9 Patterns

9.1 Business Patterns

At the time of this writing, UN/CEFACT has not specified any business patterns that could be used in this workflow. There are efforts underway to discover reference material applicable for e-Business.

1 ## 9.2   Requirements Patterns

2

3   Patterns to describe business collaborations will be developed by UN/CEFACT and
4   added as they become available

9.3 Analysis Patterns

The e-business Business Transaction Pattern metamodel of Section 8.3 provides a framework for constructing e-business collaboration model specifications. This section describes the modeling patterns that apply the metamodel to represent specific business transactions.

9.3.1 Introduction

The business transaction patterns metamodel provides a language and grammar for constructing business collaboration models. Modeling patterns are applications of the metamodel to common business transaction representations. Representations capture common structure and semantics applicable to specific business transactions. The UML activity diagram notation is used to specify business transaction patterns.

Modeling patterns are reusable, generalized business process abstractions that can be applied to many business areas. A metamodel provides the syntax and grammar for expressing designs. Modeling patterns are subjective constructions that meet the requirements of specific business transactions.

9.3.2 Business Transaction State Semantics

A business transaction specifies the contract formation process between two business partners. A contract is used to legally bind parties to a clearly stated intention (promise, obligation) and responsibilities of each party. A contract usually outlines what each party can do in the event the intended actions are not carried out (e.g., promised services not rendered, services rendered but payment not issued). Prudent parties execute (sign) contracts prior to carrying out the intended actions, to limit their liability and to protect their interests.

There are many types of business contract formation processes. For example, an "OFFER-AND-ACCEPTANCE" contract is formed when a product order is "accepted" by a vendor. An "accepted" (signed, mutually agreed upon) purchase order forms a contract between buyer and seller to provide a quantity of product at an agreed-upon price. After the contract is formed, the seller provides the product and the buyer pays for the product. In the event something goes wrong, the buyer and seller both have recourse as described in the contract.

Another example of contract formation occurs when a claim has been accepted for payment; this is a "contract" to perform the issuance of monetary payment (or another form of credit) some time after the "acceptance" (contract formation) of the claim.
All business transactions are treated as contract forming processes in that there is always an obligation (perhaps not residual) between each of the parties participating in the transaction.

The UML activity diagram notation is used to graphically specify these business transactions as design patterns. The pattern for specifying and interpreting these diagrams and the textual notation used to specify element names as well

as conditional expressions is provided in this section. Figure 9-14 illustrates a business transaction specification that does not include a responding business document and Figure 9-10 illustrates a business transaction specification that includes a responding business document.

The initial (START) state and the final (END, CONTROLFAILED, CONTRACTFAILED) state represent the state of a business transaction and not the state of any role that participates in the transaction. It is "by convention" that the initial and final states are placed into the requestor's swim lane. This has no semantic meaning with respect to any participating role. These states could be anywhere in the activity graph as they still pertain to the entire business transaction and not to any particular role. The start state and final state conditions should therefore specify conditions that shall hold before the business transaction can transition into the "default" state (a UML definition).

START State Semantics
The condition that shall hold before transitioning into the initiating transaction activity should test the following [note that a Trading Partner Agreement (TPA) contains the transaction specifications agreed to by participating partners]:
1. The ability of each employee/organization to fulfill their obligations with respect to a TPA e.g.
   a. Are the roles approved trading partners i.e. does a TPA exist that governs the terms and conditions of the transaction?
   b. Do each of the participating roles meet the criterion required for performing the activity e.g. is the employee/organization performing the role authorized to perform the role if authorization is required?
   c. Is a business document non-repudiated if required in a TPA?
   d. Are all data entities tamper-proof, confidential and authenticated as required in a TPA?
2. If a business record exists and it is also syntactically and structurally formatted with respect to the agreed message guideline in a TPA.

Start State Notation
Note that the START conditions are actually guard conditions on the transition from the initial state to the initiating activity in the activity graph. There is no pseudo state "condition" in the UML metamodel. These conditions are not, however, specified as guards in the transaction diagram to improve readability.
It is preferred that these conditions are captured using the following syntax. This improves consistency and will facilitate the translation of these conditions to OCL at a later stage.
States conditions are named in the form <Noun><Property>(<Verb>or<Code>)
- The <Noun> can be a Business Data Entity and the property is named "Status" in the form BDE Status <Code >. Purchase Order Status Open
- The <Noun> can be a Business Document with no named property in the form <Noun> <verb>. Purchase Order Exists
- The <Property> can be the name of a business process support system with no <Noun> in the form <Property><Verb>. Buyer Authorized.
Use the following notation to specify the START conditions:
- TPA Exists
- Requesting Partner Approved
- Responding Partner Approved
- <Business Document> Status <Code> etc. The values for this can be found

in the business dictionary (just search for *StatusCode in the Entity Instances table). Use only valid status from the dictionary or add another valid status to the dictionary e.g. Purchase Order Status Revoked

- <Requesting Role> Authorized e.g. Buyer Authorized
- <Business Document> Exists e.g. Purchase Order Exists
- <Business Document> Non-Repudiated
- <Business Document> Valid
- <Business Document> <Property> Tamper-Proof
- <Business Document> <Property> Confidential
- <Business Document> <Property> Authenticated

END, CONTROLFAILED and CONTRACTFAILED State Semantics

The state of the business transaction transitions into the END state if both parties in a business transaction meet the conditions agreed to in their TPA. There are two final states specified for business transactions:

1. *Contract Failure*. The state machine shall transition into the CONTRACTFAILED state if the intended business contract is not formed but none of the control conditions are violated. For example, a responding role may return a negative business acceptance document that contains a status BDE whose value is "Reject". In these cases a test on the BDE status for reject shall transition the state machine into the CONTRACTFAILED state. The contract failure end state shall only be used for business transactions that permit negative acknowledgements. In these instances the business transaction activity graph is shown in Figure 9-10. If there is no contract failure condition then the business transaction activity graph is shown in Figure 9-11 through Figure 9-15.

2. *Control Failure*. The activity shall transition into the CONTROLFAILED state if any business collaboration control parameter is violated. For example, timeouts, processing exceptions, non-repudiation and authorization exceptions. In these cases both the transaction fails and the contract is not formed.

The conditions that shall hold before transitioning into the SUCCESS state should test the following [note that a TPA contains the transaction specifications agreed to by participating partners]:

1. Each employee/organization has fulfilled their obligations with respect to a trading partner agreement (TPA) e.g.
   a. Have each of the participating roles met the criterion required for performing the activity e.g. were the employee/organization performing the roles authorized to perform the role if authorization is required?
   b. Is a business document non-repudiated if required in a TPA?
   c. Are all data entities in the responding document tamper-proof, confidential and authenticated as required in a TPA?
   d. Were all documents and business signals received by both parties as agreed to in the TPA.
2. If a business record exists and it is also syntactically and structurally formatted with respect to the agreed message guideline specified in a TPA.
3. The retry count has not exceeded the maximum specified.
4. The state machine transitions to the CONTRACTFAILED state if the conditions to transition to the END state are not met and/or a condition on a negative response is satisfied. It is redundant to re-specify the negation of all of the SUCCESS conditions in the FAILED state conditions. Therefore,

the following are the only conditions necessary for the CONTRACTFAILED conditions.

- SUCCESS and (<Business Data Element> Status <Code > and/or ….)

5. The state machine transitions to the CONTROLFAILED state if the conditions to transition to the END state and CONTRACTFAILED states are not met. It is redundant to re-specify the negation of all of the SUCCESS and CONTRACTFAILED conditions in the CONTROLFAILED state conditions. Therefore, the following are the only conditions necessary for the CONTROLFAILED conditions.

- Not SUCCESS or Not CONTRACTFAIL

END State Notation

Note that the END conditions are actually guard conditions on the transition from the end status in the activity graph. There is no pseudo state "condition" in the UML metamodel. These conditions are not, however, enumerated as guards in the transaction diagram to improve readability. It is preferred that these conditions are captured using the following syntax. This improves consistency and will facilitate the translation of these conditions to OCL at a later stage.

States conditions are named in the form <Noun><Property>(<Verb>|<Code>)

- The <Noun> can be a Business Data Entity and the property is named "Status" in the form BDE Status <Code>. Purchase Order Status Open
- The <Noun> can be a Business Document with no named property in the form <Noun> <verb>. Purchase Order Acceptance Exists
- The <Property> can be the name of a business process support system with no <Noun> in the form <Property><Verb>. Seller Authorized, Receipt Non-Repudiated.

Use the following notation to specify the END conditions:

- <Business Document> Status <Code> etc. The values for this can be found in the business dictionary (just search for *StatusCode in the Entity Instances table). Make sure you only use valid status from the dictionary or add another valid status to the dictionary e.g. Purchase Order Acceptance Status Approved
- <Responding Role> Authorized e.g. Seller Authorized
- <Business Document> Exists e.g. Purchase Order Acceptance Exists.
- <Business Signal> Exists e.g. Verification of Receipt Exists
- <Business Document> Non-Repudiated
- Verification of Receipt Non-Repudiated
- <Business Document> Valid
- <Business Signal> Valid
- <Business Document> <Property> Tamper-Proof
- <Business Document> <Property> Confidential
- <Business Document> <Property> Authenticated

9.3.3 Business Transaction Pattern Rationale

This section provides the design rationale for the time-out specification in each business transaction pattern. This pattern rational is presented within a document-processing framework that comprises the following steps.

1. Grammar validation. The task of verifying that the grammar of a message is valid (usually only the header of the message at this step).
2. Sequence validation. The task of verifying that the collaboration control information is valid with respect to the business transaction specification.
3. Schema validation. The task of verifying that the message schema is valid with respect to a message guideline agreed to by both partners. It is recommended that message receipt be acknowledged after this validation step to ensure that documents are "readable" as well as "accessible".
4. Content validation. The task of verifying that the content of a message is valid with respect to any business rules that govern the formation of a contract. It is recommended that business acceptance be acknowledged after this validation step.
5. Activity processing. The task of processing the request in the initiating business document.

Figure 9-1 illustrates the processing of an initiating message when the contract-closing (contract acceptance document) message is an acknowledgement of receipt. The acknowledgement of receipt is a business signal i.e. it does not map onto a business document.
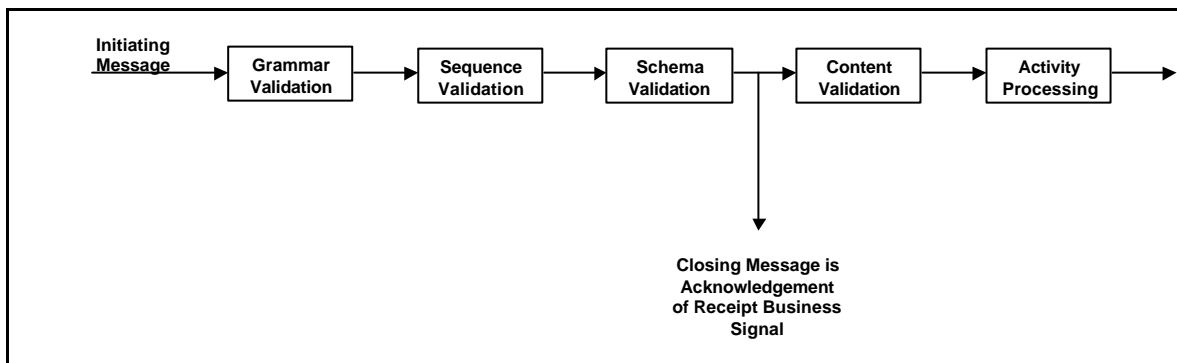
Figure 9-1   Acknowledgement of Receipt Closing Message

The table in Figure 9-2 shows example timeout parameters for this business transaction. The Information Distribution and Notification business activity specification (see Modeling Metamodel) use this design pattern.

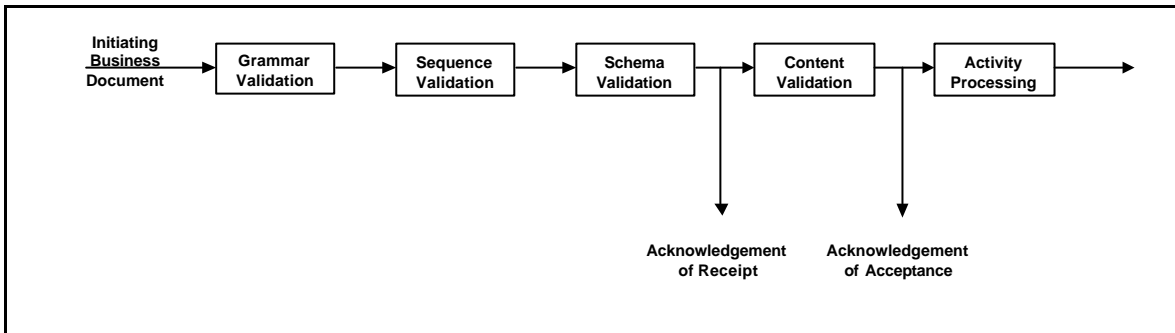| Role Name	Activity Name	Time to Acknowledge Receipt	Time to Acknowledge Acceptance	Time to Perform
Role	Activity	24hr	N/A	24hr

Figure 9-2   Acknowledgment of receipt timeout parameters

Figure 9-3 illustrates the processing of an initiating message when the closing message is an acknowledgement of acceptance. The acceptance message can

1   be either substantive or non-substantive. A substantive business acceptance
2   message includes business data from the initiating message e.g. product, price
3   and quantity in a substantive purchase order acceptance document. A
4   substantive business acceptance message contains a business document. A
5   positive non-substantive business acceptance message contains the initiating
6   business document identification data. A negative non-substantive business
7   acceptance message contains the initiating business document identification
8   data, the reason for rejection and syntactic error messages indicating the
9   business data elements in which the error was found. A positive non-
10   substantive acceptance message is a business signal i.e. it does not map onto
11   a business document. Note the following:
12
13   1. If a substantive business acceptance is required then a responding
14   **business document** is specified in a business transaction.
15   2. If a non-substantive business acceptance is required then a responding
16   business document is **not** specified in a business transaction.
17

20 **Figure 9-3 Acknowledgement of Business Acceptance Closing Message**

21
22   The table in Figure 9-4 shows example timeout parameters for this business
23   agreement. The Business Transaction Activities use this design pattern when a
24   substantive business acknowledgement of acceptance is required.
25

Role Name	Activity Name	Time to Acknowledge Receipt	Time to Acknowledge Acceptance	Time to Perform
Role	Activity	2hr	6hr	6hr

26   Figure 9-4   Acceptance acknowledgment timeout parameters

27   Figure 9-5 illustrates the processing of an initiating message when the closing
28   message is a responding business document. The Query/Response business
29   activity specification uses this design pattern.

1
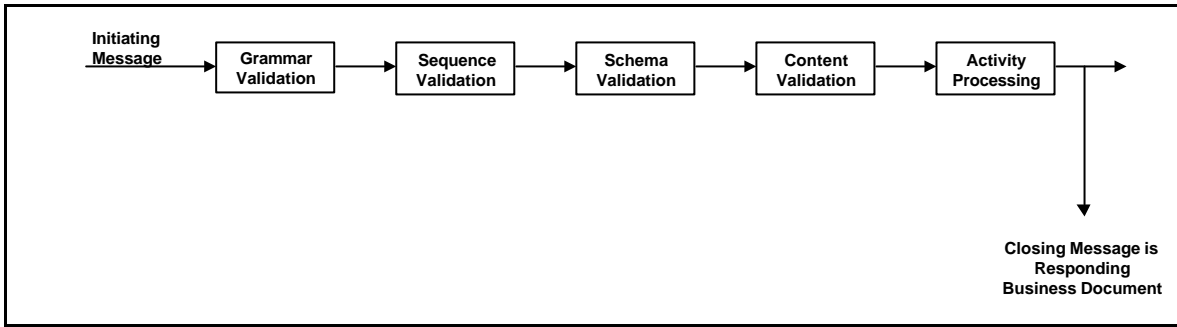
2   **Figure 9-5   Responding Business Document is Closing Message**

3

4   The table in Figure 9-6 shows example timeout parameters for this business
5   transaction.

6

Role Name	Activity Name	Time to Acknowledge Receipt	Time to Acknowledge Acceptance	Time to Perform
Role	Activity	N/A	N/A	24hr

7   Figure 9-6   Timeout parameters for no acknowledgment

8   It is possible to specify acknowledgements and a responding business
9   document as part of the business agreement. Figure 9-7 illustrates the
10   processing of an initiating message when there is a requirement for an
11   acknowledgement of receipt, a non-substantive acknowledgment of acceptance
12   and a responding document. Note that the acceptance message cannot be
13   specified as substantive i.e. a business document. It can only be a non-
14   substantive i.e. a business signal. If the acceptance shall be substantive then
15   two business transactions are required.

