure 1 describes the business metamodel.

Agent: a computational representation of a business participant. An agent has goals, and executes business tasks. For each business relationship in which an agent participates, it enacts one or more roles.

Role: an abstraction over agents that helps specify a business relationship. Each role specifies the commitments expected of the agents who play that role along with the tasks they must execute to function in that role.

Goal: a state of the world that an agent desires to be brought about [1]. An agent achieves a goal by executing appropriate tasks.

Task: a business activity viewed from the perspective of an agent.

Commitment: a directed obligation from a debtor to a creditor [8]. A commitment C(DEBTOR, CREDITOR, antecedent, consequent) denotes that the DEBTOR is obliged to the CREDITOR for bringing about the consequent if the antecedent holds.

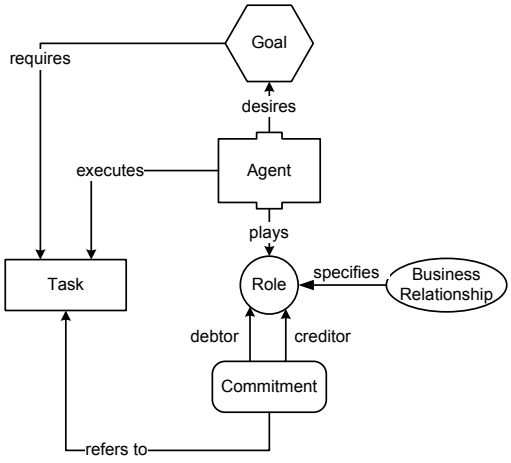


Fig. 1. Business metamodel [10]

A commitment can be in one of the following states: *inactive, active, detached, satisfied,* or *violated*. Before a commitment is created, it is *inactive*. In this state, the debtor has no obligation toward the creditor. A commitment may remain *inactive* forever if it is never created. Alternatively, if the debtor and the creditor negotiate and *create* the commitment, it becomes *active*. The negotiation may also create a *detached* commitment, that is, the antecedent of the commitment may hold upon creation. For an *active* commitment, *active timeout* may occur if neither its antecedent nor its consequent is brought about within certain time period. In that case, the commitment *expires*. If the antecedent of an *active* commitment is brought about, then the commitment is *detached*, or if its consequent is brought about, then it is *satisfied*. After a commitment is *satisfied*, if its antecedent is brought about, it remains *satisfied*. Similar to active timeout, for a *detached* commitment, *detached timeout* may occur if its consequent is not brought about, causing the commitment to be *violated*. Conversely, if the consequent of a *detached* commitment is brought about, it is *satisfied*.

Business relationship: a set of interrelated commitments among two or more roles that describe how business executes among the participating roles.

3 Business Patterns From PIPs

This section describes the approach to abstract business patterns from RosettaNet PIPs. It describes the business pattern for the Request Quote PIP 3A1 in detail, and for PIPs 3A4, 3A6, 3B3, and 3C6 in brief. Due to limited space, we present business patterns only for a subset of the PIPs.

A RosettaNet PIP specifies two-party interactions that contains two participant *roles*. Since the notion of a *role* in RosettaNet is similar to the concept of a *role* in our business model, a role from a PIP simply maps to a role in the business pattern. In a PIP, a busi-

