Protos Overview

Methodology

1. **Step 1:** From the given scenario description, extract sub-scenarios. The scope of each sub-scenario should align with the scope of a business modeling pattern. For each sub-scenario execute steps 2–4.

2. **Step 2:** Identify the role names from the sub-scenario. The scenario description typically specifies the participants using terms like company, partner, and organization. If there is a single partcipant of its kind, the description specifies a unique name for it (e.g., FedEx). For each uniquely named participant, create a role name based on the business function (e.g., Shipper) of that partcipant. In case of multiple participants of the same kind, the description may specify a role name (e.g., Repairer).

3. **Step 3:** Identify the business tasks such as goods, payment, etc. that each role executes. A scenario typically specifies the tasks as actions executed by the participants.

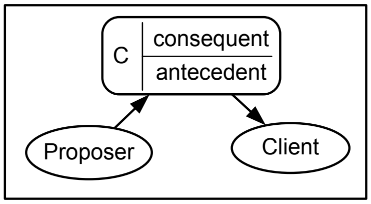
4. **Step 4:** Introduce the pattern for the sub-scenario in the business model. Rename the pattern role names with the role names from Step 2, and introduce the tasks from Step 3 in the antecedents and consequents of the appropriate commitments. In the model, the patterns will compose naturally when the roles referenced by multiple patterns overlap.

5. **Step 5:** Introduce the pattern message sequence chart (MSC) in the operational model. Rename the role and message names in the MSCs to align the ones determined in Step 2 and 3.

Business Model Patterns

Conditional Offer

This is the simplest possible pattern. It merely views a commitment as an offer.



C(PROPOSER, BENEFICIARY, antecedent, consequent)

Figure 1: The conditional offer pattern.

Intent: A proposer conditionally offers to execute a task for a client.

Motivation: For example, a conference committee member commits to a program chair to review a paper that the program chair requests the member to review. The chair makes no converse commitment.

Implementation: A commitment is created in which the proposer is the debtor, the client is the creditor, the consequent is the task that the proposer executes, and the antecedent is a condition that brings the commitment into force. Figure 1 shows this pattern.

Consequences: This presumes a benefit to the proposer from the antecedent of the commitment.

Operationalization: Figure 2 shows the message sequence chart for this pattern. $m1$ is a message that creates C, and $\top$ means *true*.

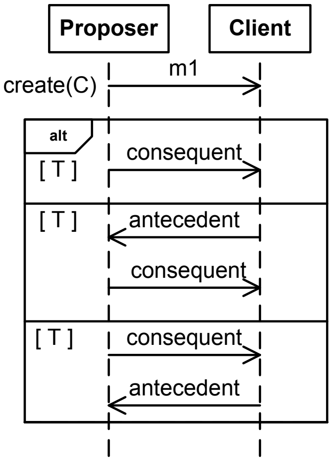
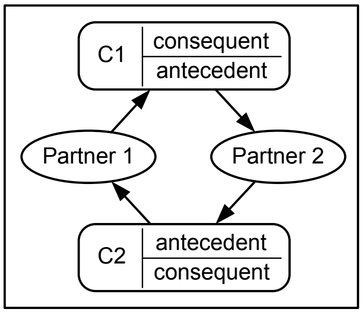


Figure 2: Message sequence chart for conditional offer pattern.

Commercial Transaction



C_1 C(PARTNER 1, PARTNER 2, antecedent, consequent)
C_2 C(PARTNER 2, PARTNER 1, consequent, antecedent)

Figure 3: Commercial transaction.

Intent: This pattern expresses a value exchange between two trading partners. The trading partners negotiate and, upon agreeing, commit to executing certain tasks for each other.

Motivation: A typical situation would be where a seller and a buyer agree to exchange goods for payment. Similarly, if two parties barter goods or services that would fit in the same pattern.

Implementation: A pair of reciprocal commitments between the trading partners treated symmetrically specify the pattern. Figure 3 shows this pattern.