16
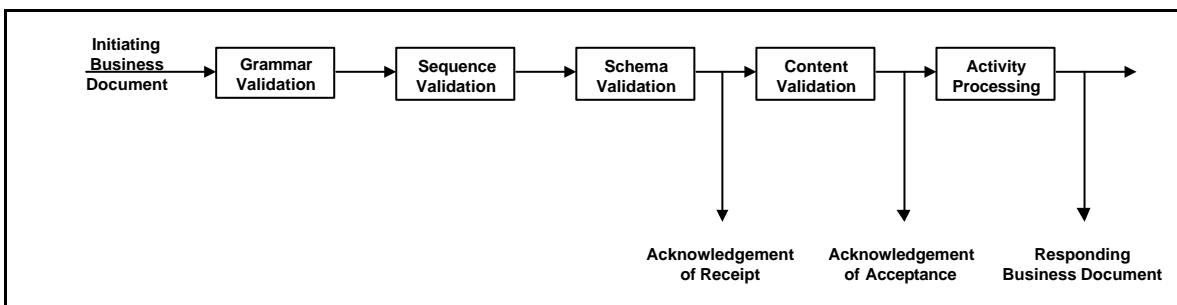
17   **Figure 9-7   Receipt, Business Acceptance and Business Document Response**

18

19   The table in Figure 9-8 shows example timeout parameters for this business
20   transaction. The Business Transaction business activity specification that

mandates the return of a non-substantive business acceptance
acknowledgement uses this design pattern.

Business Activity Performance Controls				
Role Name	Activity Name	Time to Acknowledge Receipt	Time to Acknowledge Acceptance	Time to Perform
Role	Activity	2hr	6hr	24hr

Figure 9-8   Timeout parameters for receipt, business acceptance and document reponse

Interpreting how the contact is closed using a substantive or non-substantive
acknowledgement of acceptance is a based on three cues.
1. There is a value for "Time to Acknowledge Acceptance".
2. The value for "Time to Perform" is either equal or not equal to the "Time to
    Acknowledge Acceptance".
3. There either is or is not a business document response.
Case 1:
If
1. There is a value for "Time to Acknowledge Acceptance".
2. The value for "Time to Perform" equals the "Time to Acknowledge
    Acceptance".
3. There is no business document response.
Then the acknowledgement of acceptance is non-substantive.
Case 2:
If
1. There is a value for "Time to Acknowledge Acceptance".
2. The value for "Time to Perform" equals the "Time to Acknowledge
    Acceptance".
3. There is a business document response with the verb acceptance
    appended to a noun e.g. Purchase Order Acceptance.
Then the acknowledgement of acceptance is substantive.
Case 3:
If
1. There is a value for "Time to Acknowledge Acceptance".
2. The value for "Time to Perform" does not equal the "Time to Acknowledge
    Acceptance".
3. There is a business document response.
Then the acknowledgement of acceptance is non-substantive.

9.3.3.1   Notification of Failure Semantics

The intent of the notification of failure business transaction is to revoke an
initial business contract offer if the contract formation process fails. The
requesting partner can only initiate this business transaction. A responding
partner is required to return an exception document or a negative

acknowledgement document when an error is generated.

Notification of failure shall only be initiated when a terminating transaction does not leave both parties with a mutual agreement as to the state of a business transaction. This condition exists when:

1. The initiating partner's business activity times-out when waiting for a specified response to its requesting business document.

2. The responding business document is erroneous, not authorized or not digitally signed as agreed to in a Trading Partner Agreement.

The UN/EDIFACT model trading partner agreement (http://www.unece.org/trade/untdid/texts/d240_d.htm) recommends the following procedure be agreed to by both partners in their Trading Partner Agreement so as to leave each partner with a mutual understanding of when a contract is not formed:

> "3.2.3. In the event that the originating party has not received, for a properly transmitted Message, a required acknowledgement and no further instructions have been provided, the originating party may declare the Message null and void by so notifying the receiving party."

The contract requestor initiates this business transaction when the originating partner times-out when waiting for a specified response. Where notifications are sent is defined in a trading partner agreement and may be different for each business transaction.

It is recommended that the Notification of Failure business transaction be executed over an alternate communication channel to prevent the inability to report failures potentially caused by communication failures. It is recommended that the organizational entity responding to the notification of failure is different from the organization that failed to respond to the original business document request ("offer").

In an e-business network environment, this "alternate communications channel" should at least be interpreted to mean communicating with an application server that is different from the application server that has not serviced the original business document request. Trading partners should, however, agree on this "alternate communications channel."

This business transaction is not exercised when a responding business partner encounters a business process or control exception when responding to a business document request.

9.3.4 Business Transaction Patterns

Business models may find it convenient to develop business transaction design patterns to facilitate the development of their specifications. The following six property-value conventions for business transactions have proven useful in the application of the metamodel to existing business requirements.

1. Commercial Transaction

2. Request / Confirm

3. Query / Response

4. Request / Response

5. Notification

6. Information Distribution

These conventions are applied by stereotyping the requesting business activity with the syntax shown in Figure 9-10.

Business Transaction	Stereotype
Commercial Transaction	«CommercialTransactionActivity»
Request / Confirm	«RequestConfirmActivity»
Query / Response	«QueryResponseActivity»
Request / Response	«RequestResponseActivity»
Notification	«NotificationActivity»
Information Distribution	«InformationDistributionActivity»

Figure 9-9   Business transaction stereotypes

9.3.4.1   Commercial Transaction Pattern [Contract formation, e.g., place order]

The commercial transaction design pattern is illustrated in Figure 9-10. This design pattern is best used to model the "*offer and acceptance*" business transaction process that results in a residual obligation between both parties to fulfill the terms of the contract. The following principals and definitions of offer and acceptance are taken from the following URL: http://www.anu.edu.au/law/pub/edinst/anu/contract/lectures/moles/semest1/ MContractFormationOfferAnd.html#MContrac-Whatisanoffer.

Offer and acceptance are a means of analyzing the process of negotiation to decide whether and when a contract has been made and what therefore constitute its terms.

There is no satisfactory definition of an offer beyond identifying it by reference to the fact that it can be converted into a contract by an act of acceptance. Whether it can be accepted depends upon the objective intention of the party making the statement which is alleged to be an offer.

Making an offer exposes one to the imposition of legal liability by another. In deciding whether statements amount to an offer, the courts are said to use an objective test. Therefore under the objective test an apparent intention to be bound will suffice if 2 conditions are satisfied:

- *conduct of the alleged offeror shall be such as to induce a "reasonable person" to believe that he/she is making the alleged offer.*

- *the alleged offeree shall actually hold that belief - ie believe that the offeror is making a genuine offer,*

1   *as opposed, for example, to playing a game.*

2   The pattern specifies an originating business activity sending a business
3   document to a responding business activity that may return a business
4   signal or business document as the last responding message. The pattern
5   mandates the acknowledgement of the requesting business document when
6   it passes a "Business Acceptance" test, i.e. passes the content validation
7   step as illustrated in Figure 9-7. This acknowledgement can be substantive
8   i.e. contains the terms of acceptance of a contract or it may be non-
9   substantive i.e. a general auditable business signal. The intent of this
10   business transaction pattern is to model the formation of an offer and
11   acceptance business contract[1]. If the requesting role transitions from their
12   business activity into the control failure state then the role shall initiate a
13   notification of failure (see notification design pattern) business transaction to
14   revoke their original offer.
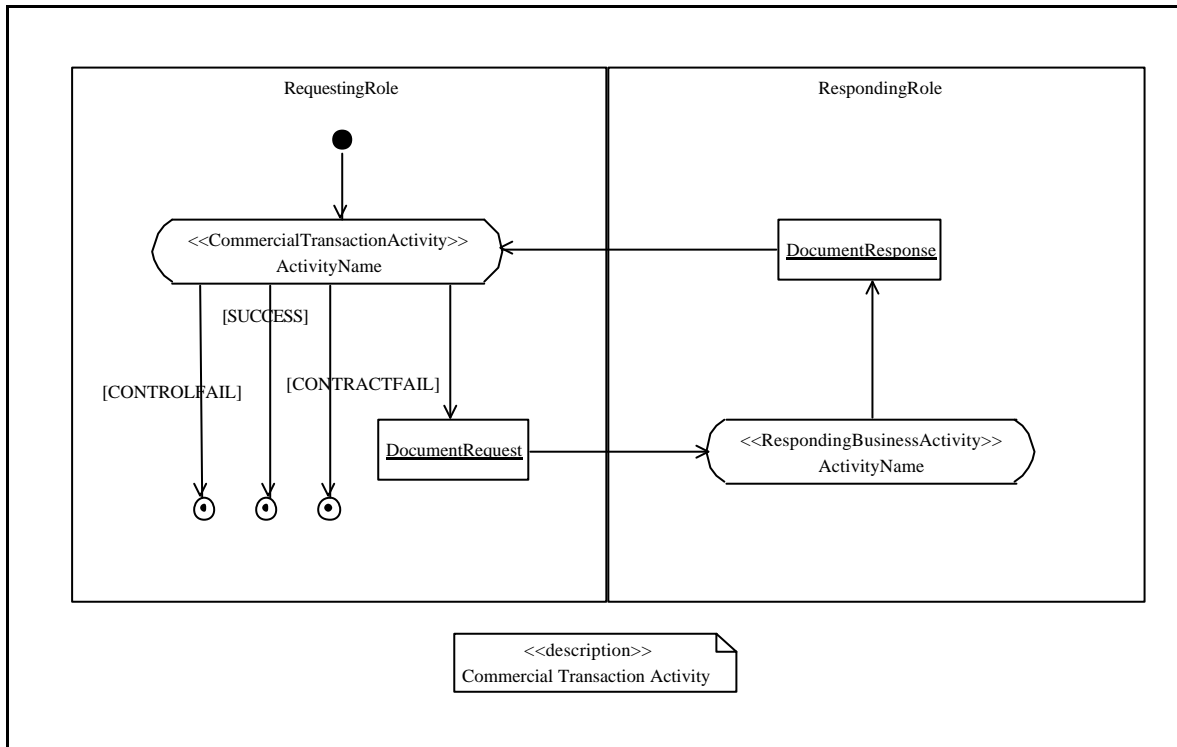
15

16 **Figure 9-10   Commercial Transaction Activity Design Pattern**
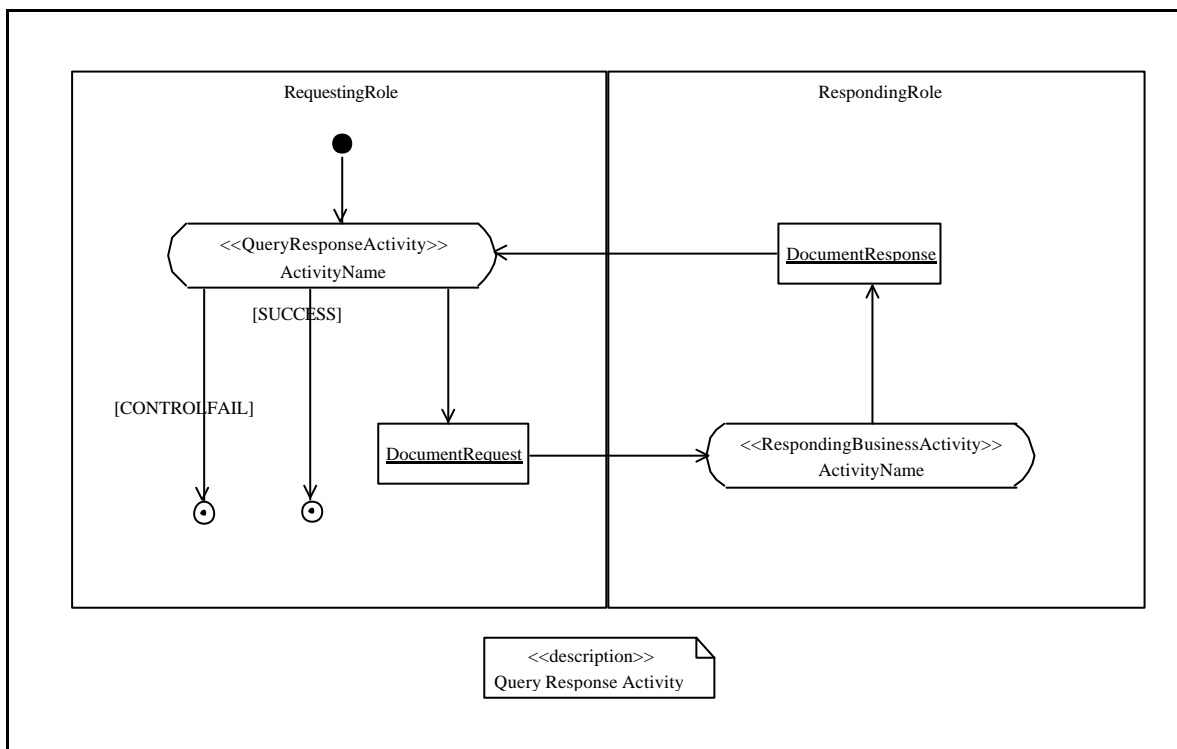
17

---

[1] Refer to the following documents to understand on-line business contract formation.
- PART 2 UNIFORM RULES OF CONDUCT FOR INTERCHANGE OF TRADE DATA BY TELETRANSMISSION (UNCID), CHAPTER 2 - Text of the Uniform Rules of Conduct, http://www.unece.org/trade/untdid/texts/d220_d.htm
- UN/ECE RECOMMENDATION No.26, THE COMMERCIAL USE OF INTERCHANGE AGREEMENTS FOR ELECTRONIC DATA INTERCHANGE, http://www.unece.org/trade/untdid/texts/d240_d.htm
- The Commercial use of Electronic Data Interchange, Section of Business Law American Bar Association, A report and model trading partner agreement, http://www.abanet.org/buslaw/catalog/5070258.html

1 ***9.3.4.2*** ***Query/Response Pattern [Static information, e.g., obtain catalog]***
2
3 Figure 9-11 illustrates the query/response design pattern. The
4 query/response design pattern specifies one business document as output
5 and one business document as input. These documents adhere to the
6 query/response business document design pattern specified in the previous
7 section. Query/Response does not permit the return of auditable business
8 signals i.e. receipt acknowledgement or business acceptance
9 acknowledgement.
10 The responding activity is most likely to be serviced by an organizational
11 role i.e. not by an employee role. There is no non-repudiation requirement
12 for these activities.