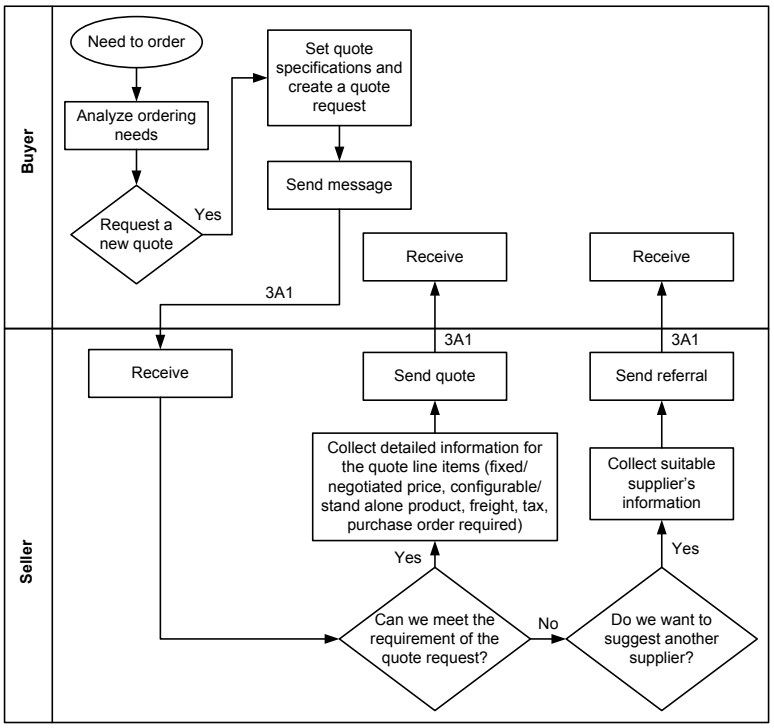


Fig. 2. Usage-Request Quote PIP 3A1 (verbatim from the RosettaNet specification [6])

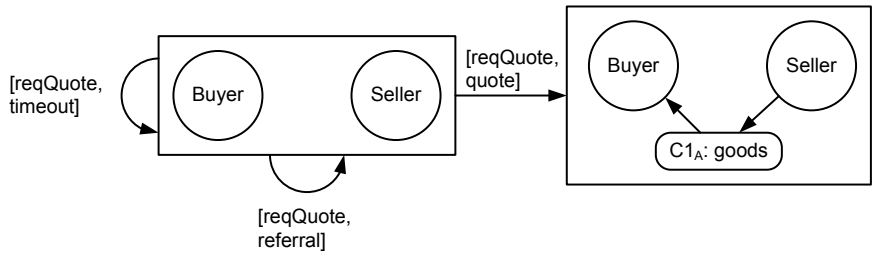
ness participant enacts one or more roles. In our business model, an agent represents a business participant and it enacts one or more roles.

In a PIP, a role enacting participant executes an *activity*. In our business model, the concept of a *task* models an activity, and a role enacting agent executes one or more tasks. Although a PIP specifies one activity per role, that activity can represent more than one real business activity. We model each of the real business activity as a task in the business pattern.

Unlike RosettaNet specification, the business pattern captures the business meaning of the tasks that the roles execute in terms of commitments. There can be multiple interpretations of the business meaning of a PIP. In this paper, we present one possible interpretation that naturally derives from the PIP's description. We now describe the business pattern for PIP 3A1.

Figure 2 shows the usage diagram from the RosettaNet description of the Request Quote PIP 3A1. It shows a buyer that desires to purchase certain goods. The buyer sends a request for a quote to a seller. If the seller can satisfy the requirements of the quote, it sends a quote to the buyer. Otherwise, the seller sends a referral of another seller to the buyer.

Figure 3 shows the business pattern for the request quote PIP 3A1. The pattern figures use the notation of Figure 1. The subscript on a commitment indicates its state:



C1 C(SELLER, BUYER, pay, goods)

