Comma: A Commitment-Based Business Modeling Methodology and its Empirical Evaluation

Pankaj R. Telang and Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
{prtelang,singh}@ncsu.edu

ABSTRACT

We introduce Comma, a methodology for developing cross-organizational business models. Comma gives prime position to patterns of business relationships understood in terms of commitments. In this manner, it contrasts with traditional operational approaches such as RosettaNet that are commonly used in industry.

We report the results of a developer study comparing Comma with a methodology recommended by the RosettaNet Consortium. Ours is one of the only evaluations of an agent-oriented methodology that (1) involves developers other than the proposing researchers and (2) compares against a traditional nonagent approach.

We found that Comma yields improved model quality, a greater focus in relative effort on the more important aspects of modeling, and a general reduction in total time despite yielding more comprehensive models. Certain anomalies in effort expended point toward the need for improved tooling.

Categories and Subject Descriptors

H.1.0 [**Information Systems**]: Models and Principles—*General*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Design, Experimentation

Keywords

Commitments, Business modeling, Methodology

1. INTRODUCTION

Real-world organizations seldom operate in isolation. To stay competitive, organizations develop deep expertise in core business functions, and outsource the rest to business partners. This results in a network of organizations with complex business relationships. Existing approaches for business modeling are of two broad types. The low-level approaches use concepts such as message ordering and control flow, and yield highly rigid models. The high-level approaches use concepts such as goals and values, and cannot be easily operationalized. Recently, researchers have begun to use social commitments for business modeling, e.g., [4], since they lead to flexible yet operationalizable models.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

We introduce Comma, a novel commitment-based business modeling methodology, which builds on a recent business metamodel [19]. Unlike traditional approaches, Comma gives prominence to patterns of business relationships. The motivation for developing abstractions such as commitments is that they would facilitate the engineering of superior solutions by helping build richer models of interaction. This is the main claim that we investigate here along with associated claims of ease of use and efficiency.

Two shortcomings of previous approaches are that, first, they do not adequately describe how to put concepts such as commitments into modeling practice, especially for the benefit of practitioners who are not multiagent systems specialists. And, second, previous approaches have not empirically evaluated their benefits in a controlled study, involving participants other than the authors. The same shortcomings, especially the second, might be said to apply on AOSE research broadly.

Contributions and Organization

The main contributions of this paper are the Comma methodology and a developer study comparatively evaluating it with respect to RosettaNet [14], a well-known traditional approach for cross-organizational processes. Our results confirm the relative effectiveness of Comma for the quality of modeling cross-organizational processes, and some benefits in ease of modeling and time expended. Further, the results yield insights for future improvements.

The rest of the paper is organized as follows. Section 2 describes the necessary background. Section 3 describes the Comma methodology. Section 4 outlines the design of the study, and Section 5 describes the study results. Section 6 discusses related work, and Section 7 concludes the paper with a discussion of future directions.

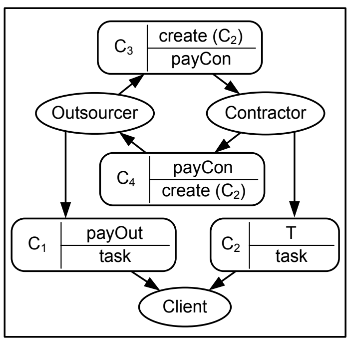
2. BACKGROUND

RosettaNet, a consortium of over 500 organizations, is a leading industry effort that develops standards for Business-to-Business integration that support business transactions worth billions of dollars. In RosettaNet, a Partner Interface Process (PIP) specifies a two-party interaction for a specific business intent. The PIPs are organized in a two-level hierarchy of *cluster* and *segment*. For example, Request Purchase Order PIP 3A4 is from Cluster 3 (Order Management) and Segment A (Quote and Order Entry). Using 3A4, a buyer sends a purchase order to a seller. Most PIPs define a two-party interaction involving a request and a response message. A modeler prepares a list of the necessary PIPs as the *RosettaNet model* of a business scenario. Next the modeler designs what we term *RosettaNet MSCs*: message sequence charts (MSCs) whose messages are derived from the PIPs.

We now describe some relevant concepts from Telang and Singh's [19] business metamodel. An *agent* models a real-world organization. The agent can play one or more roles in a business relationship. A *role* abstracts over the agents, and specifies, in a templatic form, the commitments that an agent adopting the role must participate in. A *task* is a business activity that an agent performs.

A *commitment* C(DEBTOR, CREDITOR, antecedent, consequent) means that the DEBTOR commits to the CREDITOR to bring about the consequent if the antecedent holds. The antecedent and the consequent are logical expressions over the tasks. When the antecedent of a commitment holds, the commitment detaches, and the debtor becomes unconditionally committed to the creditor to bring about the consequent. Regardless of the antecedent, if the debtor brings about the consequent, the commitment is satisfied [15]. For example, $C =$ C(BUYER, SELLER, goods, pay) means that the buyer commits to the seller to paying if the seller ships the goods. C detaches if the seller ships the goods, and satisfies if the buyer pays regardless of when the seller ships the goods. Singh [16] explains commitments further.

Telang and Singh [19] define several business (modeling) patterns, of which our study used *commercial transaction*, and *outsourcing* patterns. We briefly describe the *outsourcing pattern*, and refer the reader to [19] for further details.



C_1	C(OUTSOURCER, CLIENT, payOut, task)
C_2	C(CONTRACTOR, CLIENT, $\top$, task)
C_3	C(OUTSOURCER, CONTRACTOR, create(C2), payCon)
C_4	C(CONTRACTOR, OUTSOURCER, payCon, create(C2))

Figure 1: Outsourcing business pattern [19].