RequestingRole   RespondingRole

<<QueryResponseActivity>>
ActivityName

[SUCCESS]

[CONTROLFAIL]

DocumentRequest

DocumentResponse

<<RespondingBusinessActivity>>
ActivityName

<<description>>
Query Response Activity

13

14 **Figure 9-11 Query/Response Activity Design Pattern**

15

16 The query/response design pattern specifies a query for information that a
17 responding partner already has e.g. against a fixed data set that resides in a
18 database. The response comprises zero or more results each of which meet
19 the constraining criterion in the query. For example, a query for the products
20 under $500 will yield any number of product results in the same response all
21 of which have a price under $500. This pattern should be used when the
22 response comprises a collection of results each of which meet the
23 constraining criterion specified in the query. The Request/Response design
24 pattern should be used instead.
25

1   ### 9.3.4.3   Request/Response Pattern [Dynamic information, e.g., obtain Buyer
2   ID, obtain quote]
3
4   Figure 9-12 illustrates the request/response design pattern. Note that there
5   is usually no residual obligation between both parties to fulfil the terms of a
6   contract as in the Business Transaction Activity pattern. For example, a
7   request for price and availability does not result in the responding party
8   allocating product for future purchase and it does not result in the requesting
9   party being obligated to purchase the products. This pattern specifies the
10   exchange of a requesting and responding business document.
11   Acknowledgement of business acceptance is not permitted – use the
12   "Business Transaction Activity" stereotype if this is required.
13   The responding activity is most likely serviced by organizational or
14   employee roles. Non-repudiation is an optional requirement for these
15   activities.
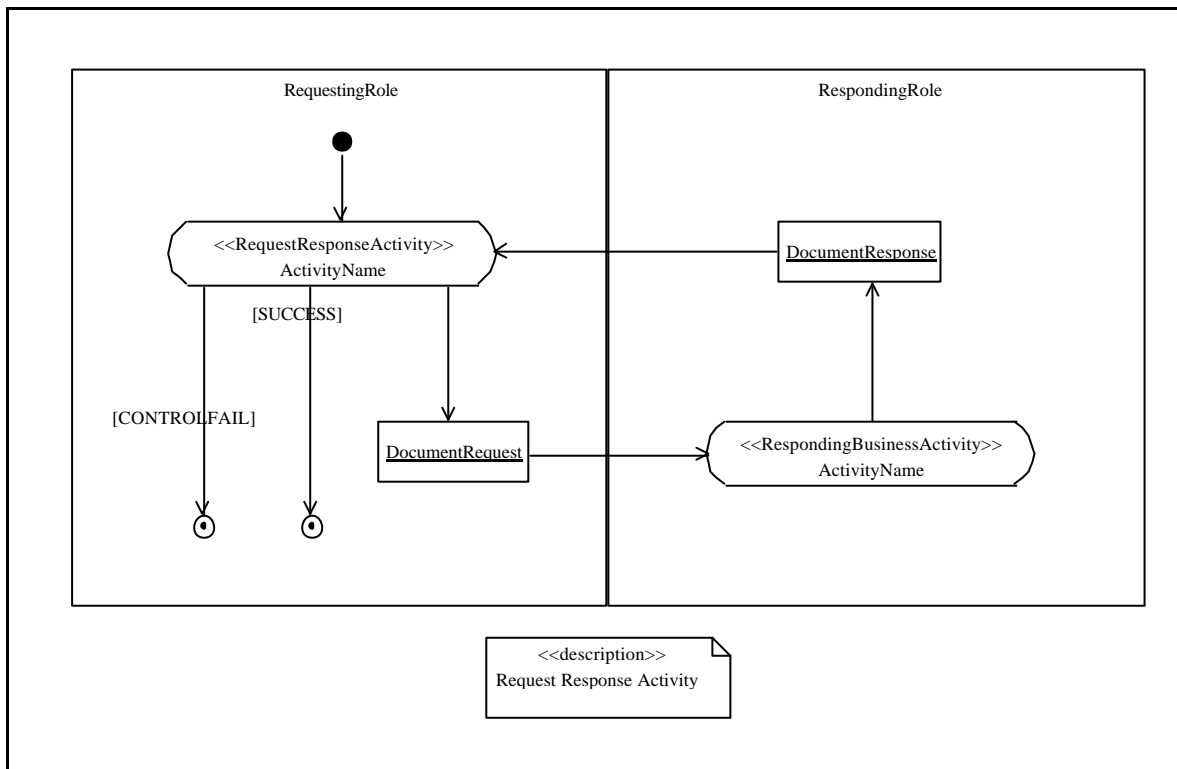
16

17   **Figure 9-12 Request/Response Activity Design Pattern**

18
19   The request/response activity pattern shall be used for business contracts
20   when an initiating partner requests information that a responding partner
21   already has and when the request for business information requires a
22   complex interdependent set of results. For example, a price and availability
23   request may constrain the response such that the sum of all products
24   returned in each of the results (one response may comprise zero or more
25   results) shall be less than 100. This response requires some business
26   processing on a query before a response is returned to the requestor. This
27   flow pattern is used in conjunction with the Request/Response business

1 document design pattern that includes syntax for expressing Business
2 Constraints that apply to the collection of results in the response. If there is
3 no "aggregate" or "interdependent" constraints that shall be applied to a set
4 of results then the query/response pattern shall be used.

9.3.4.4 *Request/Confirm Pattern [Status information, e.g., Obtain order status]*

8 Figure 9-13 illustrates the request/confirm design pattern. Note that there is
9 usually no residual obligation between both parties to fulfill the terms of a
10 contract as in the Commercial Transaction Activity pattern. For example, a
11 request for authorization to sell certain products expects a confirmation
12 response to the request that confirms if the requestor is authorized or not
13 authorized to sell the products. This pattern specifies the exchange of a
14 requesting and responding business document. If acknowledgement of
15 receipt is expected it is the initiator's obligation to follow up on the request
16 until an acknowledgement of receipt is received. Acknowledgement of
17 business acceptance is not permitted – use the "Commercial Transaction
18 Activity" pattern if this is required.
19 The responding activity is most likely serviced by organizational or
20 employee roles. Non-repudiation is an optional requirement for these
21 activities.
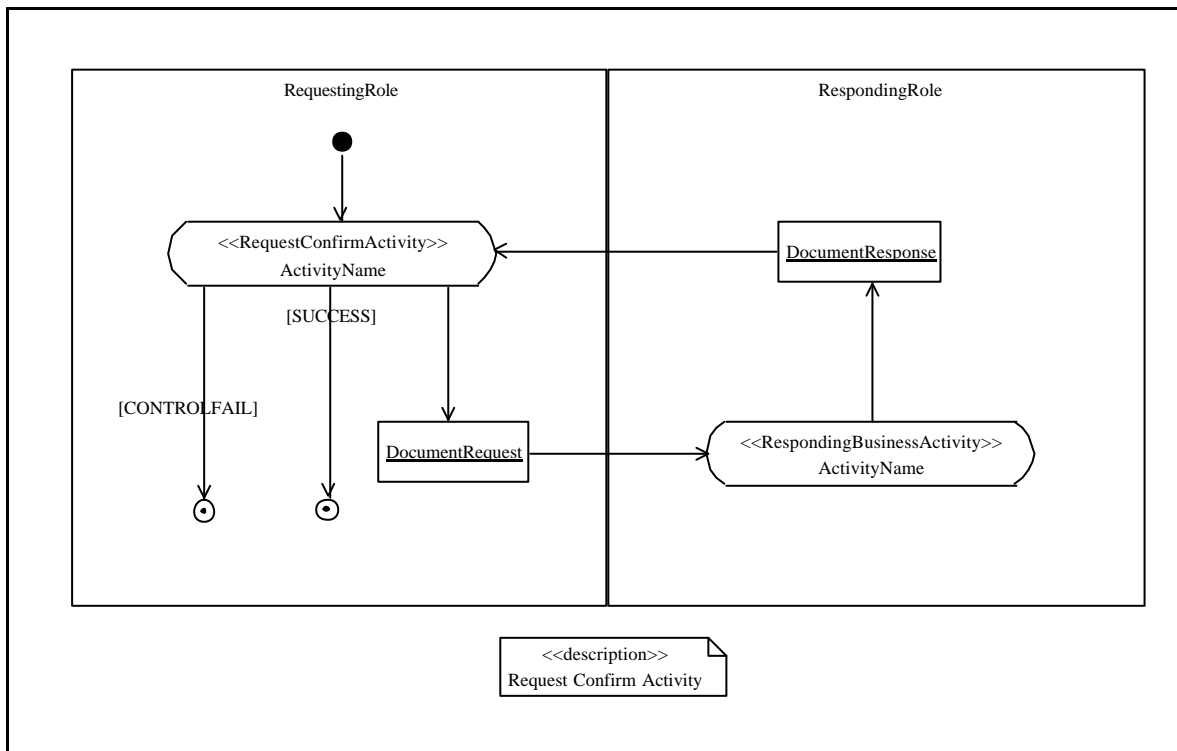
Figure 9-13 Request/Confirm Activity Design Pattern

25 The request/confirm activity pattern shall be used for business contracts
26 where an initiating partner requests confirmation about their status with
27 respect to previously established contracts or with respect to a responding

partner's business rules.

9.3.4.5   *Information Distribution Pattern*

Figure 9-14 illustrates the information distribution design pattern. This pattern specifies the exchange of a requesting business document and the return of an acknowledgement of receipt business signal. This pattern is used to model an *informal* information exchange business transaction that therefore has no non-repudiation requirements.
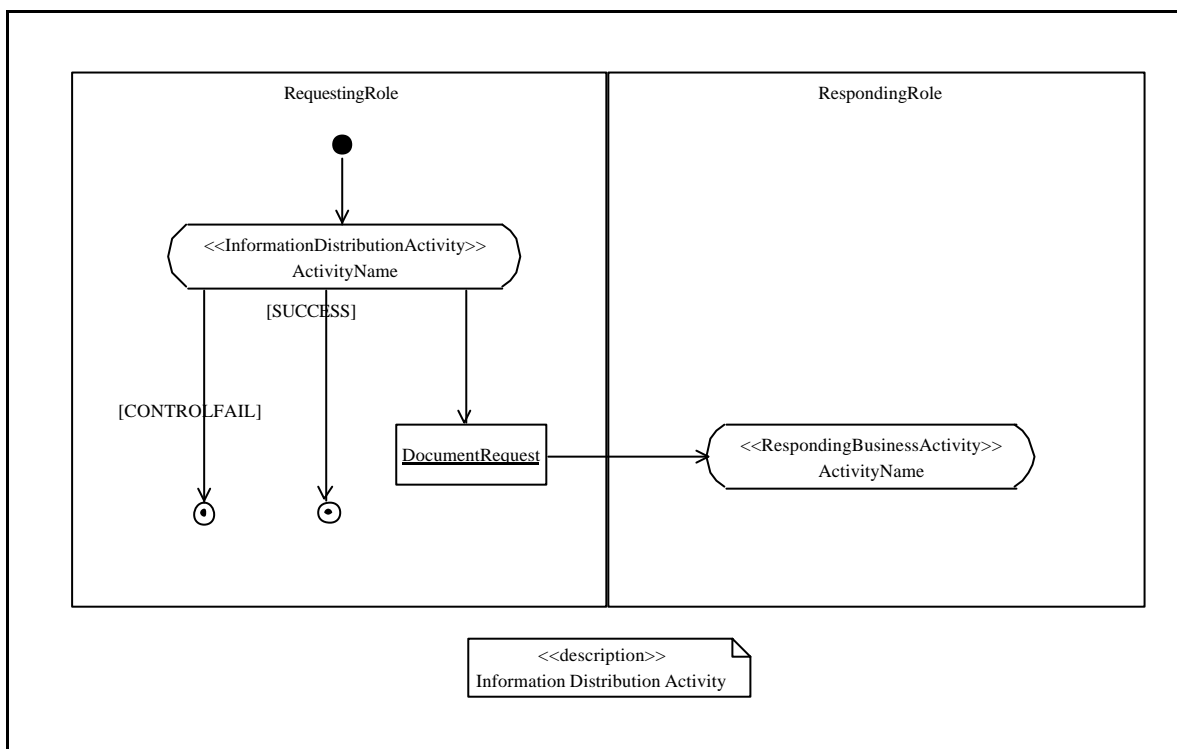
Figure 9-14 Information Distribution Design Pattern

9.3.4.6   *Notification Pattern*

Figure 9-15 illustrates the notification design pattern. This pattern specifies the exchange of a notifying business document and the return of an acknowledgement of receipt business signal. This pattern is used to model a *formal* information exchange business transaction that therefore has non-repudiation requirements.
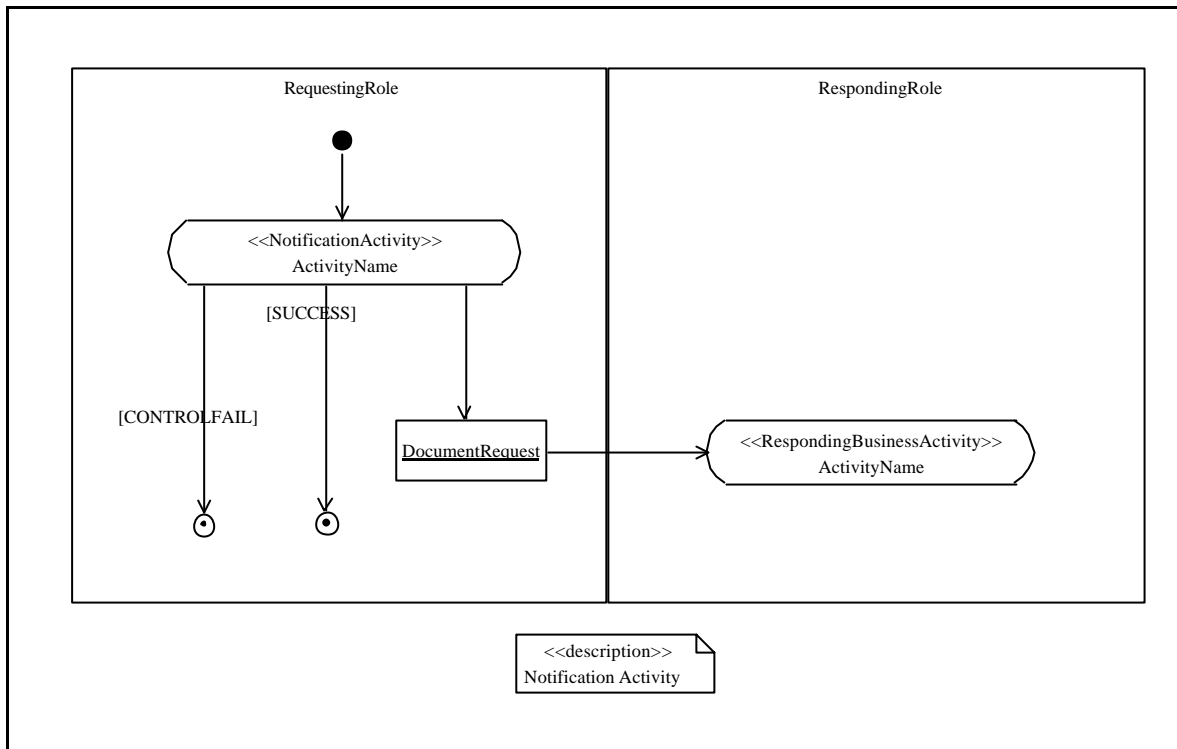
Figure 9-15 Notification Design Pattern

9.3.5 *Property values for Business Transaction activities*

The table in Figure 9-16specifies the property-values for requesting business activities for each of the business transaction stereotypes.

	Time to Acknowledge Receipt	Time to Acknowledge Acceptance	Time to Perform	Authorization Required	Non-repudiation of Origin and Content	Non-repudiation of Receipt	Recurrence
Commercial Transaction	2hrs	6hr	24hr	true	true	true	3
Request / Confirm	null	Null	24hrs	false	false	true	3
Request / Response	null	Null	4hrs	false	false	null	3
Query / Response	null	Null	4hrs	false	false	null	3
Notification	24hrs	Null	24hrs	false	true	true	3
Information Distribution	24hrs	null	24hrs	false	false	false	3

2   **Figure 9-16**   **Requesting business activity property-values**

3

4   The table in Figure 9-17 specifies the property-values for responding business
5   activities for each of the business transaction stereotypes.