Role	Tasks
Buyer	reqQuote, pay
Seller	quote, referral, goods

Fig. 3. Business pattern: Request Quote PIP 3A1

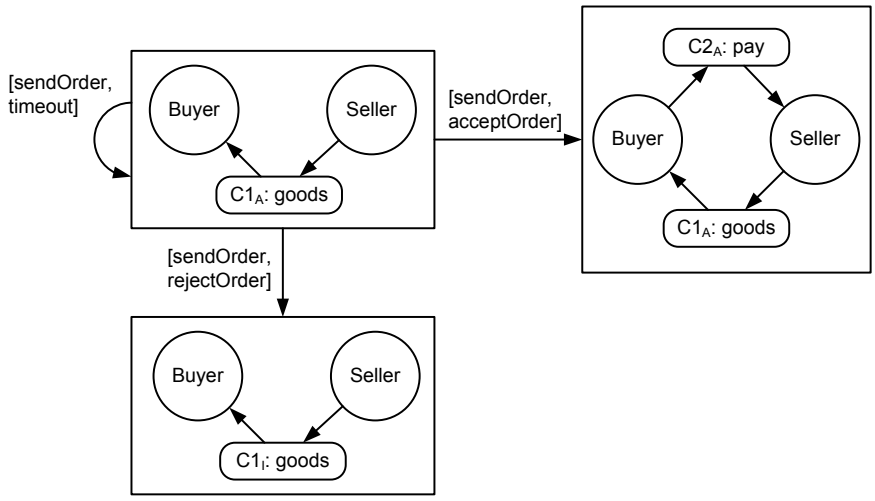
I for inactive, **A** for active, **D** for detached, and **S** for satisfied. The roles in Figure 3 are the same as the roles in the PIP. The BUYER role is capable of executing the tasks reqQuote, and pay. The task reqQuote is mapped from the *request quote* activity of the PIP. The SELLER role is capable of executing the tasks quote, referral, and goods. In the PIP, the SELLER executes *confirm quote* activity that means either sending a quote, or sending a referral. It derives the two seller tasks, quote and referral. If the seller sends a quote, the seller commits (C1) to the buyer to shipping the goods if the buyer pays. On the contrary, if the seller sends a referral, or if the seller fails to respond and a timeout occurs, then no commitment is created.

Similar to the business pattern for PIP 3A1, we derive the patterns for PIPs 3A4, 3A6, 3B2, and 3C6. Figure 4 shows the business pattern for the Request Purchase Order PIP 3A4. This PIP presumes that a commitment C1 exists from the seller to the buyer to sell certain goods for some price. The buyer sends a purchase order to the seller. If the seller accepts the order, the buyer becomes committed (C2) to the seller to pay if the seller ships the goods. On the contrary, if the seller rejects the order, then the seller cancels its existing commitment C1.

Figure 5 shows the business pattern for the Distribute Order Status PIP 3A6. After a buyer and a seller negotiate, and create commitments C1 and C2, the seller reports the progress of its task of shipping the goods to the buyer. If the seller reports that the goods are shipped, then the seller satisfies its commitment C1. This detaches the buyer's commitment C2 to pay the seller.

Figure 6 shows the business pattern for the Notify of Advance Shipment PIP 3B2. In this pattern, the seller ships the goods, and sends a ship notice to the buyer. By shipping the goods, the seller discharges its commitment C1.

Figure 7 shows the business pattern for the Notify of Remittance Advice PIP 3C6. The buyer sends a remittance advice to the seller, to pay for an order. This PIP presumes



C1 C(SELLER, BUYER, pay, goods)
C2 C(BUYER, SELLER, goods, pay)

Role	Tasks
Buyer	sendOrder, pay
Seller	acceptOrder, rejectOrder, goods

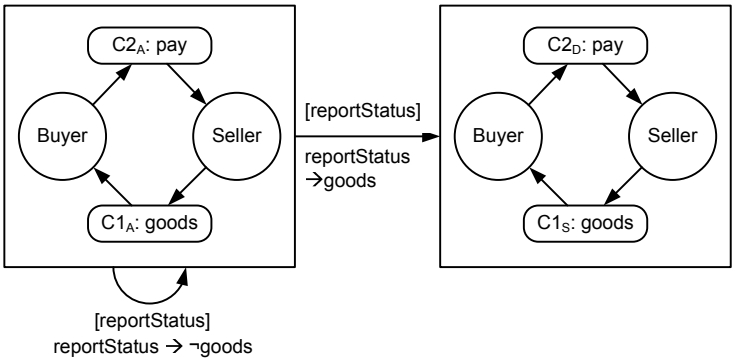
Fig. 4. Business pattern: Request Purchase Order PIP 3A4

that the buyer has a commitment C2 to pay the seller. The buyer discharges C2 by sending the remittance advice.