Figure 1 shows the *outsourcing pattern* in Telang and Singh's notation. An oval represents a role; the label in the oval is the role name. A rounded rectangle represents a commitment. The rectangle shows the commitment name in the left-hand side, and in the right-hand side it shows the antecedent on the top and the consequent on the bottom. Two directed edges connect a commitment to roles: from the debtor to the commitment and from the commitment to the creditor. In the outsourcing pattern, an outsourcer delegates a task to a subcontractor. Here, C_1 is the original commitment from the outsourcer to a client to execute a task if the client pays the outsourcer (payOut). C_2 is the outsourced commitment from the contractor to the client to execute the same task. The antecedent of C_2 is true ($\top$), which means that it is unconditional. C_3 and C_4 are the commitments in which the outsourcer and the contractor commit to pay (payCon) and to create C_2, respectively.

3. COMMA

For each business pattern, such as those proposed by Telang and Singh [19], we develop a set of generalized (templatic) message sequence charts that operationalize that pattern.

Figure 2 shows the MSCs for the outsourcing pattern using UML 2.0 sequence diagram [11] operators OPT(ion) and ALT(ernative).

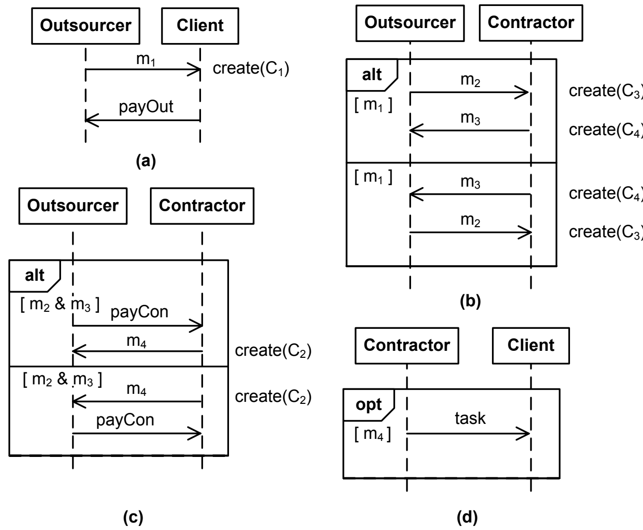


Figure 2: Message sequence charts for outsourcing.

We go beyond UML in labeling each message with its meaning. A message labeled with a proposition, usually part of the antecedent or consequent of some commitment, simply brings about that proposition. A message labeled m_i for some i means an operation on some commitment (such as its creation), which we annotate on the side. In Figure 2(a), the outsourcer sends m_1 to the client, which creates commitment C_1. The client sends payOut to the outsourcer upon receiving m_1, which detaches C_1 since it is C_1's antecedent. In Figure 2(b), after receiving m_1, the outsourcer sends m_2 to the contractor, and after receiving m_2 the contractor sends m_3 to the outsourcer. Alternatively, the contractor first sends m_3 to the outsourcer, and after receiving m_3, the outsourcer sends m_2 to the contractor. m_2 creates C_3 and m_3 creates C_4. In Figure 2(c), after m_2 and m_3 are exchanged, the outsourcer sends payCon to the contractor and the contractor sends m_4 to the outsourcer in either order. Now payCon satisfies C_3 and detaches C_4; and, m_4 creates C_2 and satisfies C_4. In Figure 2(d), after m_4 is exchanged, the contractor sends task (message) to the client. This satisfies C_1 and C_2 since task is their consequent. As part of creating a model, a modeler substitutes the message labels m_i with domain-specific terms.

The Comma methodology begins from an informally described real-life cross-organizational scenario and produces formal business and operational models. Table 1 summarizes Comma.

Step 1 A *subscenario* is a fragment of the given scenario. From the given scenario description, extract subscenarios such that each match a pattern from the Comma pattern library.

Step 2 For each subscenario, identify its roles. A subscenario usually describes participants using a combination of generic terms (e.g., Company, Partner, and Organization) and specific names (e.g., FedEx). This step involves creating roles based on business function (e.g., Shipper) that remove any ambiguity, such as if Partner and Organization refer to the same entity.

Step 3 For each subscenario, identify business tasks (e.g., goods and payment) that a role executes. A scenario typically specifies the tasks as actions executed by the participants.

Step 4 From the Comma pattern library, introduce into the business model a pattern corresponding to each subscenario. Rename the pattern characters with the roles from Step 2, and introduce the tasks from Step 3 as the antecedents and con-

Table 1: Comma methodology steps.

Step	Description	Input	Output
1	Extract subscenarios corresponding to Comma patterns	Real-life cross-organizational scenario	Subscenarios
2	Identify roles from each subscenario	Subscenario	Roles
3	Identify business tasks from each subscenario	Subscenario	Tasks
4	Introduce a Comma pattern for each subscenario	Comma pattern, subscenario, roles, tasks	Business model
5	Introduce MSCs for each Comma pattern	Comma pattern MSCs, subscenario, roles, tasks	Operational model

sequents of the appropriate commitments. The patterns compose naturally when the same roles are referenced by more than one pattern.

Step 5 For each Comma pattern, introduce its MSC into the operational model. Rename the roles and messages in the MSCs to align them with those determined in Steps 2 and 3. Customize the MSCs to capture any subscenario-specific operational details, such as additional messages, guards, and loops.

4. DESIGN OF THE STUDY

Our study used an initial scenario based on real-life cross-organizational business processes, inspired by the Oracle Quote-To-Cash (QTC) process [12, 19], and two modifications of the scenario.

$\mathcal{S}_i$, the initial scenario, involves MedEq, a company that sells medical equipment. MedEq designs the equipment in house, and out-sources manufacturing to two contract manufacturers, FlexMan and SoleMan, and shipping to two shippers, FedUp and UpFed. To purchase the equipment, a customer submits its requirements to MedEq. MedEq analyzes the requirements, and creates a proposal containing the equipment details, and a quoted price. The customer may accept the proposal or negotiate for a better price. There can be up to two iterations between MedEq and the customer before they either agree upon the price, or abort the transaction. If MedEq and a customer reach an agreement, the customer proceeds to placing an order and specifying the equipment, shipping address, contact information, and payment information. Upon receiving the order, MedEq validates the order. MedEq accepts the order if it is valid and rejects it otherwise. MedEq maintains warehouses in which it stocks the equipment. In case the ordered equipment is in stock, MedEq requests a shipper to ship the equipment to the customer. MedEq pays the shipping charges to the shipper.