	Time to Acknowledge Receipt	Time to Acknowledge Acceptance	Time to Perform	Authorization Required	Non-repudiation of Origin and Content
Business Transaction	2hrs	6hr	24hr	true	true
Request / Confirm	2hrs	null	24hrs	true	false
Request / Response	null	null	4hrs	false	false
Query / Response	null	null	4hrs	false	false
Notification	24hrs	null	24hrs	false	false
Information Distribution	24hrs	null	24hrs	false	false

6   **Figure 9-17**   **Responding business activity property-values**

It is recommended that the stereotype in Figure 9-18 be used on the responding business activity when authorization is required for a responding activity to respond to a business document request.

Business Activity	Stereotype
Authorized Activity	«AuthorizedActivity»

Figure 9-18   Authorized responding activity stereotype

Another convention that makes the application of these stereotypes easier is to only stereotype the requesting business activity when a symmetrical business relationship is designed. With this convention the time to perform, time to acknowledge receipt, time to acknowledge acceptance, non-repudiation and authorization requirements are assumed symmetrical and thus applicable equally to both the requesting and responding business activities.

9.3.6 Timeout Exceptions

A time-out parameter shall be specified whenever a requesting partner expects one or more responses to a business document request. A requesting partner shall not remain in an infinite wait state. There shall be a time-out parameter specified for each expected response. There are four possible responses and hence four potential time-out specifications:

- *Acknowledge Receipt.* The time a responding role has to acknowledge receipt of a business document.

- *Non-Substantive Acknowledge Business Acceptance.* The time a responding role has to non-substantively acknowledge business acceptance of a business document.

- *Substantive Acknowledge Business Acceptance.* The time a responding role has to substantively acknowledge business acceptance of a business document.

- *Perform Transaction.* The time a business transaction has to complete.

The time-out value for each of the time-out parameters is absolute i.e. not relative to each other. All timers start when the requesting business document is sent. The timer values shall comply with the well-formedness rules in Section 8.3.

If the retry count is not zero and a time-out condition is signaled for any of the expected responses then the original business document shall be resent from the initiating partner role. The original business document shall be sent even if responding acknowledgements have already been received.

If an initiating partner receives a response after a time-out condition is signaled and the original business document has already been resent then this response shall be ignored. A responding partner that receives a business document from a retry shall terminate their responding transaction for the previous business document and the retry request shall be serviced.

Upon sending a business document retry, it SHALL be guaranteed that the sending party resends an identical business document, save for a timestamp. Otherwise, a receiving partner shall be capable of rolling back an incoming business document at any point in time through the acknowledgment interval, acceptance interval, and back-end processing interval.

When the time to perform an activity equals the time to acknowledge receipt or the time to acknowledge business acceptance then the highest priority time out exception shall be used when the originator provides a reason for revoking their original business document offer. The time to perform exception is lower priority than both the time to acknowledge receipt and the time to acknowledge business acceptance.

9.3.7 Business Protocol Exceptions

A business protocol exception terminates the business transaction. The following are business protocol exceptions.

1. Negative acknowledgement of receipt. The structure/schema of a message is invalid.
2. Negative acknowledgement of acceptance. The business rules are violated.
3. Performance exceptions. The requested business action cannot be performed.
4. Sequence exceptions. The order or type of a business document or business signal is incorrect.
5. Syntax exceptions. There is invalid punctuation, vocabulary or grammar in the business document or business signal.
6. Authorization exceptions. Roles are not authorized to participate in the business transaction.
7. Business process control exceptions. Business documents are not signed for non-repudiation.
8. A responding role that throws a business protocol exception signals the exception back to the requesting role and then terminates the business transaction. A requesting role that throws a business protocol exception terminates the transaction and then sends a notification revoking the offending business document request. The requesting role cannot send a business signal to the responding role.

9.3.8 Pattern Property Modification Rules

The following rules apply when modifying design pattern properties.

1. If the convention for a time property value is >0 then it cannot be changed to NA.

2. If the convention for a time property value is NA then it cannot be changed. This is because the change would add or remove a role interaction that is not allowed, as it will change the convention.

3. The non-repudiation values specified in the convention cannot be changed except that both Query/Response and Request/Confirm can be changed to non-repudiation required. Changing any of the other non-repudiation values would change the semantic meaning of the business transaction.

4. The Authorization Required property can only be changed to N. It cannot be changed to Y if it is already set to N according to the convention.

9.3.9 Requesting Business Activity

Preconditions and post-conditions should be specified when there are structure or content constraints that apply to the document when it is used in a particular business transaction. Preconditions are specified in the guard of a transition from the initial pseudo state to a requesting business activity. Post-conditions are specified in the guard of a transition from a requesting business activity to state vertex that is the state of the machine when the business activity is successfully performed.

9.3.10 Object Flow

Business document flow states specify the business document flow between roles as they perform business activities. Each business document has a source and target business activity.

An object flow has a type that is a document envelope. An envelope contains one or more structured and unstructured business documents. A business document is signed if non-repudiation of origin and content is required. A detached signature is used to provide non-repudiation of origin and content of a business document as it pertains to the entire document. The signature shall be part of the business document for authorization as the content of the document is authorized.

Structured business documents contain information entities. Information entities contain other information entities. Containment is modeled using UML associations.

1 **9.4 Design Patterns**
2
3 Networked business services and business agents are configured to execute
4 business transactions and business collaboration agreements. The UML sequence
5 diagram notation is used to specify Business Service interactions. The following
6 Business Service interactions are possible.
7   1. Service-Service.
8   2. Agent-Service-Service.
9   3. Service-Service-Agent.
10   4. Service-Agent-Service.
11   5. Agent-Service-Agent
12
13 *9.4.1 Service-Service*
14
15 Figure 9-19 illustrates the Service-Service business service interaction pattern
16 used in the business transaction patterns of Section 8.3.

17
18 **Figure 9-19   Service-Service pattern**

1
2
3   Figure 9-20 shows the class diagram for the Service-Service pattern, which is
4   the base pattern.

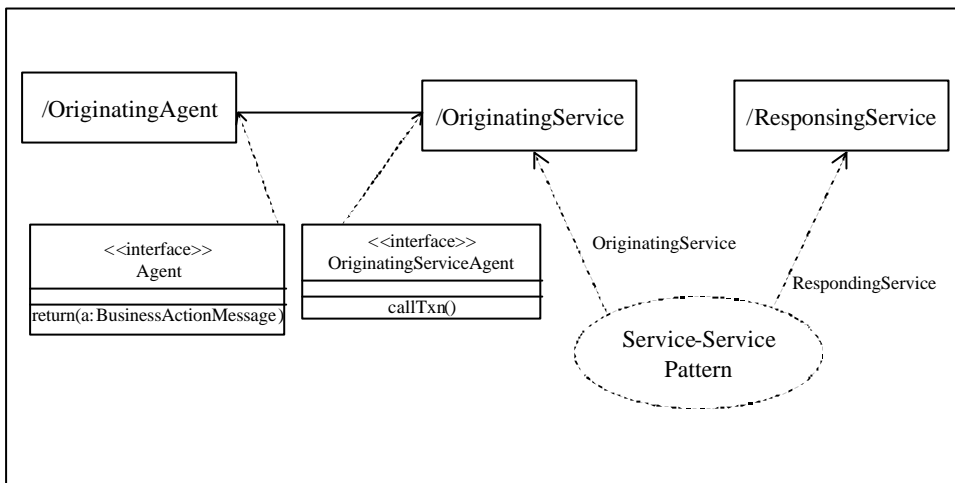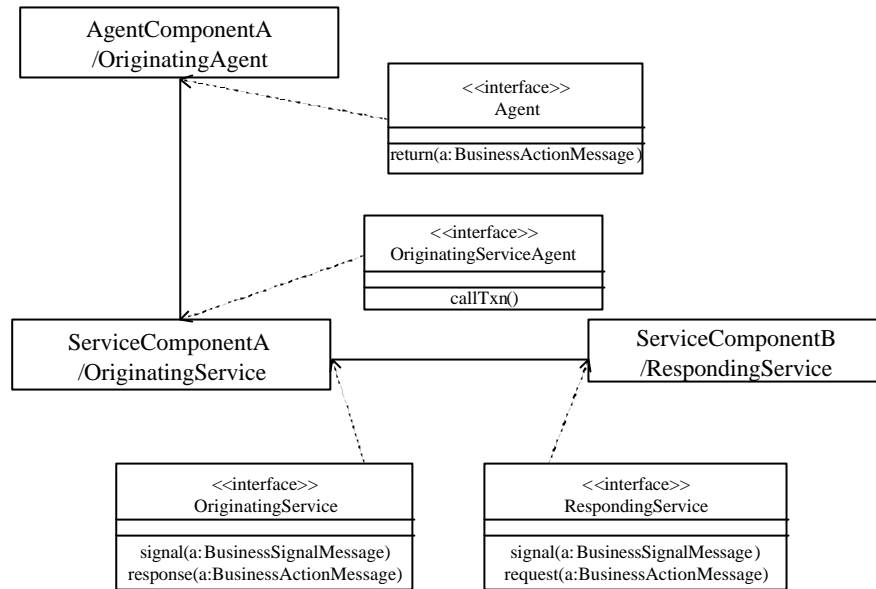| ServiceComponentA /OriginatingService		ServiceComponentB /RespondingService

<<interface>> OriginatingService	<<interface>> RespondingService
signal(a:BusinessSignalMessage) response(a:BusinessActionMessage)	signal(a:BusinessSignalMessage) request(a:BusinessActionMessage)

5

6   **Figure 9-20   Service-Service pattern class diagram**

7
8   There are five variations within the Service-Service pattern:
9   1. Interaction Pattern A applies to the Business Transaction Pattern where time
10   to perform equals time to acknowledge acceptance and there is no
11   responding business document.
12   2. Interaction Pattern B also applies to the Business Transaction Pattern where
13   time to perform equals time to acknowledge acceptance and a responding
14   business document.
15   3. Interaction Pattern C also applies to the Business Transaction Pattern
16   where time to perform is greater than time to acknowledge acceptance.
17   4. Interaction Pattern D applies to the Query/Response, Request/Response,
18   and Request/Confirm Patterns.
19   5. Interaction Pattern E applies to the Information Distribution and Notification
20   Patterns.

1

2

3   <u>Business Transaction Activity</u>

4

5   Figure 9-21, Figure 9-22, and Figure 9-23 illustrate Interaction Patterns A, B,
6   and C respectively.

7

8

9

10   **Figure 9-21   Service-Service Interaction Pattern A**

1

Figure 9-22   Service-Service Interaction Pattern B

2

3

Figure 9-23 **Service-Service Interaction Pattern C**

1 <u>Query/Response, Request/Response, and Request/Confirm Activities</u>
2
3 Figure 9-24 illustrates Interaction Pattern D.
4
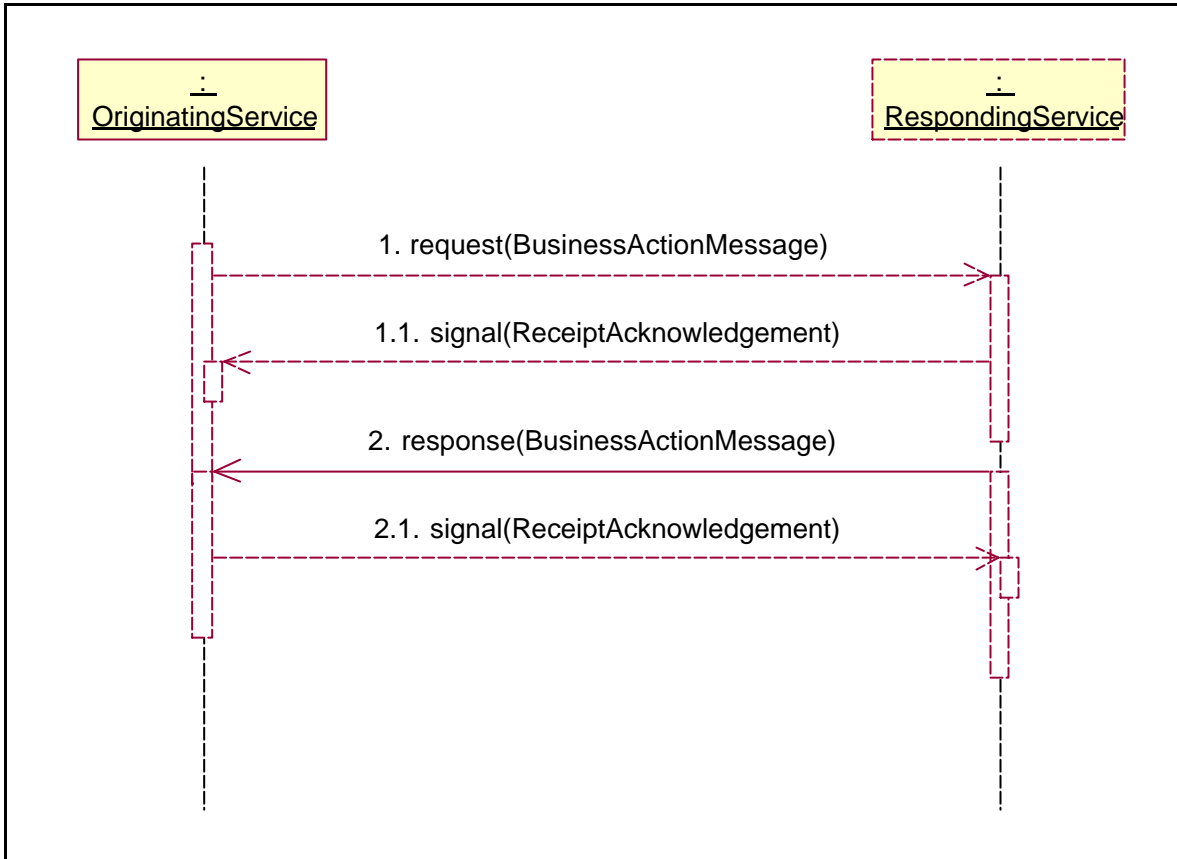
5