4 Evaluation: Aerospace Aftermarket Services

This section applies the RosettaNet business patterns to model the cross-organizational processes developed under the European Union CONTRACT project [11] in the domain of aerospace aftermarket services.

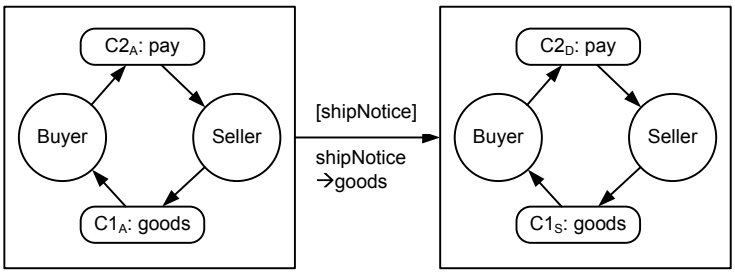
Figure 8 shows a high-level process flow of aerospace aftermarket services. The participants of this process are an airline operator, an aircraft engine manufacturer, and a parts manufacturer. The aircraft engine manufacturer provides engines to the airline operator. Additionally, the manufacturer services the engines to keep them operational. If a plane waits on the ground for an engine to be serviced, the manufacturer pays a penalty to the airline operator. As part of the agreement, the airline operator regularly provides engine health data to the manufacturer, and may proactively request the manufacturer to perform engine maintenance referred to as scheduled maintenance. The manufacturer analyzes the health data and informs the operator of required engine maintenance referred to as unscheduled maintenance. As part of servicing the engine, the manufacturer



C1 C(SELLER, BUYER, pay, goods)
C2 C(BUYER, SELLER, goods, pay)

Role	Tasks
Buyer	pay
Seller	reportStatus, goods

Fig. 5. Business pattern: Distribute Order Status PIP 3A6



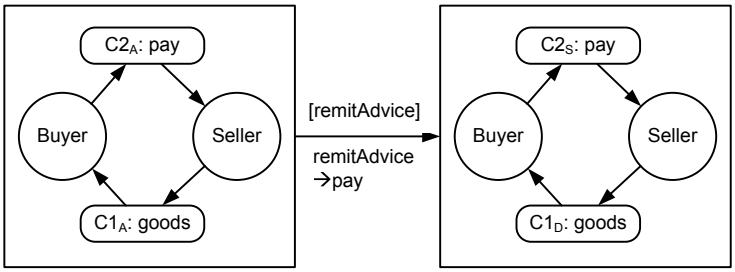
C1 C(SELLER, BUYER, pay, goods)
C2 C(BUYER, SELLER, goods, pay)

Role	Tasks
Buyer	pay
Seller	shipNotice, goods

Fig. 6. Business pattern: Notify of Advance Shipment PIP 3B2

can either refurbish or replace the engine. The manufacturer maintains a supply of engines by procuring parts from a parts manufacturer. Figure 9 shows how the RosettaNet patterns model the business arrangement between the manufacturer and the operator.