If the equipment necessary to fulfill an order is not in stock, MedEq places a stock replenishment order with a contract manufacturer. The contract manufacturer employs a shipper to ship the equipment to MedEq's warehouse. MedEq pays the contract manufacturer for the equipment. Once the equipment is in stock, MedEq fulfills the customer's order.

$\mathcal{S}_f$, the first modification, adds a new participant, a value-added reseller, MedRes. MedRes sells, installs, and supports (i.e., services) medical equipment. The customer now places its order with MedRes, who orders the equipment from MedEq and provides the installation and support itself. The customer pays MedRes, and MedRes pays MedEq. MedRes supports the equipment as needed. The rest of the scenario remains unchanged.

$\mathcal{S}_s$, the second modification, removes the contract manufacturers SoleMan and FlexMan from the original scenario. The rest of the scenario is unchanged.

4.1 Study Solution

Figure 3 shows the solution Comma model for the initial scenario, $\mathcal{S}_i$. For brevity, we present only the final Comma model and

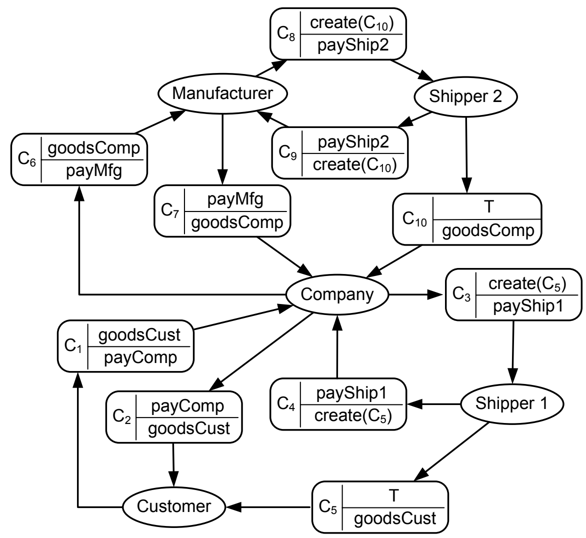


Figure 3: Comma model for $\mathcal{S}_i$.

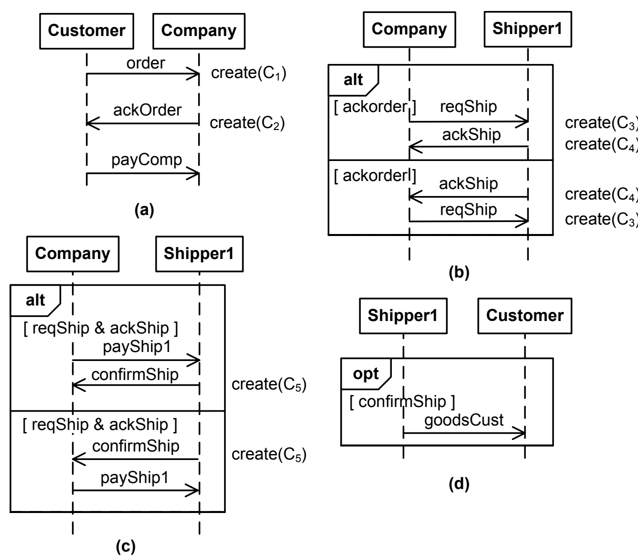


Figure 4: Example Comma MSCs for $\mathcal{S}_i$.

omit the outputs of the intermediate methodology steps. The model is composed from the commercial transaction and the outsourcing patterns. For example, the commercial transaction pattern captures MedEq (Company) and the customer agreeing to exchange medical equipment for certain price. The model commitments C_1 and C_2 correspond to this pattern: in C_1, the customer commits to paying the company (payComp) if the company provides the equipment (goodsCust), and in C_2, the company commits to providing the equipment if the customer pays. The outsourcing pattern models MedEq employing a shipper (Shipper 1) to ship the medical equipment to the customer. The model commitments C_2, C_3, C_4, and C_5 correspond to this pattern: C_2 is the original commitment, C_5 is the outsourced commitment, and C_3 and C_4 are the commitments in which the company and the shipper commit to paying and to creating C_5, respectively. Figure 4 shows four of the ten MSCs for the initial scenario, $\mathcal{S}_i$, developed using Comma. These MSCs correspond to MedEq outsourcing the shipping to a shipper. We omit further description of these MSCs since Section 3 describes the outsourcing MSCs in detail.

Table 2: RosettaNet model PIPs for $\mathcal{S}_i$.

PIP	Name (shortened)	Subscenario
3A1	Request quote	Customer, MedEq negotiate
3A4	Purchase order	Customer orders from MedEq
3B12	Request shipping	MedEq ships to Customer
3C3	Notify of invoice	Shipper invoices MedEq, MedEq invoices customer, shipper invoices manufacturer, manufacturer invoices MedEq
3C4	Reject invoice	MedEq, customer, or manufacturer reject invoice
3C6	Remittance advice	MedEq pays the shipper, customer pays MedEq, manufacturer pays shipper, MedEq pays manufacturer
7B5	Manufacturing order	MedEq orders from manufacturer
3B12	Request shipping	Manufacturer ships to MedEq

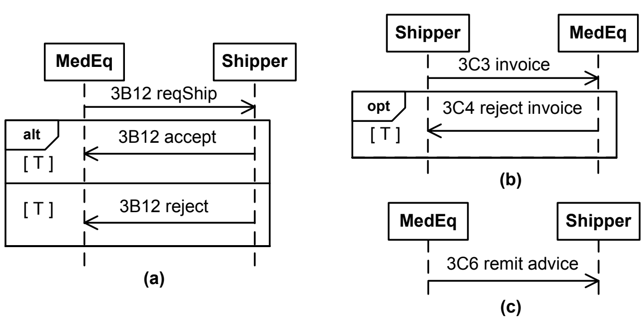


Figure 5: Example RosettaNet MSCs for $\mathcal{S}_i$.