Consequences: In general, the antecedents and consequents of the commitments are composite expressions. Importantly, we need a mechanism to ensure progress by in essence breaking the symmetry.

Operationalization: Figure 4 shows the message sequence chart for this pattern. $m1$ and $m2$ messages respectively create $\mathsf{C1}$ and $\mathsf{C2}$, and $\top$ means *true*.

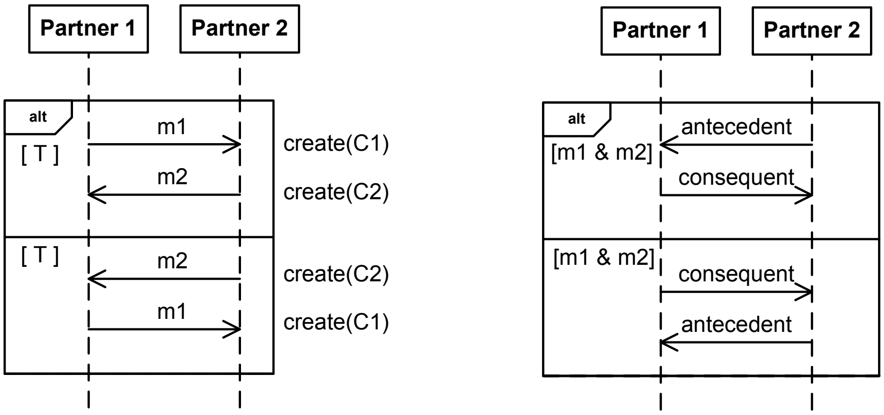
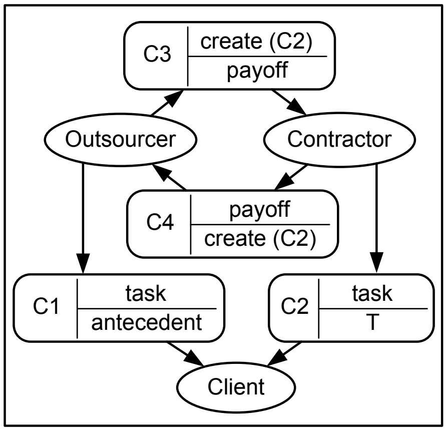


Figure 4: Message sequence chart for commerical transaction.

Outsourcing



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

Figure 5: Outsourcing

Intent: An outsourcer delegates a task to a subcontractor, typically because the outsourcer lacks the necessary capabilities or expects some other benefit, such as reduced costs or a lower risk of failure.

Motivation: Many business organizations outsource noncore activities. As an example, consider a customer who signs up for cable television service. The cable operator commits to the customer for installation. Instead of staffing its entire service area directly, the cable operator outsources the installation task to its local partners in various regions.

Implementation: The outsourcer has a commitment C_1 towards its client to execute a task. The outsourcer and the contractor negotiate, and agree that the contractor will create the commitment C_2 to execute the task if the outsourcer pays. Conversely, the outsourcer commits to pay the contractor if the contractor creates C_2. Note that the antecedent of this commitment is true ($\top$), which means that it is unconditional. We say that the commitment C_2 *covers* the commitment C_1. Eventually when the contractor creates C_2 the original commitment becomes pending. Figure 5 shows this pattern.

Consequences: The commitment from the outsourcer is pending and must either be discharged or reactivated depending on how the contractor performs.

Operationalization: Figure 6 shows the message sequence chart for this pattern. $m1$, $m2$, $m3$ and $m4$ messages respectively create C1, C3, C4 and C2, and $\top$ means *true*.