Using PIP 3A1, the operator requests a quote from the manufacturer to provide aircraft engine along with the maintenance service. The operator also requires the man-



C1 C(SELLER, BUYER, pay, goods)
C2 C(BUYER, SELLER, goods, pay)

Role	**Tasks**
Buyer	remitAdvice, pay
Seller	goods

Fig. 7. Business pattern: Notify of Remittance Advice PIP 3C6

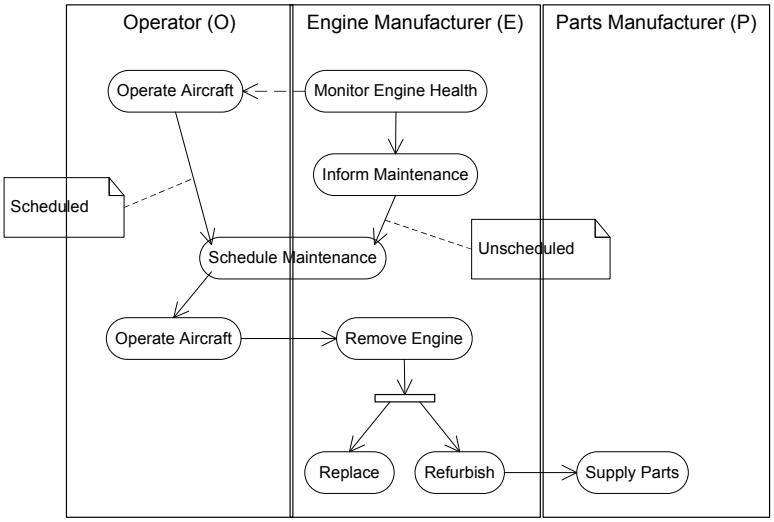
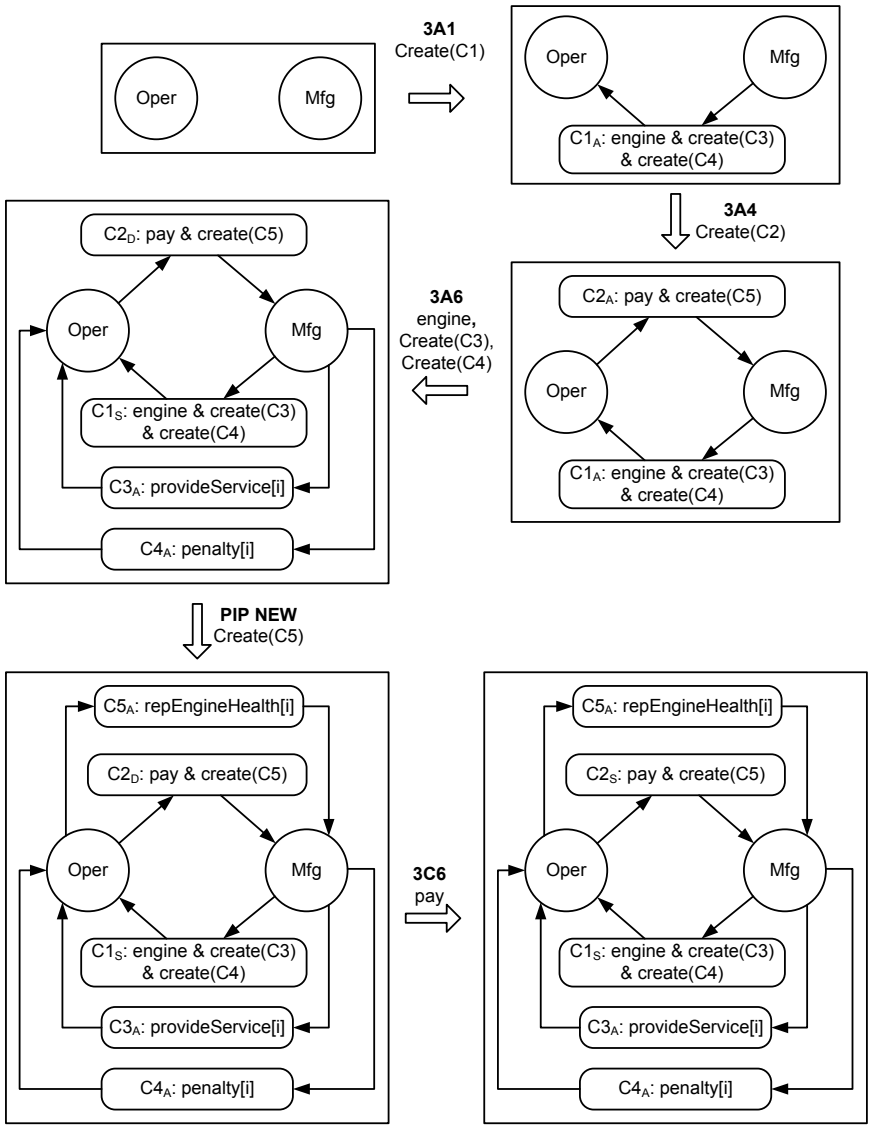