Table 2 shows the RosettaNet model PIPs for the initial scenario, $\mathcal{S}_i$. For example, the customer uses PIP 3A1 to request a quote from MedEq. Figure 5 shows three of the thirteen MSCs for the initial scenario, $\mathcal{S}_i$, developed using RosettaNet. Figure 5(a) is the MSC for PIP 3B12 in which MedEq requests the shipper to ship the equipment to the customer. The shipper either accepts or rejects the request. The shipper invoices MedEq using PIP 3C3 in Figure 5(b). MedEq may reject the invoice using PIP 3C4. In Figure 5(c), MedEq notifies the shipper of remittance advice using PIP 3C6.

4.2 Study Mechanics and Threat Mitigation

We conducted a developer study with 34 subjects (graduate computer science students). Three exercises, corresponding to the three scenarios, $\mathcal{S}_i$, $\mathcal{S}_f$, and $\mathcal{S}_s$, comprised the study. The study used a *between-subject* experimental design [9]. For each exercise, the study divided the subjects into two groups who applied different methodologies to model the same scenario. We carefully designed the study to mitigate the well-known threats [9] to its validity.

To mitigate the threat of skill differences between the participants, prior to the exercises, we surveyed the study subjects to gather information on their educational background, and experience in process modeling and software engineering. We then divided the participants into two groups, *A* and *B*, of approximately equal skill levels. The first exercise compared groups *A* and *B*, and the subsequent exercises split and merged the same groups. For the first exercise, the subjects in groups *A* and *B* developed a model and MSCs for $\mathcal{S}_i$ using RosettaNet and Comma, respectively.

For the second and third exercises, a primary threat was the learning effect, because after the first exercise, subjects would be familiar with the methodology they used. To mitigate this threat, we divided each group into two subgroups of equal size and combined a subgroup from each group to form new groups *A'B'* and *A"B"*. A secondary threat was variance in the initial models developed by different subjects and their lack of familiarity with models developed by others. To mitigate this threat, we developed $\mathcal{C}$ and $\mathcal{R}$, respectively, Comma and RosettaNet model and MSCs for the initial scenario $\mathcal{S}_i$.

In the second exercise, group *A'B'* began from $\mathcal{C}$ and applied Comma, and group *A"B"* began from $\mathcal{R}$ and applied RosettaNet, both to account for $\mathcal{S}_f$.

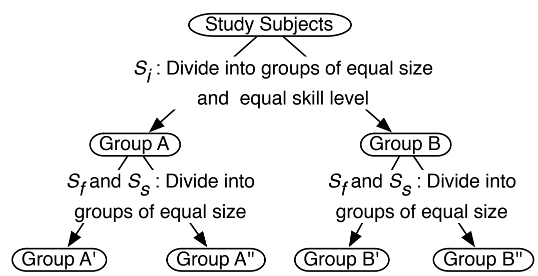


Figure 6: Our approach of grouping the subjects.

In the third exercise, we swapped the two groups. Group *A'B'* reviewed $\mathcal{R}$ and applied RosettaNet, and group *A"B"* reviewed $\mathcal{C}$ and applied Comma, both to account for $\mathcal{S}_s$.

Figure 6 summarizes how the study divided the subjects into groups, and Table 3 summarizes the exercises.

The subjects self-reported the time and difficulty for each methodology in a *work log*. To mitigate the threat of a subject forgetting to report relevant information, we required each subject to submit his or her work log three days a week, regardless of the effort they spent in that period.

4.3 Dependent Variables

This section describes the dependent variables of the study that we use to compare Comma and RosettaNet.

Quality of the models, assessed by experts, using the measures of Table 4. (A higher value is better for each.)

Difficulty in completing a methodology step as (subjectively) reported by a subject. Difficulty ranges over extremely easy, easy, neutral, difficult, and extremely difficult. Subjects reported the difficulty in a work log; we calculate the percentage of responses for each difficulty level. In most reports, we combine best two as *easy* and the worst two as *difficult*.

Table 3: Study exercises.

Exercise	Group A		Group B	
	Group A'	Group A"	Group B'	Group B"
1	Develop RosettaNet model and MSCs for $\mathcal{S}_i$		Develop Comma model and MSCs for $\mathcal{S}_i$	
2	Modify $\mathcal{C}$ to model $\mathcal{S}_f$	Modify $\mathcal{R}$ to model $\mathcal{S}_f$	Modify $\mathcal{C}$ to model $\mathcal{S}_f$	Modify $\mathcal{R}$ to model $\mathcal{S}_f$
3	Modify $\mathcal{R}$ to model $\mathcal{S}_s$	Modify $\mathcal{C}$ to model $\mathcal{S}_s$	Modify $\mathcal{R}$ to model $\mathcal{S}_s$	Modify $\mathcal{C}$ to model $\mathcal{S}_s$

Table 4: Quality measures, as judged by experts.

Measure	Captures a methodology's
Model Coverage. Percentage of models that fully cover the problem scenario	Completeness in modeling a scenario
Model Precision. Percentage of models that include no aspects unrelated to the problem scenario	Effectiveness in avoiding bloated models
MSC Structure. Percentage of MSCs with correct and complete guards	Soundness: fewer errors in outcomes
MSC Flexibility. Average number of ALT(ernative) blocks per MSC	Support for participants' flexibility
MSC Abstraction. Percentage of MSCs that use a role, not an agent, name	Support for reusability of models

Time taken to complete a methodology step as reported by a subject: a continuous variable in the unit of hours. Subjects reported the time they spent in a work log; we summed up the time for each subject.