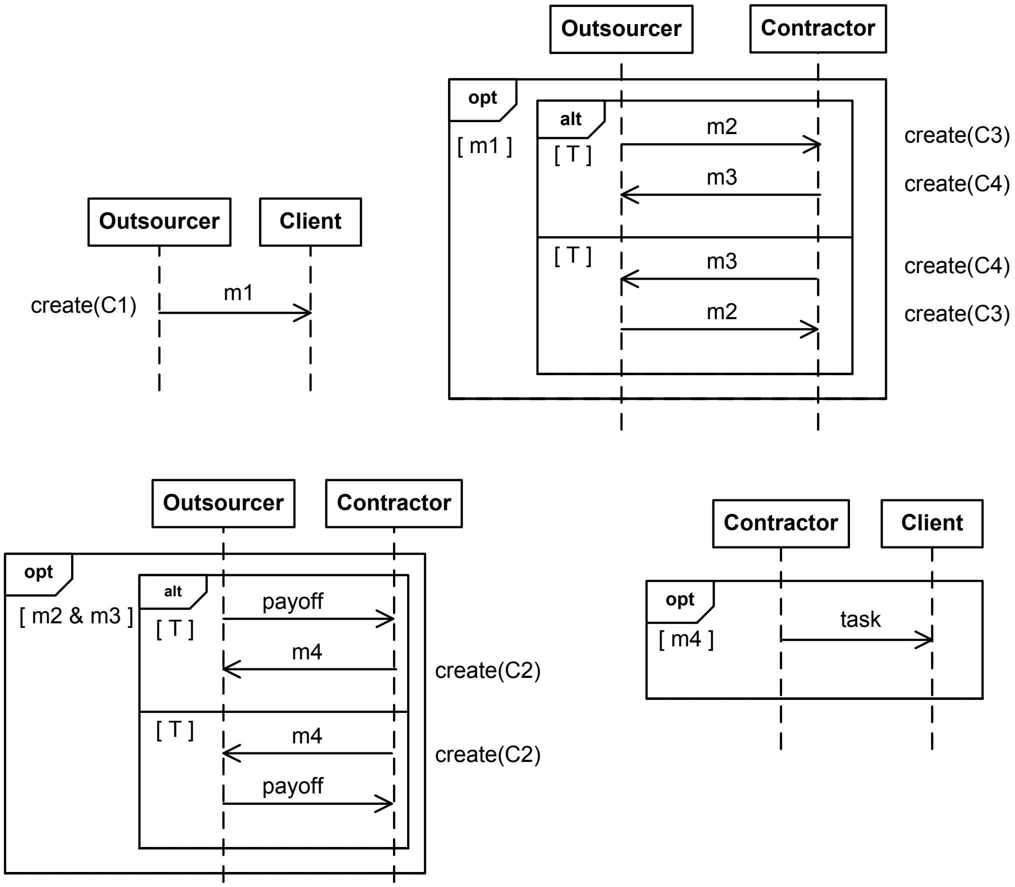
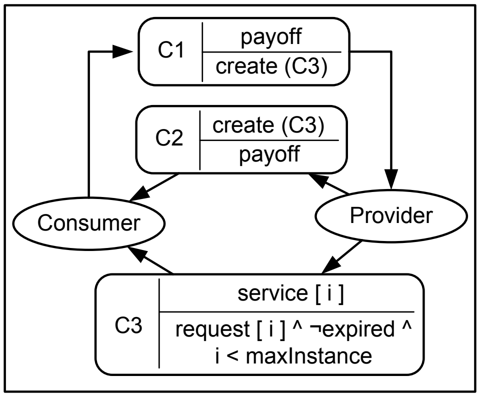


Figure 6: Message sequence chart for outsourcing.

Standing Service Contract



C_1 $\mathsf{C}(\text{CONSUMER}, \text{PROVIDER}, \mathsf{create(C3)}, \mathsf{payoff})$
C_2 $\mathsf{C}(\text{PROVIDER}, \text{CONSUMER}, \mathsf{payoff}, \mathsf{create(C3)})$
C_3 $\mathsf{C}(\text{PROVIDER}, \text{CONSUMER}, \mathsf{request[i]} \wedge \neg\mathsf{expired} \wedge (\mathsf{i} \leq \mathsf{maxInstance}), \mathsf{service[i]})$

Figure 7: Standing service contract.