Fig. 8. A high-level model of the aerospace aftermarket process (verbatim from the Amoeba [2] paper, originally from CONTRACT project [11])

ufacturer to pay a penalty if an engine is down. By sending a quote, the manufacturer creates the commitment C1. In C1, the manufacturer commits to (a) provide the engine, (b) create the service commitment C3, and (c) create the commitment C4 to pay the penalty in case of the engine downtime. The manufacturer makes the commitment C1 with the condition that the operator pays for the offer, and creates the commitment $C5$ to provide engine health data if requested.



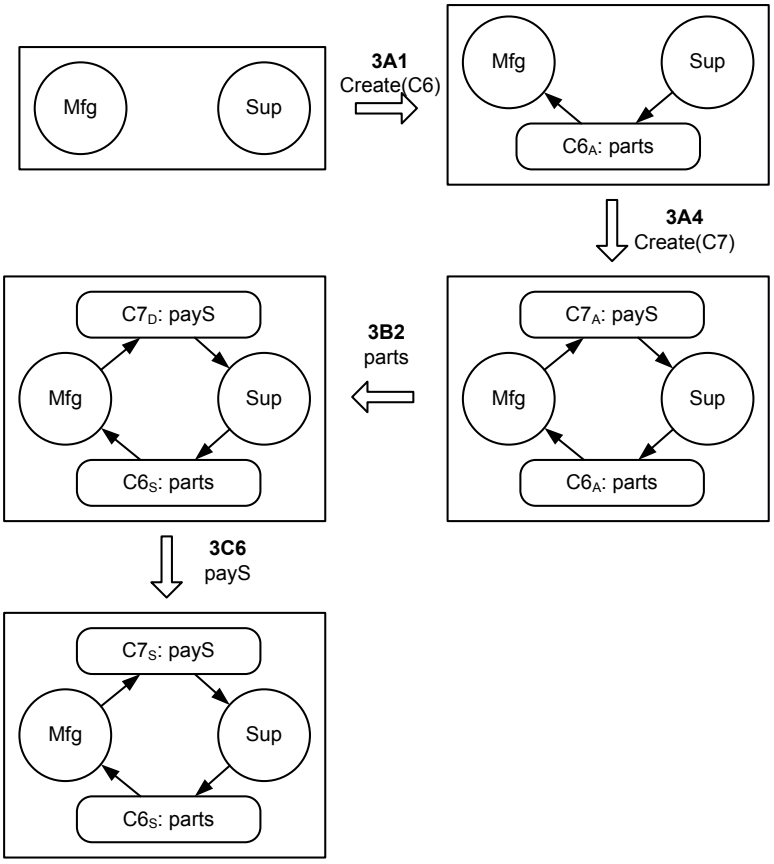
C1 C(MFG, OPER, pay ∧ create(C5), engine ∧ create(C3) ∧ create(C4))
C2 C(OPER, MFG, engine ∧ create(C3) ∧ create(C4), pay ∧ create(C5))
C3 C(MFG, OPER, requestService[i] ∧ ¬expired, provideService[i])
C4 C(MFG, OPER, engineDown[i] ∧ ¬expired, penalty[i])
C5 C(OPER, MFG, reqEngineHealth[i] ∧ ¬expired, reportEngineHealth[i])

Fig. 9. Operator and manufacturer interactions modeled using business patterns

If the operator finds the quote acceptable, it sends a purchase order to the manufacturer using PIP 3A4. By sending the purchase order, the operator commits to the

manufacturer to pay, and to create the commitment C5 if the manufacturer fulfills the offer.