5. STUDY RESULTS

This section describes the key findings from the study.

5.1 Quality

Figures 7 and 9 show the quality measurements of the two methodologies from the initial exercise $\mathcal{S}_i$.

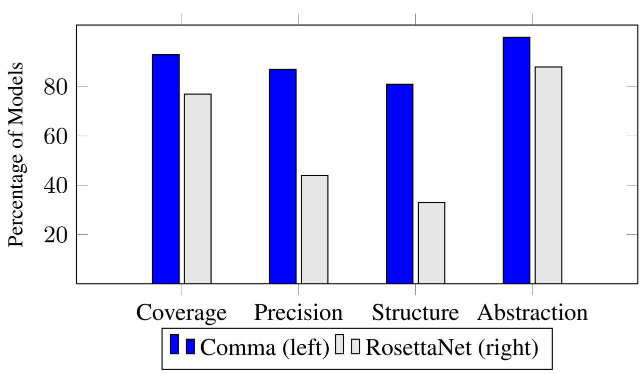


Figure 7: Quality of the models of $\mathcal{S}_i$.

Observation 1*:* As Figure 7 shows, both model coverage and model precision are superior for Comma (93% and 87%, respectively) than for RosettaNet (77% and 44%, respectively).

Observation 1 suggests that Comma is more effective than RosettaNet in creating complete and precise models. We credit this to the systematic nature of Comma and the fact that it focuses attention on the relevant commitments and MSCs. On the contrary, the RosettaNet models tend to contain several superfluous PIPs.

Observation 2*:* As Figure 7 shows, the percentage of models in which MSCs do not miss any necessary guards is higher for Comma (81%) than for RosettaNet (33%).

Since RosettaNet focuses on individual interactions in the form of PIPs, a modeler often loses an overall perspective on the scenario. The modeler develops an MSC for each PIP, but fails to relate the MSCs to each other via appropriate guards. In contrast, Comma forces a modeler to think in terms of the commitment life cycle. For example, a message that satisfies a commitment should be preceded by a message that creates the commitment.

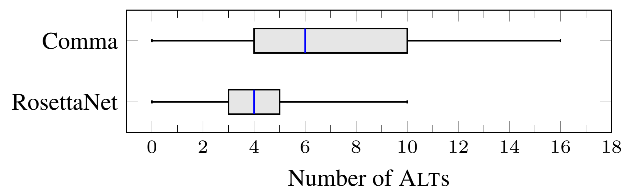


Figure 9: Flexibility of the MSCs produced for $\mathcal{S}_i$.

Observation 3*:* As Figure 9 shows, Comma MSCs use a higher median number of ALTs per model (six) than RosettaNet MSCs (four).

RosettaNet tends to lead to rigid MSCs, i.e., those with only a few alternative paths. The MSCs included with Comma patterns promote flexibility, which is inherent in the commitment-based approach. As a telling example, almost all subjects developed RosettaNet MSCs in which the Customer pays MedEq strictly after MedEq ships the ordered equipment. In contrast, many subjects developed Comma MSCs in which the Customer may pay MedEq either before or after MedEq ships the ordered equipment, a situation that has been discussed since the earliest works on commitment protocols [21].

Observation 4*:* The percentage of models in which MSCs use a role name instead of a participant name is higher for Comma (100%) than for RosettaNet (88%).

Observation 4 supports the idea that Comma emphasizes role abstraction and more naturally yields reusable MSCs.

Since the second and the third exercises began from the models that we provided, the resulting models are of higher quality, and without perceptible difference between the two methodologies. Therefore, we present quality results only for the first exercise.

5.2 Difficulty

Figure 8 shows the percentage of work log responses corresponding to each difficulty level for the three exercises.

Observation 5*:* In $\mathcal{S}_i$, the percentage of easy responses is smaller for RosettaNet (21.6%) than for Comma (27.5%), and the percentage of difficult responses is higher for RosettaNet (28.3%) than for Comma (23.7%).

Observation 5 suggests that Comma modeling is relatively easier as compared to RosettaNet modeling.

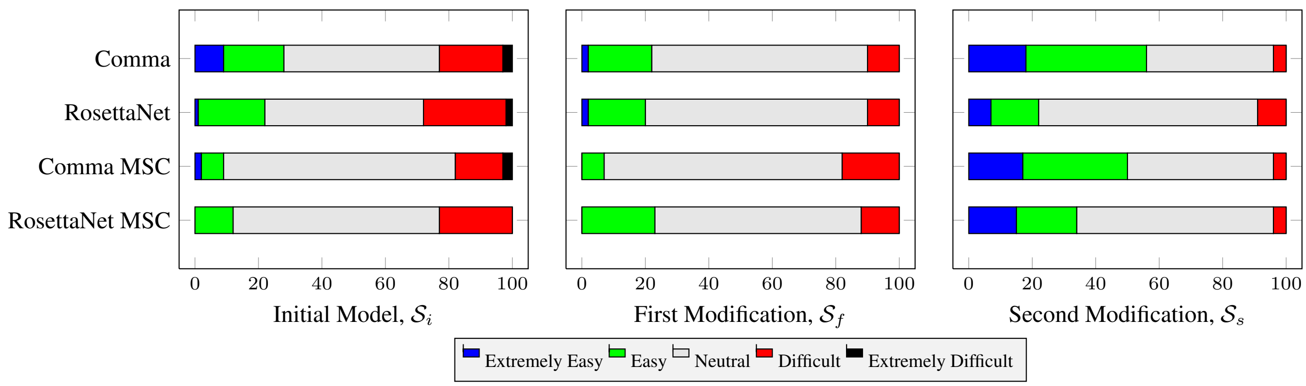


Figure 8: Difficulty of modeling, as percentage of responses by the subjects.