6 **Figure 9-24   Service-Service Interaction Pattern D**

1

2

3   Information Distribution and Notification Activities

4

5   Figure 9-25 illustrates Interaction Pattern E.

6
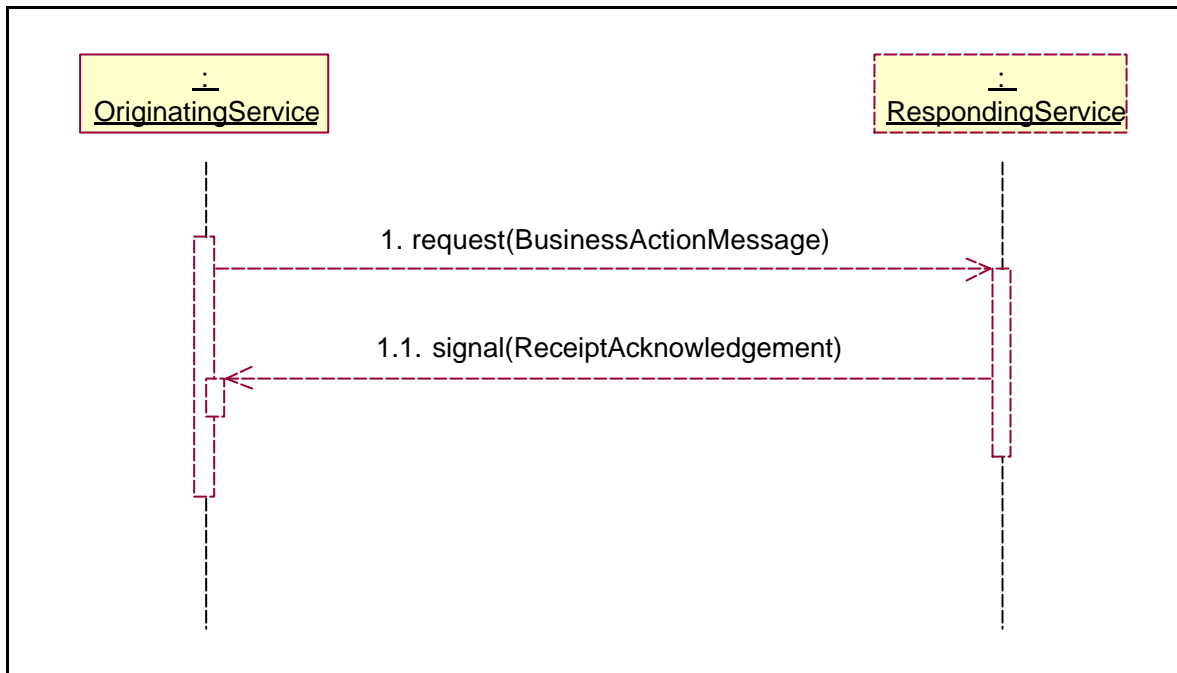
7

8   **Figure 9-25   Service-Service Interaction Pattern E**

1
2
9.4.2 *Agent-Service-Service*

4
5 Figure 9-26 illustrates the Agent-Service-Service business service interaction
6 pattern used in the business transaction patterns of Section 8.3.
7
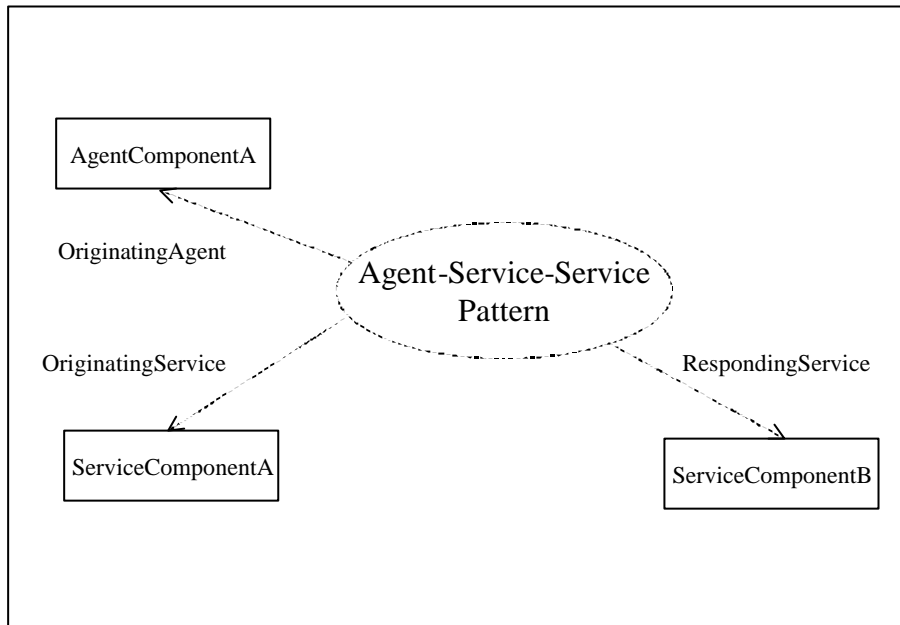
8
9 **Figure 9-26   Agent-Service-Service Pattern**

10

1
2
3   Figure 9-27 shows the class diagram for the Agent-Service-Service pattern, and
4   Figure 9-28 shows the class diagram for the Agent-Service-Service unfolded
5   from the base pattern.

6
7

8   **Figure 9-27   Agent-Service-Service pattern class diagram**

1

AgentComponentA
/OriginatingAgent

<<interface>>
Agent

return(a: BusinessActionMessage)

<<interface>>
OriginatingServiceAgent

callTxn()

ServiceComponentA
/OriginatingService

ServiceComponentB
/RespondingService

<<interface>>
OriginatingService

signal(a: BusinessSignalMessage)
response(a: BusinessActionMessage)

<<interface>>
RespondingService

signal(a: BusinessSignalMessage)
request(a: BusinessActionMessage)

2
3

4   **Figure 9-28   Agent-Service-Service pattern class diagram unfolded**

5
6
7
8
9   There are five variations within the Agent-Service-Service pattern:
10   1. Interaction Pattern A applies to the Business Transaction Pattern where time
11   to perform equals time to acknowledge acceptance and there is no
12   responding business document.
13   2. Interaction Pattern B also applies to the Business Transaction Pattern where
14   time to perform equals time to acknowledge acceptance and a responding
15   business document.
16   3. Interaction Pattern C also applies to the Business Transaction Pattern
17   where time to perform is greater than time to acknowledge acceptance.
18   4. Interaction Pattern D applies to the Query/Response, Request/Response,
19   and Request/Confirm Patterns.
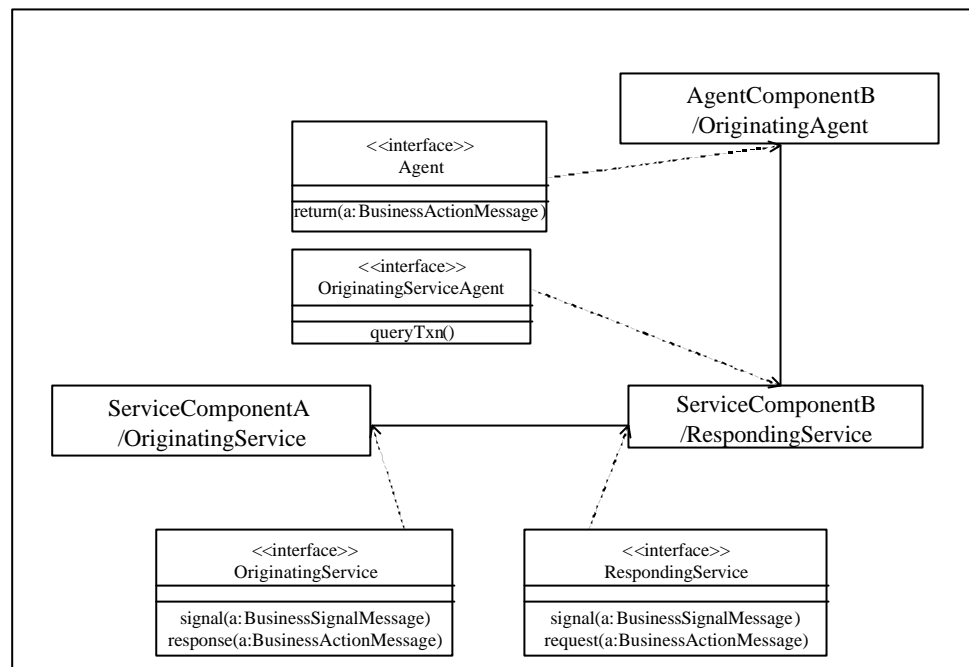20   5. Interaction Pattern E applies to the Information Distribution and Notification
21   Patterns.

1
2
3
4   <u>Business Transaction Activity</u>
5
6   Figure 9-29, Figure 9-30, and Figure 9-31 illustrate Interaction Patterns A, B,
7   and C respectively.
8

9
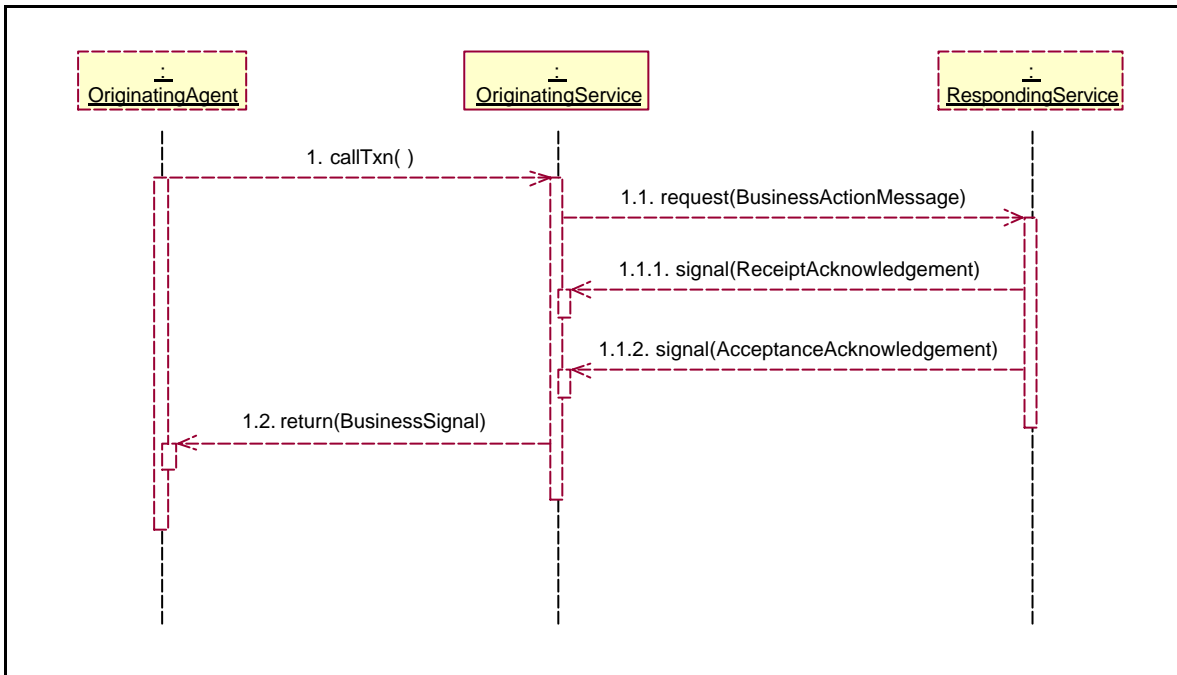
10   **Figure 9-29   Agent-Service-Service Interaction Pattern A**

11
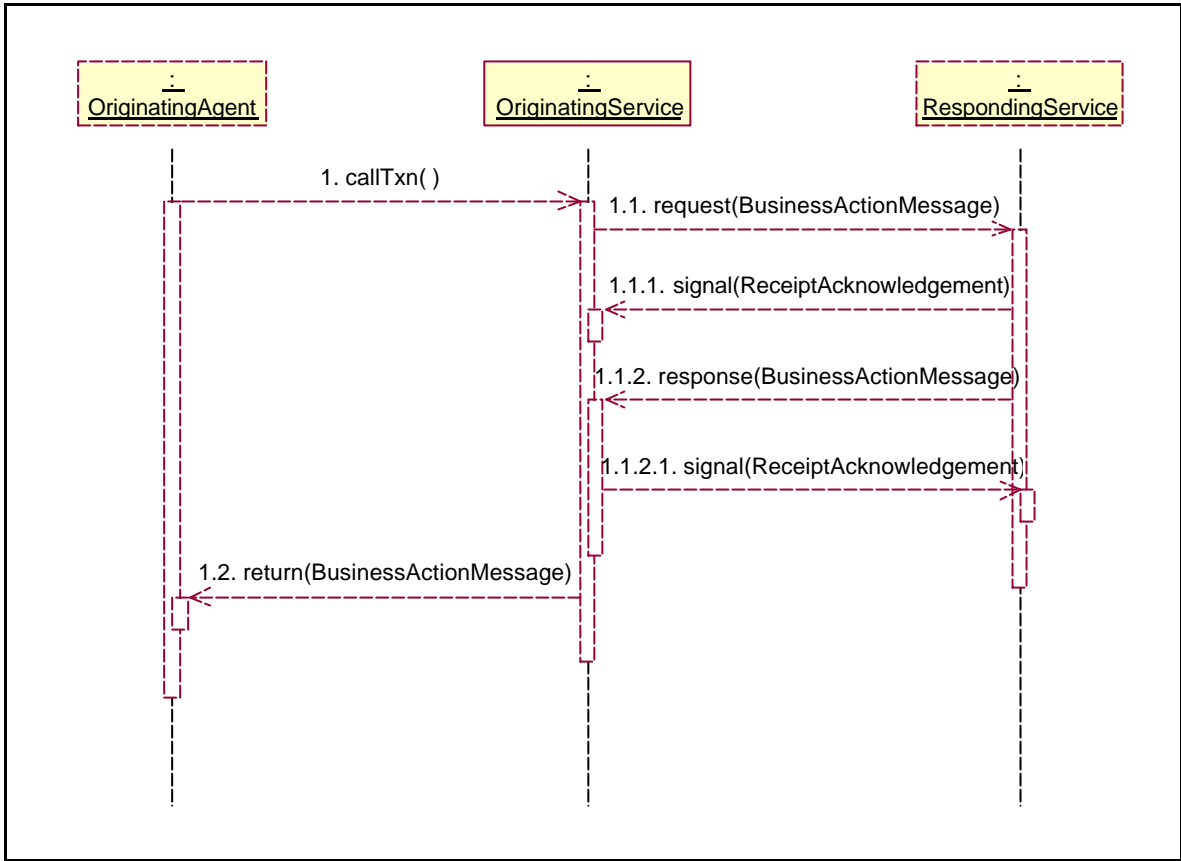
Figure 9-30   Agent-Service-Service Interaction Pattern B
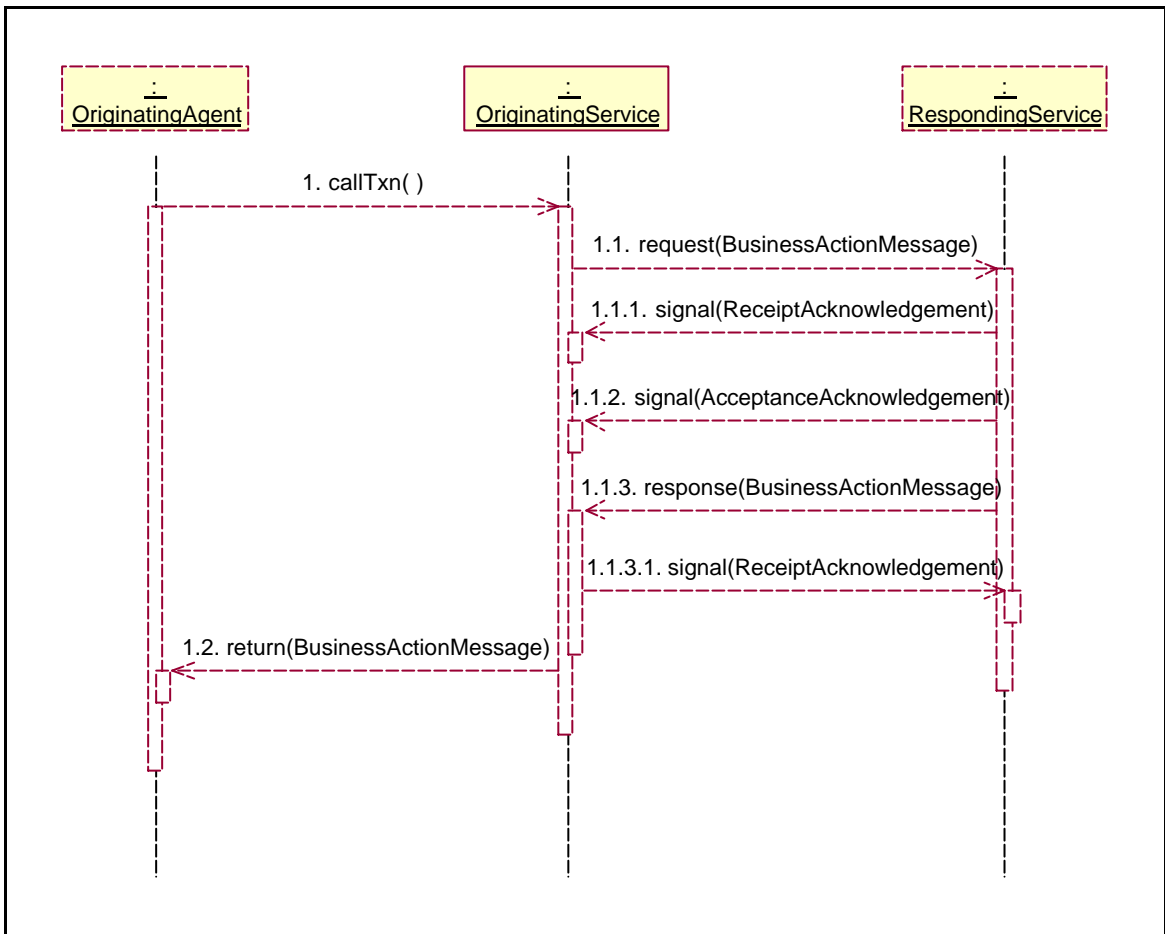
Figure 9-31   Agent-Service-Service Interaction Pattern C

1
2
3   Query/Response, Request/Response, and Request/Confirm Activities
4
5   Figure 9-32 illustrates Interaction Pattern D.