Using PIP 3A6, the manufacturer informs the operator that it has shipped the engine, and created the commitments C3 and C4. The manufacturer satisfies its commitment C1, and it detaches the commitment C2. In the commitment C3, the manufacturer commits to the operator to provide service on the engine upon a request from the operator. There can be many instances of the service requests, and the manufacturer commits to provide the service for each of those requests so far as the request is made before an expiration date. As per the commitment C4, the manufacturer commits to the operator to pay a penalty in case of the engine downtime. Similar to the service commitment C3, the manufacturer commits to pay penalty for each instance of engine downtime that occurs before an expiration date.



C6 C(SUP, MFG, paySup, parts)
C7 C(MFG, SUP, parts, paySup)

Fig. 10. Manufacturer and supplier interactions modeled using business patterns

In the aerospace scenario, the operator (buyer) needs to notify the manufacturer (seller) that it created the commitment C5. The RosettaNet specification lacks a pattern for such a case in which a buyer notifies a seller of satisfying certain terms of a purchase. We assume a new PIP for this case. Using this PIP, the operator notifies the manufacturer that it created the commitment C5. As per C5, the operator commits to report the engine health to the manufacturer if the manufacturer requests for it.

The operator pays the manufacturer by sending a remittance advice using PIP 3C6. This satisfies the commitment C2.

The above interactions result in the business model with the active commitments C3, C4, and C5. These commitments form the basis of the business engagement between the operator and the manufacturer. For example, due to the commitment C3, an operator request for service, prior to the expiration date, creates a detached commitment from the manufacturer to the operator to service that instance of request. The commitment C3 remains active forever, but after the expiration date it never detaches. Therefore, the participants can purge the commitment after the expiration date has passed.

Figure 10 shows how the business patterns model the business engagement between the manufacturer, and the supplier. When the manufacturer desires to purchase certain parts, it sends a quote request to the supplier using PIP 3A1. If the supplier responds with a quote, the supplier commits (C6) to shipping the parts, if the manufacturer pays the supplier.

If the manufacturer finds the quote acceptable, it sends an order to the supplier using PIP 3A4. This creates the reverse commitment C7 from the manufacturer to the supplier to pay if the supplier ships the parts.

Eventually the supplier builds the parts, and ships them to the manufacturer. The supplier notifies the manufacturer of the shipment using PIP 3B2. This satisfies the supplier's commitment C6 to ship the parts, and detaches the manufacturer's commitment C7 to pay the supplier.

The manufacturer pays the supplier by sending a remittance advice using PIP 3C6. The manufacturer therefore satisfies its commitment C6 to pay the supplier.

5 Discussion

This paper presents an approach to abstract business patterns from RosettaNet PIPs. It applies these patterns to model a real-life use case. The paper highlights some of the shortcomings of the existing PIPs, and identifies the need for new PIPs. For example, in real-life purchase scenarios, a buyer not only pays a seller, but also satisfies terms and conditions of the sale. However, RosettaNet assumes that the buyer only pays the seller, and it has the PIP 3C6 for that purpose. A PIP that the buyer can use to notify the seller that it satisfied certain terms and conditions is lacking in RosettaNet.

Several RosettaNet PIPs merely allow the participants to exchange information. They do not create or manipulate commitments. Since our patterns give primacy to the commitments that the participant agents create and manipulate, they are not suitable for such PIPs.

5.1 Related Work

Nitto *et al.* [5] elicit that the systems in the open environment will need to be highly dynamic, and self-adaptive. Toward that end, they call out the need for natural, and high-level design abstractions for modeling such systems. Our research aims at developing such abstractions. This paper extracts business patterns from RosettaNet PIPs in terms of the high-level abstractions from our business metamodel.