To identify the underlying cause of the extreme difficulty reports about Comma modeling, we analyzed the reported difficulty for each step. The analysis revealed that Comma Step 4, composing patterns to create a model, significantly contributes to the difficulty. This finding indicates the need for simplifying Step 4.

Observation 6*:* In $\mathcal{S}_i$, the percentage of difficult responses in developing MSCs using Comma (18.3%) is smaller than using RosettaNet (23.0%). However, the percentage of extremely difficult responses to developing MSCs using Comma (3.3%) is larger than using RosettaNet (0%).

Observation 6 is mixed. Although Comma appears to have been easier than RosettaNet overall, the number of subjects who found Comma extremely difficult was greater than the corresponding number for RosettaNet. This emphasizes the need for simplifying Comma Step 5, developing MSCs. A modeling tool, already under development, can assist a modeler by creating a base MSC model using the pattern MSCs.

Observation 7*:* Comma modeling has 0% extremely difficult responses, and 9.9% somewhat difficult responses in $\mathcal{S}_f$, as compared to 2.8% extremely difficult responses, and 20.9% percent somewhat difficult responses in $\mathcal{S}_i$.

We explain Observation 7 based on two factors. First, some of the subjects gained experience modeling using Comma in the initial exercise. Second, the subjects started the first modification $\mathcal{S}_f$ from a solution that we provided.

Relative to $\mathcal{S}_i$ and $\mathcal{S}_f$, $\mathcal{S}_s$ has increased responses with lower difficulty levels. This is partially due to the learning that the subjects gained from the first two exercises, and partially since $\mathcal{S}_s$ was a relatively easy exercise.

Observation 8*:* In $\mathcal{S}_s$, the percentages of easy responses for modifying the Comma model (56.2%) and MSCs (50%) are higher than for modifying the RosettaNet model (22.1%) and MSCs (33.3%).

Observation 8 suggests that with some experience, Comma becomes simpler than RosettaNet.

5.3 Time

Figure 10 shows boxplots of the time taken by the subjects to develop Comma and RosettaNet models and MSCs in the three exercises. Throughout, we remove each outlier: a point that is greater than the third quartile or smaller than the first quartile by 1.5 times the interquartile range—i.e., the difference between the third and first quartiles.

Observation 9*:* In $\mathcal{S}_i$, the median time to develop a model is smaller for Comma (6.7 hours) than for RosettaNet (10 hours).

Observation 9 suggests that Comma is more efficient than RosettaNet for creating a business model.

Observation 10*:* In $\mathcal{S}_i$, the median time to develop MSCs is somewhat greater for Comma (6 hours) than for RosettaNet (5.5 hours).

Although Comma appears less efficient than RosettaNet, as Section 5.1 shows, the MSCs produced from Comma are of higher quality than those produced from RosettaNet.

Observation 11*:* In $\mathcal{S}_i$, the spreads of the times for developing the model and MSCs are smaller for Comma than for RosettaNet.

Observation 11 indicates that Comma is more predictable than RosettaNet in terms of development effort.

Observation 12*:* Using Comma, the median modeling time for the first modification $\mathcal{S}_f$ (6.6 hours) is about the same as that for the initial exercise $\mathcal{S}_i$ (6.7 hours).

Observation 12 is surprising to us. We expected the Comma modeling time for $\mathcal{S}_f$ to be smaller than for $\mathcal{S}_i$. We attribute this result to a couple of key factors. First, the subjects needed time to comprehend the solutions we provided. Second, the subjects followed the same steps for modifying the model as the steps they followed for creating the model in the initial exercise. Comma should be improved to guide modelers in modifying existing business models.

Observation 13*:* In $\mathcal{S}_f$, the median modeling time is higher for Comma (6.6 hours) than for RosettaNet (4 hours).

Observation 13 conflicts with Observation 9 from the initial exercise $\mathcal{S}_i$. A primary reason for this result is the difference in the nature of the artifacts involved. A RosettaNet model is expressed as a textual list of PIPs, modifying which is easy. A Comma model is expressed as a graph of business relationships, modifying which is time consuming. Indeed, since we did not provide a Comma modeling tool, subjects expended considerable effort in developing the graphical models using drawing tools such as Visio.

Observation 14*:* In $\mathcal{S}_f$, the median time to modify MSCs is lower for Comma (1 hour) than for RosettaNet (2.3 hours).

Observation 14 suggests that the Comma methodology is more efficient as compared to the RosettaNet methodology for developing MSCs. Note that this result is an improvement over Observation 10 from the initial exercise $\mathcal{S}_i$ in favor of Comma, indicating the benefit of learning.

Observation 15*:* In $\mathcal{S}_s$, the median times to modify the Comma model (2.75 hours) and MSCs (0.75 hours) are slightly smaller than the median times to modify the RosettaNet model (3 hours) and MSCs (1 hour), respectively.

Observation 15 suggests that Comma is slightly more efficient

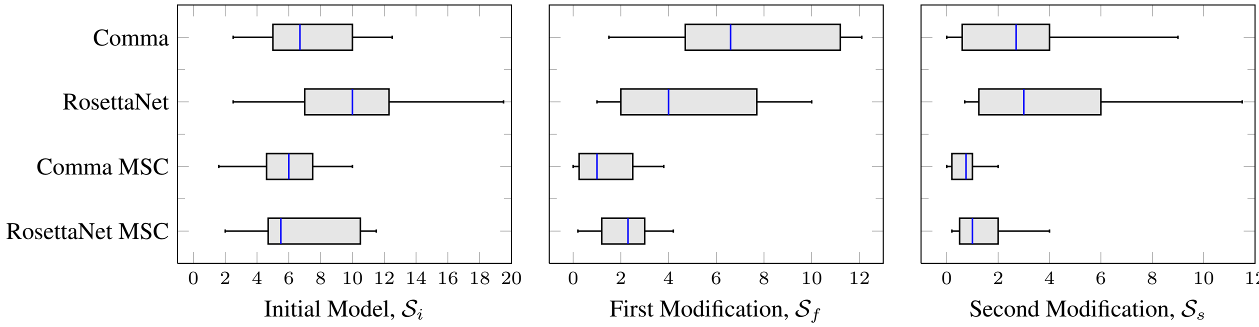


Figure 10: Time in hours expended in creating models, as reported by subjects.