6
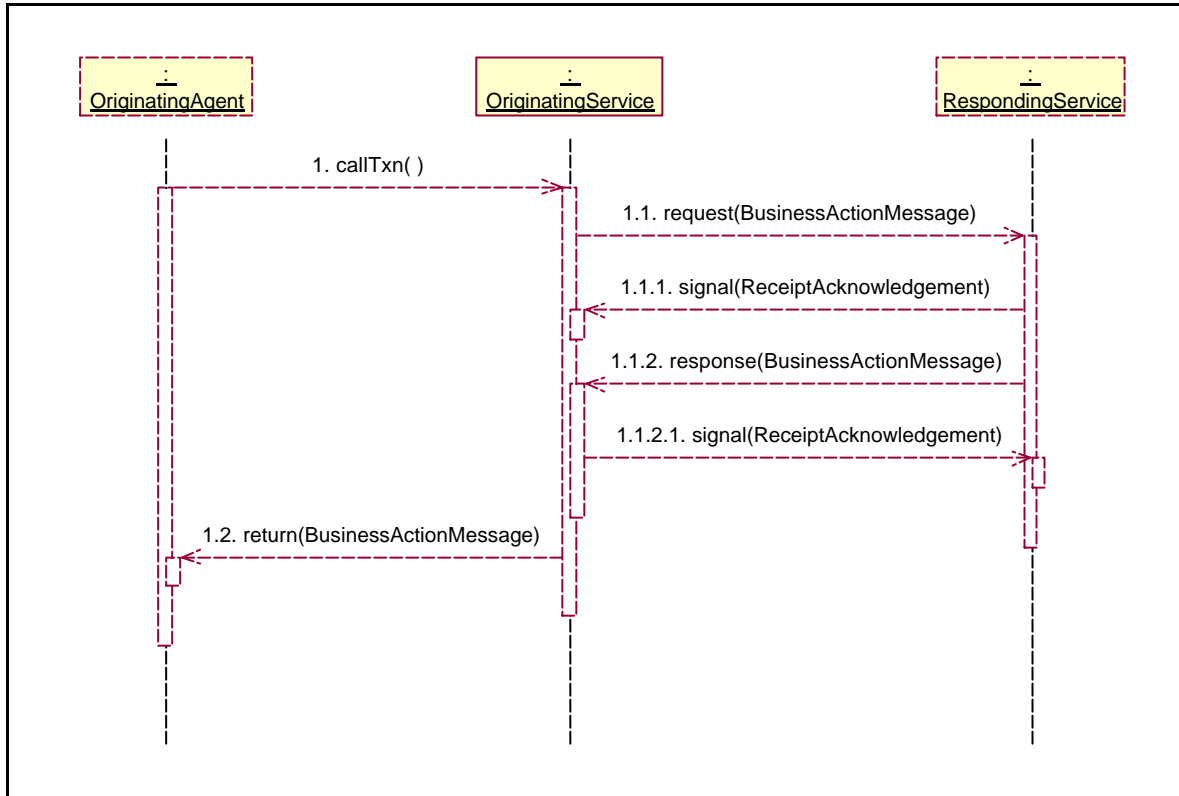
7   **Figure 9-32   Agent-Service-Service Interaction Pattern D**

1
2   <u>Information Distribution and Notification Activities</u>
3
4   Figure 9-33 illustrates Interaction Pattern E.
5
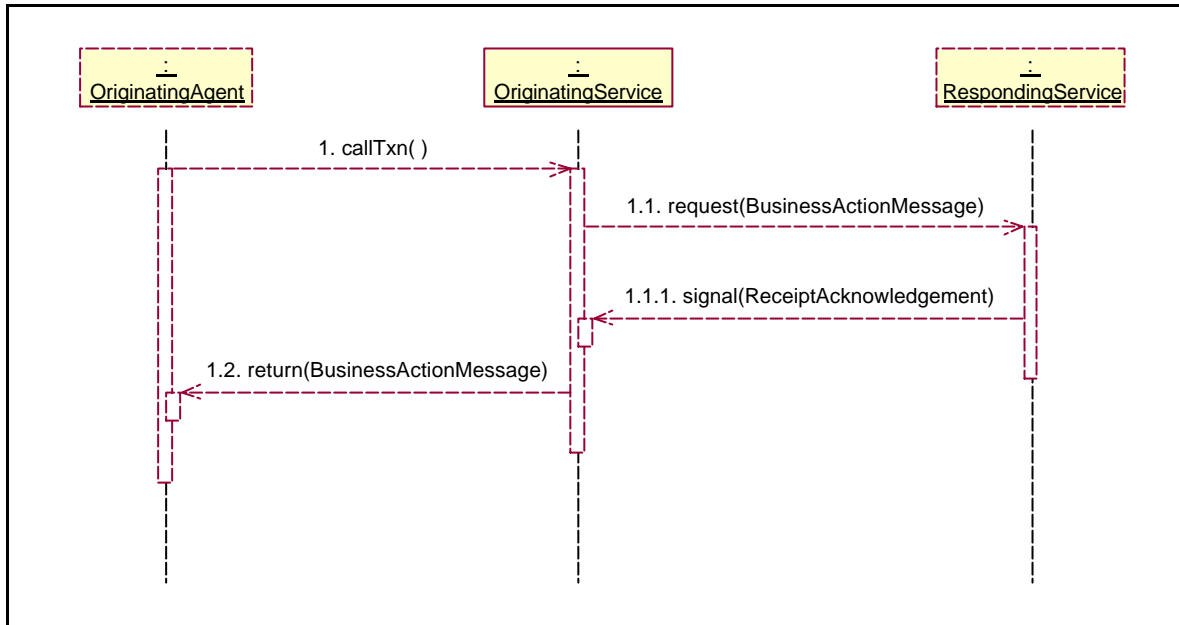
6

7   **Figure 9-33   Agent-Service-Service Interaction Pattern E**

1
2 ### 9.4.3 *Service-Service-Agent*
3
4   Figure 9-34 illustrates the Service-Service-Agent business service interaction
5   pattern used in the business transaction patterns of Section 8.3.
6
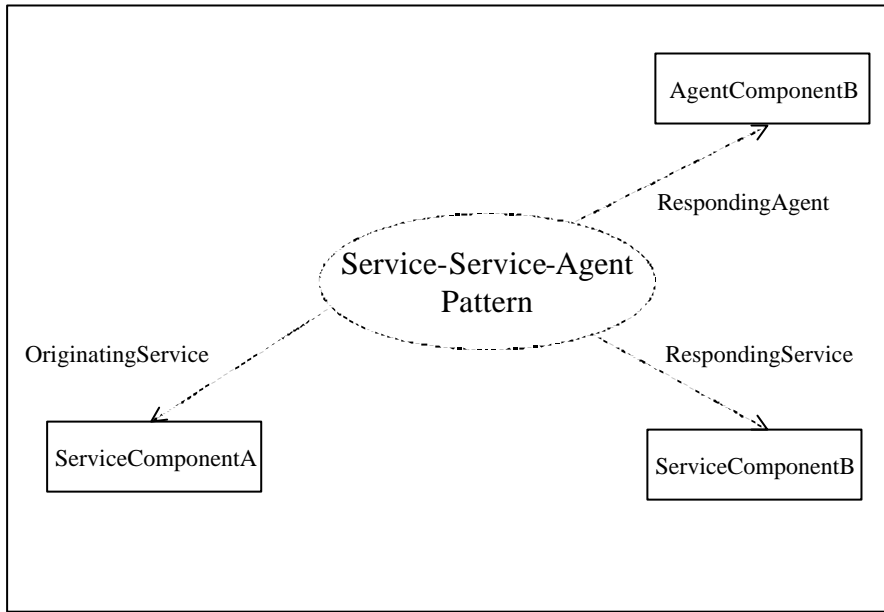
7
8   **Figure 9-34   Service-Service-Agent pattern**

1
2
3   Figure 9-35 shows the class diagram for the Service-Service-Agent pattern, and
4   Figure 9-36 shows the class diagram for the Service-Service-Agent unfolded
5   from the base pattern.
6

7
8   **Figure 9-35**   **Service-Service-Agent pattern class diagram**

1

2
3

Figure 9-36   Service-Service-Agent pattern class diagram unfolded

4
5
6
7
8
9
10
11   There are five variations within the Service-Service-Agent pattern:
12   1. Interaction Pattern A applies to the Business Transaction Pattern where time
13   to perform equals time to acknowledge acceptance and there is no
14   responding business document.
15   2. Interaction Pattern B also applies to the Business Transaction Pattern where
16   time to perform equals time to acknowledge acceptance and a responding
17   business document.
18   3. Interaction Pattern C also applies to the Business Transaction Pattern
19   where time to perform is greater than time to acknowledge acceptance.
20   4. Interaction Pattern D applies to the Query/Response, Request/Response,
21   and Request/Confirm Patterns.
22   5. Interaction Pattern E applies to the Information Distribution and Notification
23   Patterns.

1
2
3   Business Transaction Activity
4
5   Figure 9-37, Figure 9-38, and Figure 9-39 illustrate Interaction Patterns A, B,
6   and C respectively.
7

8

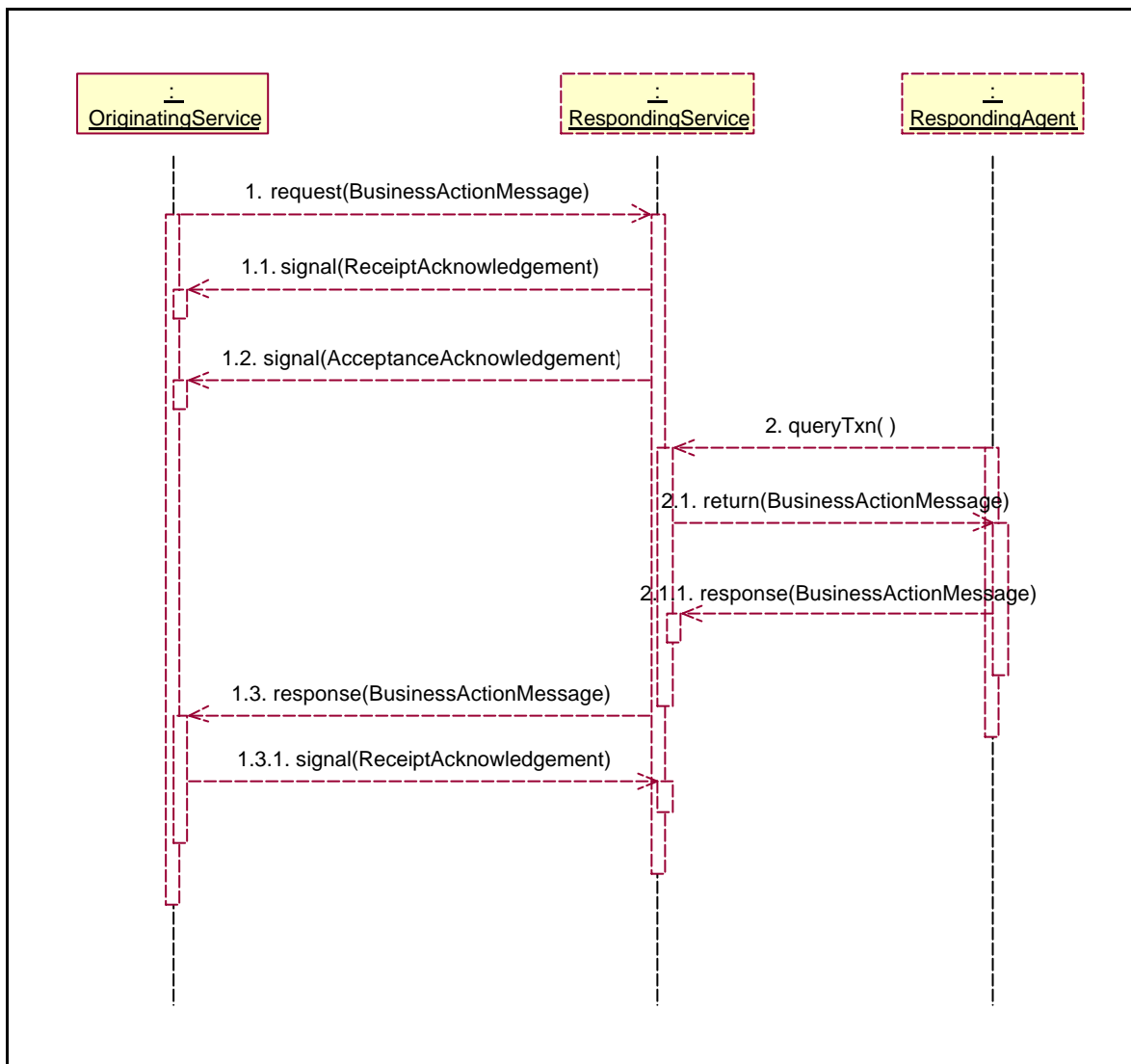9   **Figure 9-37   Service-Service-Agent Interaction Pattern A**

10

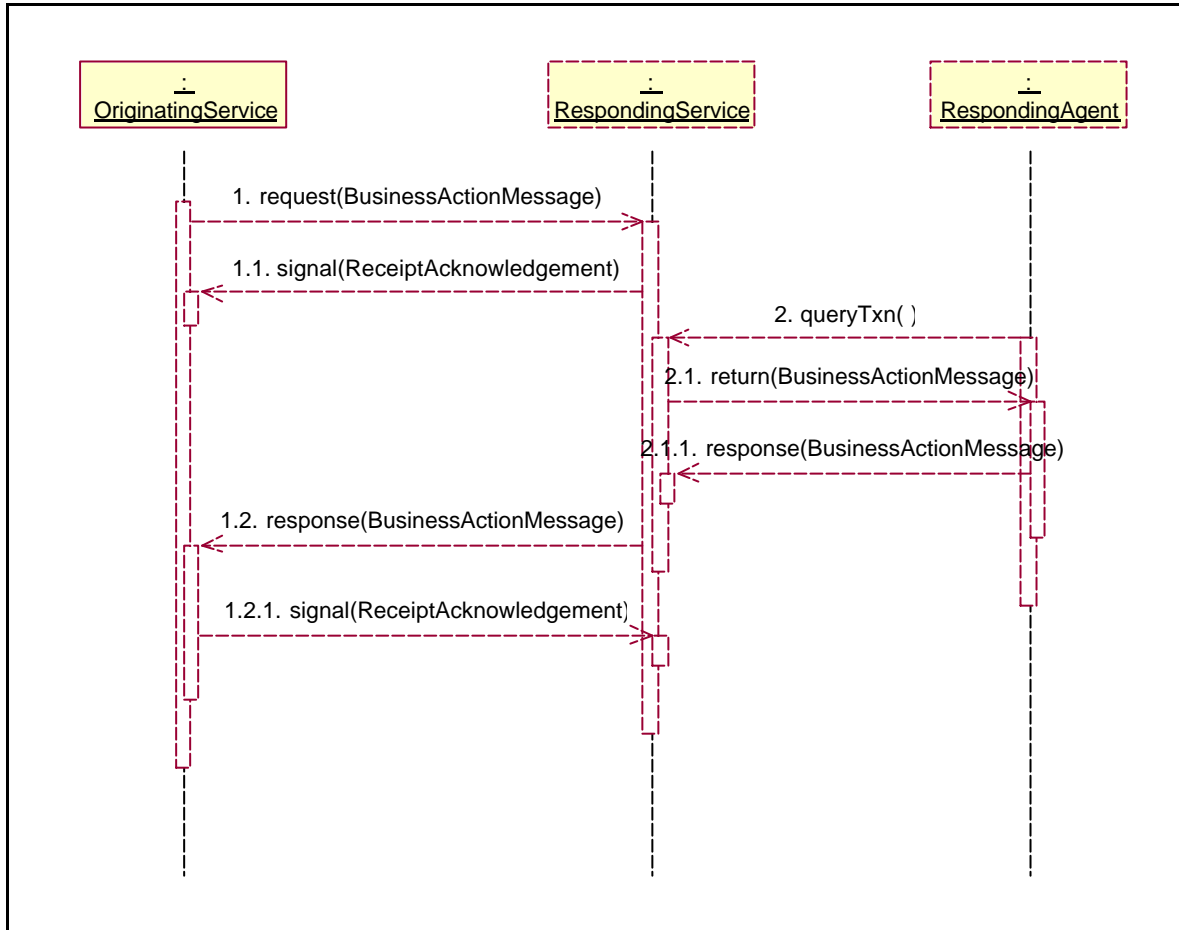**Figure 9-38   Service-Service-Agent Interaction Pattern B**