There is a large body of research work that aims at creating a catalog of reusable patterns for business interactions. However, most of this work is based on low-level abstractions. For instance, Zdun *et al.* [12] propose a set of pattern primitives, and patterns for process-driven service oriented architecture. These patterns use low-level concepts of data and control flow to model the service interactions. They fail to specify the business relationships between the participants. In contrast, the patterns from this paper are founded on the business relationships defined in terms of commitments.

Singh *et al.* [9] propose a set of commitment oriented patterns for service interactions. They describe a pattern using a statechart that shows lifecycles of the commitments. In contrast, this paper describes patterns using a graphical language based on the business metamodel. Our graphical language is more natural as it emphasizes the roles and tasks in addition to showing commitments, and their states.

Kotinurmi *et al.* [4] and Haller *et al.* [3] incorporate semantics at the lower-level of data in RosettaNet PIPs. They develop an ontology using Web Service Modeling Language (WSML) for the PIP payloads and choreographies. In contrast, our work identifies business level meaning of the PIPs in terms of the commitments.

5.2 Future Directions

This work opens up several interesting directions. Of these, we are pursuing the development of formal techniques that involve formalizing the patterns and to verify compliance of low-level operational model with respect to a given business model. We expect also to develop a catalog of well-defined reusable patterns for business modeling. A systematic methodology to specify business models in high-level terms would also be crucial to the greater success of this effort.

References

1. BRG. The business motivation model, 2007.
2. Nirmit Desai, Amit K. Chopra, and Munindar P. Singh. Amoeba: A methodology for modeling and evolution of cross-organizational business processes. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 19(2):6:1–6:45, October 2009.
3. Armin Haller, Paavo Kotinurmi, Tomas Vitvar, and Eyal Oren. Handling heterogeneity in rosettanet messages. In *Symposium on Applied Computing*, pages 1368–1374, 2007.
4. Paavo Kotinurmi and Tomas Vitvar. Adding semantics to rosettanet specifications. In *Proceedings of the 15th international conference on World Wide Web*, pages 1059–1060, 2006.
5. Elisabetta Di Nitto, Carlo Ghezzi, Andreas Metzger, Mike P. Papazoglou, and Klaus Pohl. A journey to highly dynamic, self-adaptive service-based applications. *Automated Software Engineering*, 15(3-4):313–341, 2008.

6. RosettaNet. Overview: Clusters, Segments, and PIPs, 2008. www.rosettanet.org.
7. Nick Russell, Arthur, Wil M. P. van der Aalst, and Natalya Mulyar. Workflow control-flow patterns: A revised view. Technical report, BPMcenter.org, 2006.
8. Munindar P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7:97–113, 1999.
9. Munindar P. Singh, Amit K. Chopra, and Nirmit Desai. Commitment-based service-oriented architecture. *IEEE Computer*, 42(11):72–79, November 2009.
10. Pankaj R. Telang and Munindar P. Singh. Business modeling via commitments. In *Proceedings of the 7th AAMAS Workshop on Service-Oriented Computing: Agents, Semantics, and Engineering (SOCASE)*, volume 5907 of *LNCS*. Springer, 2009.
11. C. J. van Aart, Jiří Chábera, Martin Dehn, Michal Jakob, Kristof Nast-Kolb, J. L. C. F. Smulders, Patrick P. A. Storms, Camden Holt, and Malcolm Smith. Use case outline and requirements. Deliverable D6.1, IST CONTRACT Project, 2007. http://tinyurl.com/6adejz.
12. Uwe Zdun, Carsten Hentrich, and Schahram Dustdar. Modeling process-driven and service-oriented architectures using patterns and pattern primitives. *ACM Transactions on the Web*, 1(3), 2007.