Table 5: Hypothesis testing for model and MSC development times.

ID	**Time for Exercise**	**Comma Mean (μ_c)**	**RosettaNet Mean (μ_r)**	**Alternative Hypothesis**	**Null Hypothesis $[\mu_c = \mu_r]$ p-value**	**Accepted at p-value of 5%?**
H_1	$\mathcal{S}_i$-Model	7.19	10.05	$\mu_c < \mu_r$	0.046	×
H_2	$\mathcal{S}_i$-MSC	6.22	6.73	$\mu_c < \mu_r$	0.610	✓
H_3	$\mathcal{S}_f$-Model	7.59	4.84	$\mu_c > \mu_r$	0.026	×
H_4	$\mathcal{S}_f$-MSC	1.42	2.26	$\mu_c < \mu_r$	0.062	✓
H_5	$\mathcal{S}_s$-Model	2.77	3.74	$\mu_c < \mu_r$	0.290	✓
H_6	$\mathcal{S}_s$-MSC	0.70	1.29	$\mu_c < \mu_r$	0.053	✓

than RosettaNet for modifying models. This agrees with Observation 9 from the initial exercise $\mathcal{S}_i$.

Observation 16*:* In $\mathcal{S}_s$, the spreads of times taken in modifying the model and MSCs are smaller for Comma than for RosettaNet.

Observation 16 agrees with Observation 11, and reconfirms that Comma is more predictable than RosettaNet.

The above observations are from the descriptive statistics summarized by the box plots. We now present the results of formal hypothesis testing that checks if the difference between the timings of the two methodologies is statistically significant. Table 5 summarizes the hypotheses and the outcome of the *independent samples t-test* for each of them. H_1, H_3, and H_5 test the statistical significance of the difference between the modeling time of the two methodologies in $\mathcal{S}_i$, $\mathcal{S}_f$, and $\mathcal{S}_s$, respectively. H_2, H_4, and H_6 test the statistical significance of the difference between the MSC development time of the two methodologies in $\mathcal{S}_i$, $\mathcal{S}_f$, and $\mathcal{S}_s$, respectively. In H_1, the alternative hypothesis is $\mu_p < \mu_r$, that is, the mean time to develop the Comma model μ_p is less than the mean time to develop the RosettaNet model μ_r. The corresponding null hypothesis is $\mu_p = \mu_r$, that is, the mean time to develop the Comma model is the same as the mean time to develop the RosettaNet model. The t-test rejects the null hypothesis with *p value* of 0.046 at the 0.05 level of significance. This confirms that Comma is more efficient than RosettaNet in $\mathcal{S}_i$, which agrees with Observation 9.

The t-test rejects the null hypothesis in H_3. This indicates that RosettaNet is more efficient than Comma in the first modification $\mathcal{S}_f$. We discuss the reasons behind this result in Observation 13.

Since the t-test accepts H_2, H_4, H_5, and H_6, we conclude that the time differences for (1) modeling in $\mathcal{S}_s$ and (2) developing MSCs in all exercises is not statistically significant.

6. RELATED WORK

Researchers have proposed several agent-oriented software development methodologies [5, 13, 3, 20]. Many of these methodologies focus on modeling a multiagent system that is under the control of a single organization. In contrast, Comma models cross-organizational relationships. In Comma, a high-level model based on commitments captures the social relationship among agents (the organizations that are business partners). Unlike Comma, many of the current AOSE methodologies lack an appropriate abstraction for modeling social relationship between the agents.

Tropos [2] resembles Comma in terms of employing high-level concepts. A key difference between the two is how they model social relationships: Tropos employs goal and other dependencies whereas Comma employs commitments. Unlike dependencies, commitments are flexible as they can be manipulated. Commitments reflect the autonomy of the partners since each debtor adopts its commitments through its autonomous actions (communications).

Amoeba [6] employs commitment protocols for process modeling. Amoeba and Comma share the same underlying notion of commitments. In contrast to Comma, which is a methodology for business relationship modeling, Amoeba is a methodology for lower-level interaction modeling, and seeks to specify the protocols whose composition corresponds to the given business process.

Telang and Singh [18] approach RosettaNet from the opposite end to the present paper. They abstract out business modeling patterns from RosettaNet PIPs, in essence by identifying the commitments of the business partners involved that are implicitly understood in each PIP. That is, Telang and Singh discuss how to create and apply patterns that could be included in the Comma library. They use the commitment life cycle as a basis for verifying process specifications.

Mazouzi et al. [10] model agent interaction protocols using Agent UML (AUML), and subsequently translate them into Colored Petri Nets (CPN) to verify low-level properties such as liveness. In contrast, in Comma, a modeler first develops a high-level business model, which provides the correctness properties at a business level [19]. Starting from a business model, the modeler develops agent interaction MSCs. Comma employs model-checking to verify if the MSCs satisfy the business model [19].

Spanoudakis et al. [17] and Garcia-Magarino et al. [7] describe an application of Model-Driven Engineering (MDE) for AOSE. MDE can significantly improve the efficiency of Comma. A modeler can transform a Comma business model into an operational model, such as MSC, using automated model transformation.

Hofreiter et al. [8] describe UMM, UN/CEFACT's Modeling Methodology, a methodology to model inter-organizational business processes as global choreographies. Unlike Comma, UMM fails to capture the high-level business relationships between the process participants. Instead it focuses on the low-level message exchanges, and thus leads to rigid models.