Figure 9-39   Service-Service-Agent Interaction Pattern C

1   <u>Query/Response, Request/Response, and Request/Confirm Activities</u>
2
3   Figure 9-40 illustrates Interaction Pattern D.
4
5

7   **Figure 9-40   Service-Service-Agent Interaction Pattern D**

1 <u>Information Distribution and Notification Activities</u>

2

3 Figure 9-41 illustrates Interaction Pattern E.

4

5

6
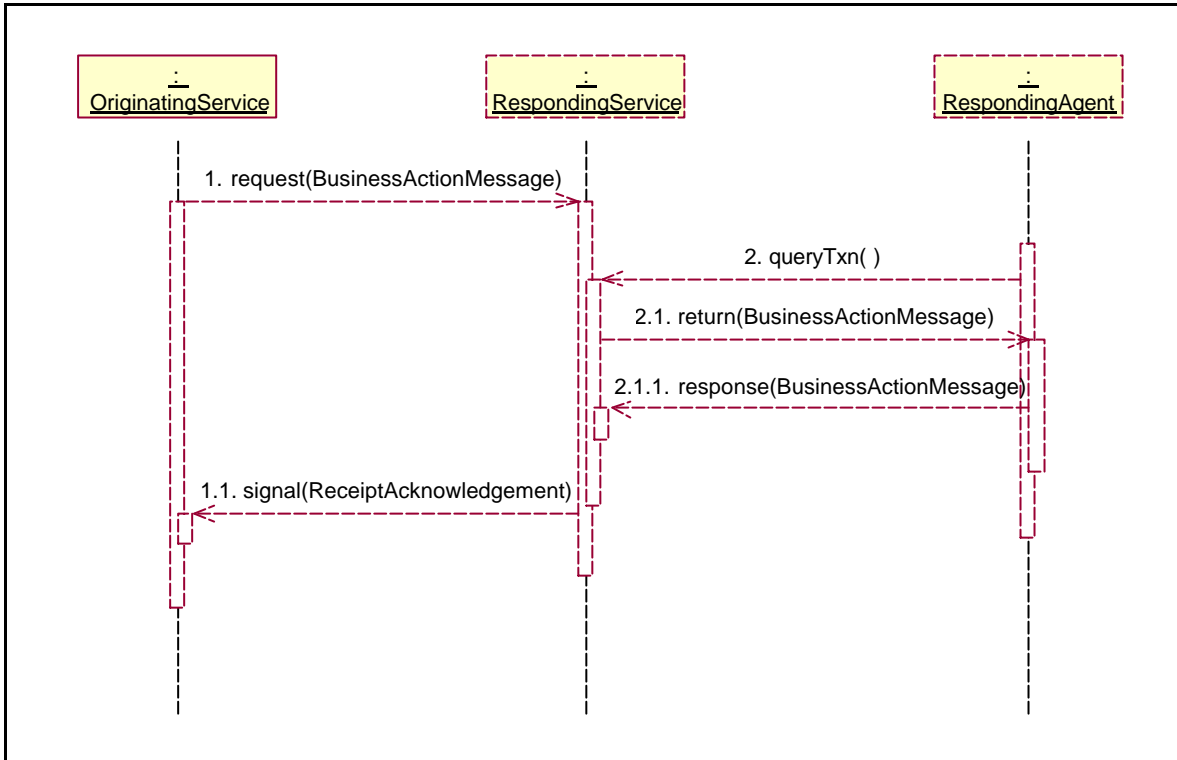
1 ***9.4.4   Service-Agent-Service***

2

3   Figure 9-42 illustrates the Service-Agent-Service business service interaction
4   pattern used in the business transaction patterns of Section 8.3.

5

6

AgentComponentA

AgentComponentB

OriginatingAgent

RespondingAgent

Service-Agent-Service
Pattern

OriginatingService

RespondingService

ServiceComponentA

ServiceComponentB
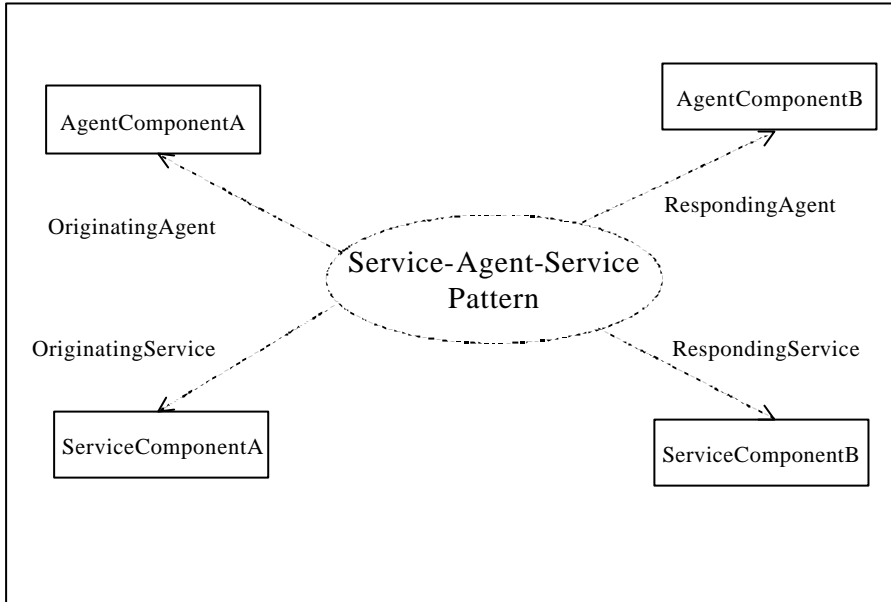
7

8   **Figure 9-42   Service-Agent-Service pattern**

1
2 Figure 9-43 shows the class diagram for the Service-Agent-Service pattern, and
3 Figure 9-44 shows the class diagram for the Service-Agent-Service unfolded
4 from the base pattern.
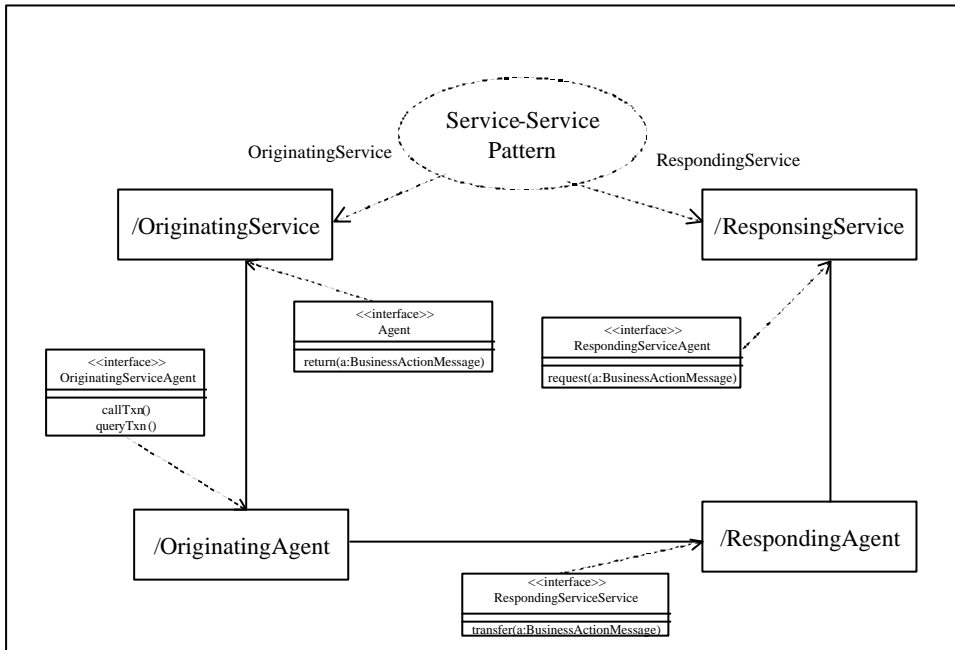
5

6 **Figure 9-43 Service-Agent-Service pattern class diagram**

1
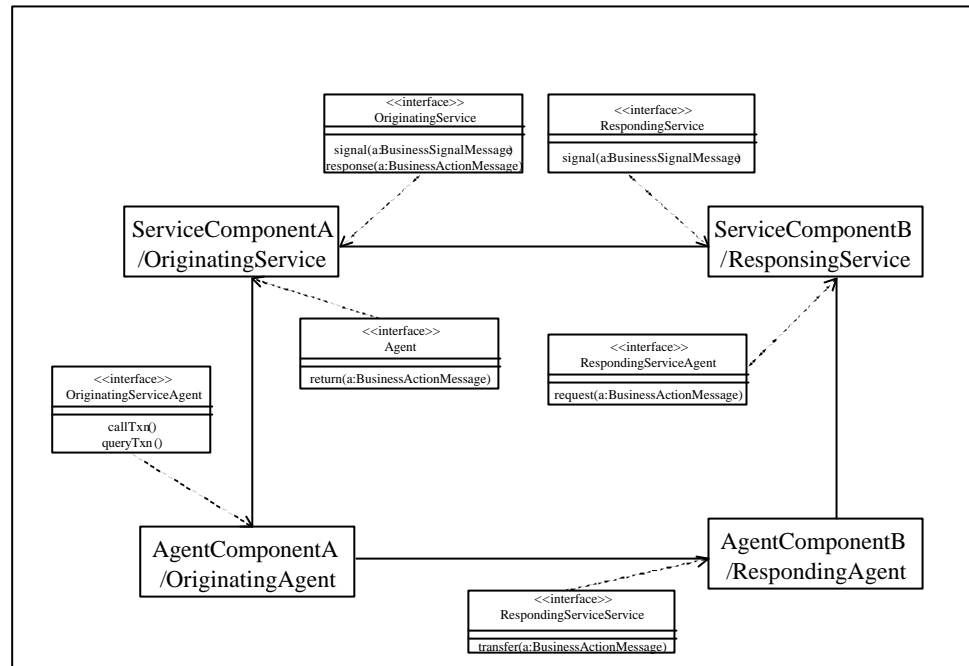
4   **Figure 9-44   Service-Agent-Service pattern class diagram unfolded**

5

6   There are six variations within the Service-Agent-Service pattern:
7   1. Interaction Pattern A applies to the Business Transaction Pattern where time
8   to perform equals time to acknowledge acceptance and there is no
9   responding business document.
10   2. Interaction Pattern B also applies to the Business Transaction Pattern where
11   time to perform equals time to acknowledge acceptance and a responding
12   business document.
13   3. Interaction Pattern C also applies to the Business Transaction Pattern
14   where time to perform is greater than time to acknowledge acceptance.
15   4. Interaction Pattern D applies to the Query/Response and
16   Request/Response Patterns.
17   5. Interaction Pattern E applies to the Request/Confirm Pattern.
18   6. Interaction Pattern F applies to the Information Distribution and Notification
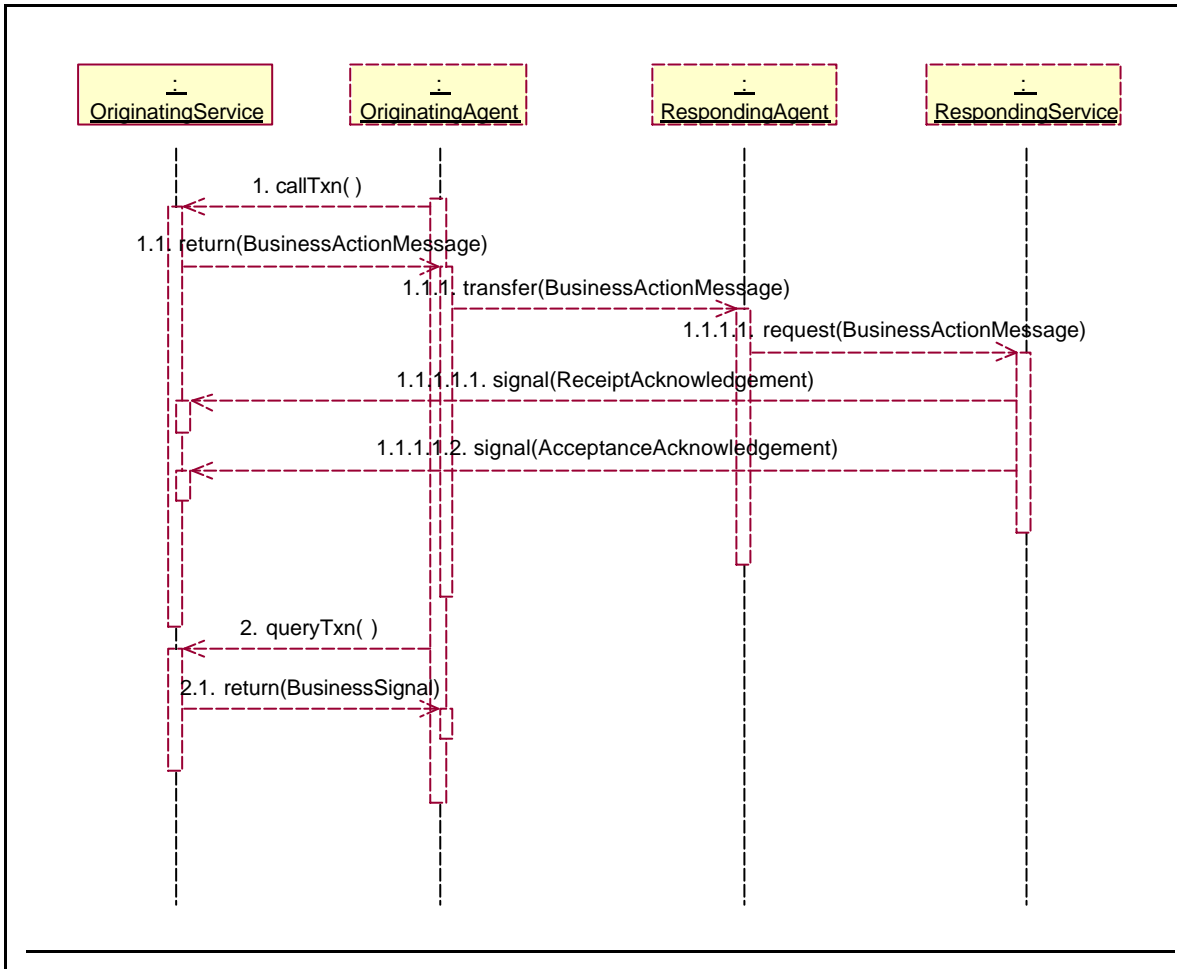19   Patterns.