Intent: A provider sells a long-lived service to a consumer. The service can be bounded by a combination of duration and number of requests.

Motivation: A business service such as plumbing maintenance refers to (potentially) numerous service instances. Whenever the faucet leaks (within specified limitations), the plumber will fix it.

Implementation: The provider and consumer negotiate, and upon agreement, create a pair of commitments C_1 (the consumer commits to paying the service provider, if the service provider commits to provide service), and C_2 (the converse of C_1). Commitment C_3 is the standing service contract. In C_3, the provider commits to the consumer to serve each request prior to expiration up to a fixed number of requests. Figure 7 shows this pattern.

Consequences: Each service request should take a bounded effort.

Operationalization: Figure 8 shows the message sequence chart for this pattern. $m1$, $m2$ and $m3$ messages respectively create C1, C2 and C3, and $\top$ means *true*.

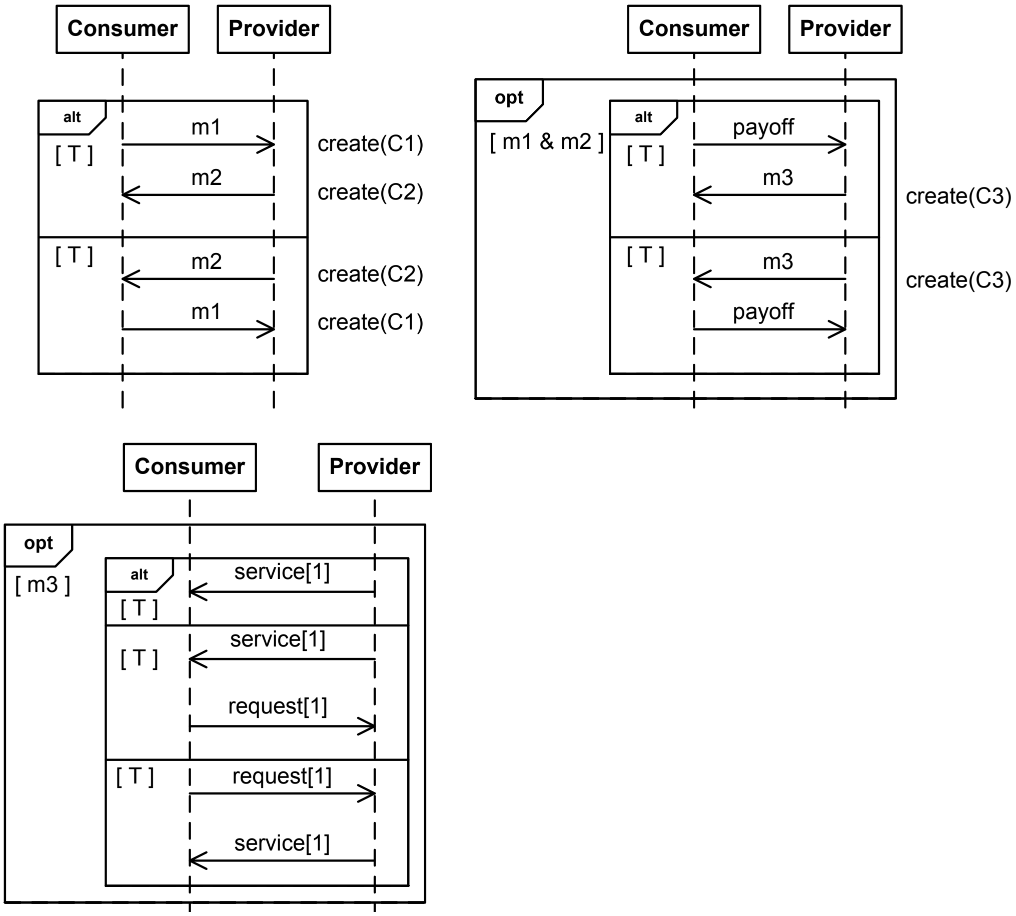


Figure 8: Message sequence chart for service contract.