7. CONCLUSIONS AND DIRECTIONS

We introduced Comma, a novel commitment-based methodology for business modeling. We carried out a substantial empirical evaluation of the effectiveness of Comma. We note in passing that such evaluations are not yet common in AOSE, though they are quite prevalent in the broader software engineering community.

Let us summarize the lessons we learned. Our study confirmed the benefits in quality that we expected from Comma because of its foundation in commitments. Specifically, Comma does better on every quality measure: model coverage and precision, and MSC structure (guards), flexibility, and abstraction. The study demonstrated gains in ease of use from Comma in producing models but yielded mixed results with respect to MSCs. Comma yields a superior MSC product, but with a slightly greater difficulty. We expect to see benefits from improving the tooling and training materials supporting Comma. The time spent shows an improvement for Comma though with anomalies. Here too we conjecture that improved tooling and training will prove crucial.

Some important future directions follow naturally from this research. First, on the theoretical side, we are considering expanding Comma to account for a richer variety of norms, e.g., in the spirit of Aldewereld et al. [1], than just commitments. Second, on the practical side, enhanced tooling is an obvious theme. A natural extension would be to support MDE using Comma, as remarked above. Further, we will enhance Comma so it provides guidance for situations where a model must be modified to accommodate evolving requirements.

Third, on the empirical side, we will conduct additional developer studies. Specifically, although our study design mitigated many important threats to validity that can arise in a comparative study, it did not consider important challenges to business interoperation in practice, such as dealing with a legacy system. We conjecture that increasing the complexity of a scenario will tilt the balance further in favor of commitment-based approaches: we defer such evaluations to future research. Further, a threat to validity of any empirical evaluation is whether the subjects correspond closely to the target population (industry practitioners, in our case) in their expertise, experience, and motivation. In their broadest scope, such problems are not readily amenable to comparative research studies, but we plan to explore simplified versions of them.

Acknowledgments

We thank Amit Chopra, Anup Kalia, Emerson Murphy-Hill, and the anonymous reviewers for their helpful comments.

8. REFERENCES

[1] H. Aldewereld, S. Álvarez-Napagao, F. Dignum, and J. Vázquez-Salceda. Making norms concrete. *Proc. AAMAS*, pp. 807–814, 2010.

[2] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *JAAMAS*, 8(3):203–236, 2004.

[3] C. Cheong and M. P. Winikoff. Hermes: Designing flexible and robust agent interactions. V. Dignum, ed., *Handbook of Research on MAS*, ch. 5, pp. 105–139, 2009.

[4] A. K. Chopra, F. Dalpiaz, P. Giorgini, and J. Mylopoulos. Modeling and reasoning about service-oriented applications via goals and commitments. *Proc. CAiSE*, pp. 417–421, 2010.

[5] S. A. Deloach, M. F. Wood, and C. H. Sparkman. Multiagent systems engineering. *Int'l J. Soft. Engg. Know. Engg.*, 11(3):231–258, 2001.

[6] N. Desai, A. K. Chopra, and M. P. Singh. Amoeba: A methodology for modeling and evolution of cross-organizational business processes. *ACM TOSEM*, 19(2):6:1–6:45, Oct. 2009.

[7] I. García-Magariño, J. J. Gómez-Sanz, and R. Fuentes-Fernández. Model transformations for improving multi-agent systems development in INGENIAS. *Proc. AOSE 2009*, *LNCS* 6038, pp. 51–65, 2011.

[8] B. Hofreiter, C. Huemer, P. Liegl, R. Schuster, and M. Zapletal. UN/CEFACT's Modeling Methodology (UMM): A UML profile for B2B e-commerce. *2nd Int'l Wkshp. Best Pract. UML (ER)*, 2006, pp. 19–31.

[9] N. Juristo and A. M. Moreno. *Basics of Software Engineering Experimentation*. Kluwer, 2001.

[10] H. Mazouzi, A. E. F. Seghrouchni, and S. Haddad. Open protocol design for complex interactions in multi-agent systems. *Proc. AAMAS*, pp. 517–526, 2002.

[11] Object Management Group, *UML 2.0 Superstructure Specification*, Oct. 2004.

[12] Oracle. Automating the Quote-to-Cash process. 2009, http://www.oracle.com/us/industries/045546.pdf.

[13] L. Padgham and M. Winikoff. Prometheus: A practical agent-oriented methodology. *Agent-Oriented Methodologies*, ch. 5, pp. 107–135, 2005.

[14] RosettaNet, *Overview: Clusters, segments, and PIPs*, 2008, http://www.rosettanet.org.

[15] M. P. Singh. Semantical considerations on dialectical and practical commitments. *AAAI*, pp. 176–181, 2008.

[16] M. P. Singh. Commitments in multiagent systems. In F. Paglieri et al.(eds.) *The Goals of Cognition*, College Pubs., London, pp. 1–29, 2012. In press; available at http://www.csc.ncsu.edu/faculty/mpsingh/papers/drafts/Commitments-for-MAS.pdf

[17] N. Spanoudakis and P. Moraitis. Model-driven agents development with ASEME. *Proc. AOSE*, 2010.

[18] P. R. Telang and M. P. Singh. Abstracting and applying business modeling patterns from RosettaNet. *Proc. ICSOC*, pp. 426–440, 2010.

[19] P. R. Telang and M. P. Singh. Specifying and verifying cross-organizational business models. *IEEE Trans. Services Comput.*, 5, 2012. http://www.csc.ncsu.edu/faculty/mpsingh/papers/mas/TSC-11-business.pdf

[20] H. Weigand, V. Dignum, J.-J. C. Meyer, and F. Dignum. Specification by refinement and agreement: Designing agent interaction using landmarks and contracts. *Proc. ESAW*, *LNCS* 2577, pp. 257–269, 2002.

[21] P. Yolum and M. P. Singh. Commitment machines. *Proc. ATAL 2001*, *LNAI* 2333, pp. 235–247, 2002.