2

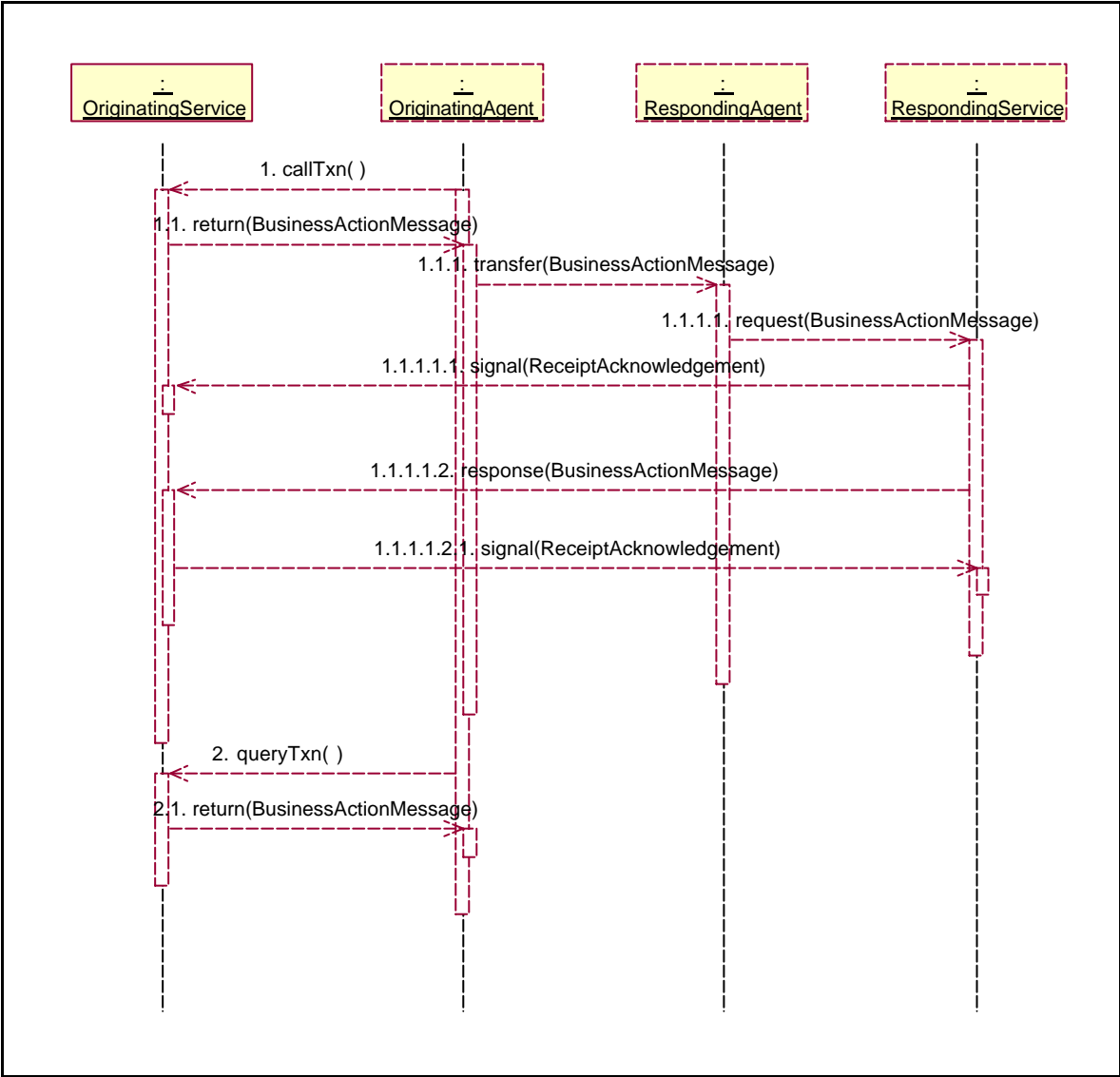3 <u>Business Transaction Activity</u>

4

5 Figure 9-45, Figure 9-46, and Figure 9-47 illustrate Interaction Patterns A, B,
6 and C respectively.

7

8

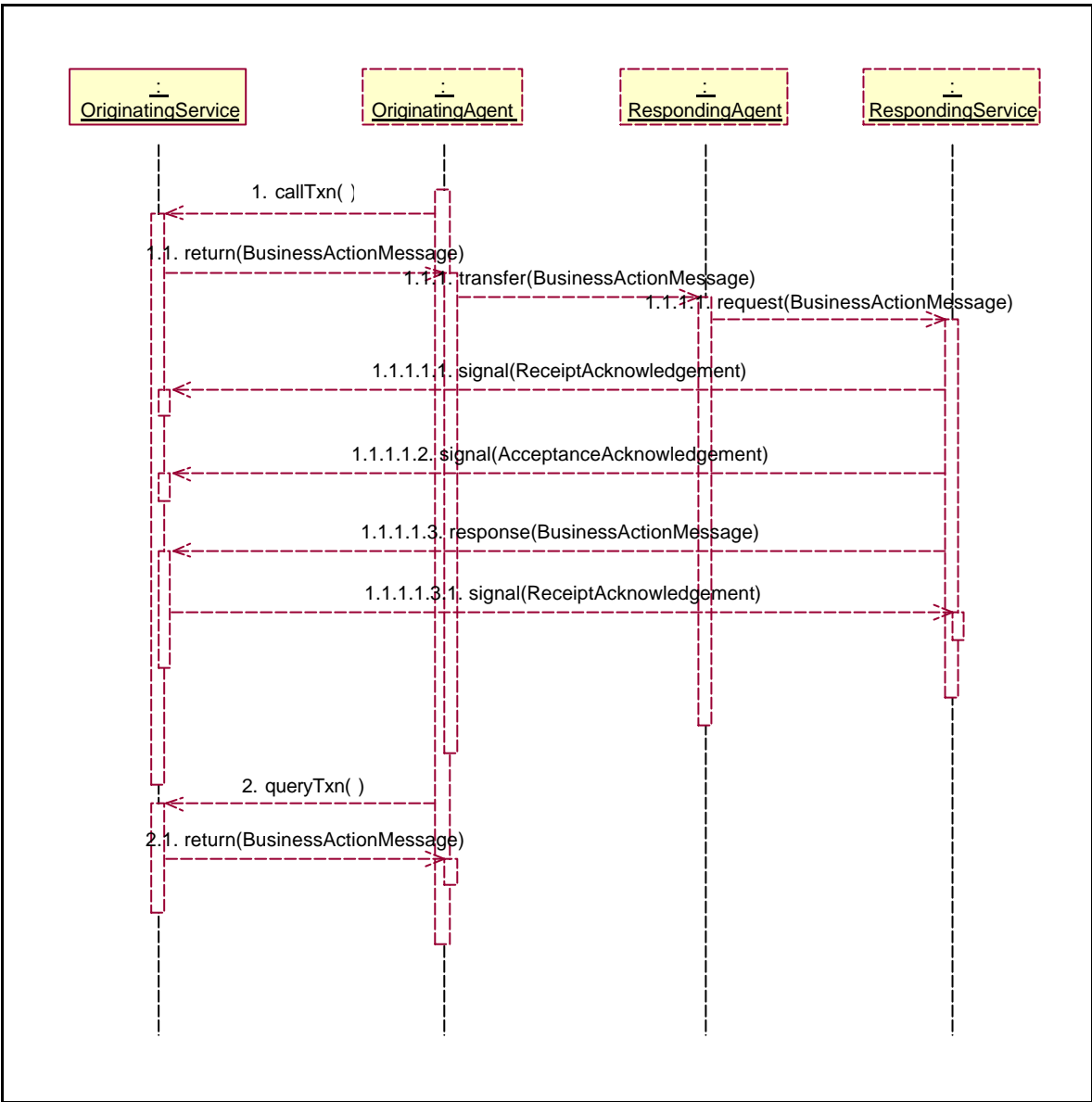9 **Figure 9-45   Service-Agent-Service Interaction Pattern A**

10

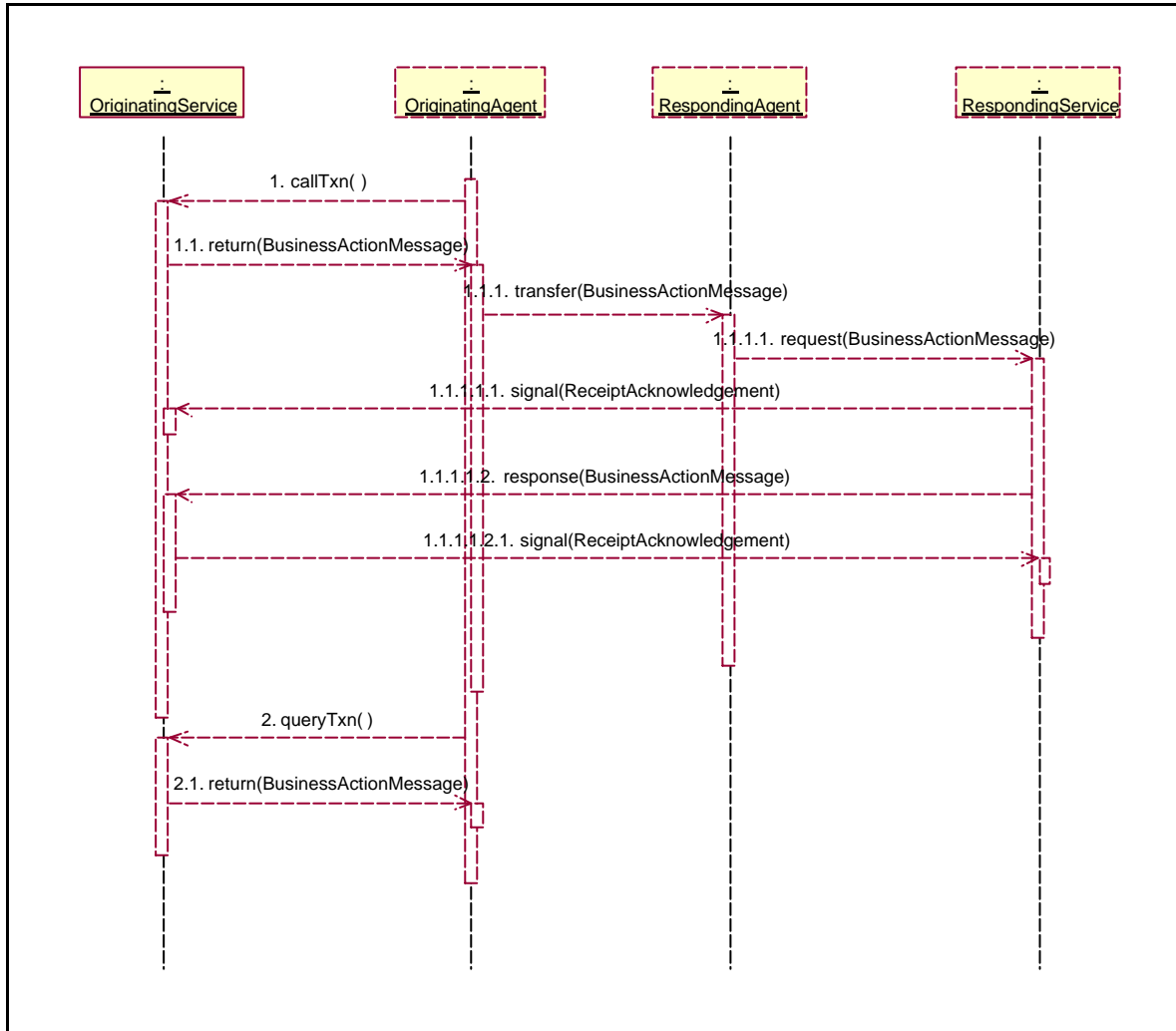**Figure 9-46   Service-Agent-Service Interaction Pattern B**

Figure 9-47   Service-Agent-Service Interaction Pattern C

1

2

1   Query/Response and Request/Response Activities

2

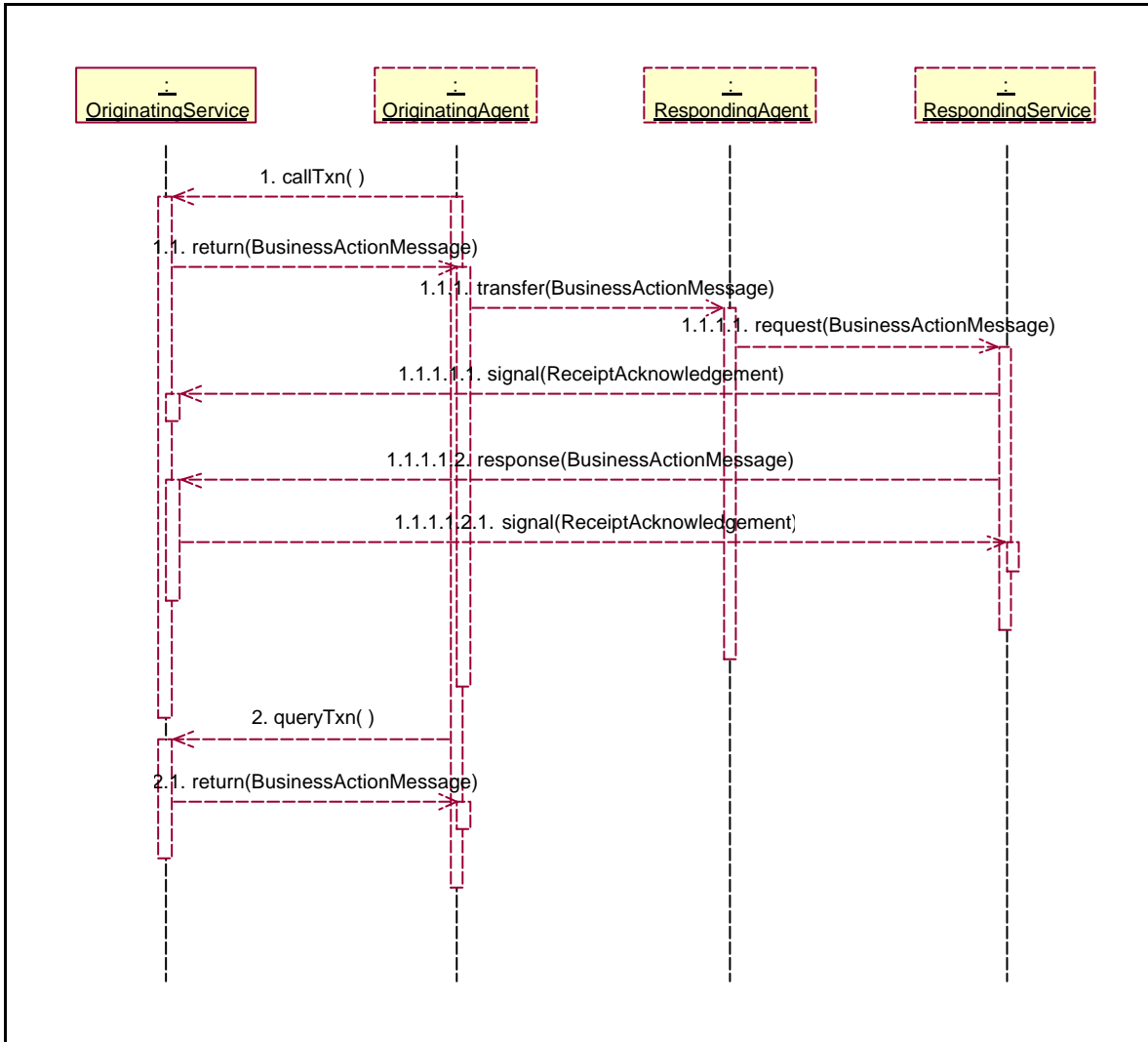3   Figure 9-48 illustrates Interaction Pattern D.

4

5

6

7   **Figure 9-48   Service-Agent-Service Interaction Pattern D**

1 <u>Request/Confirm Activity</u>

2

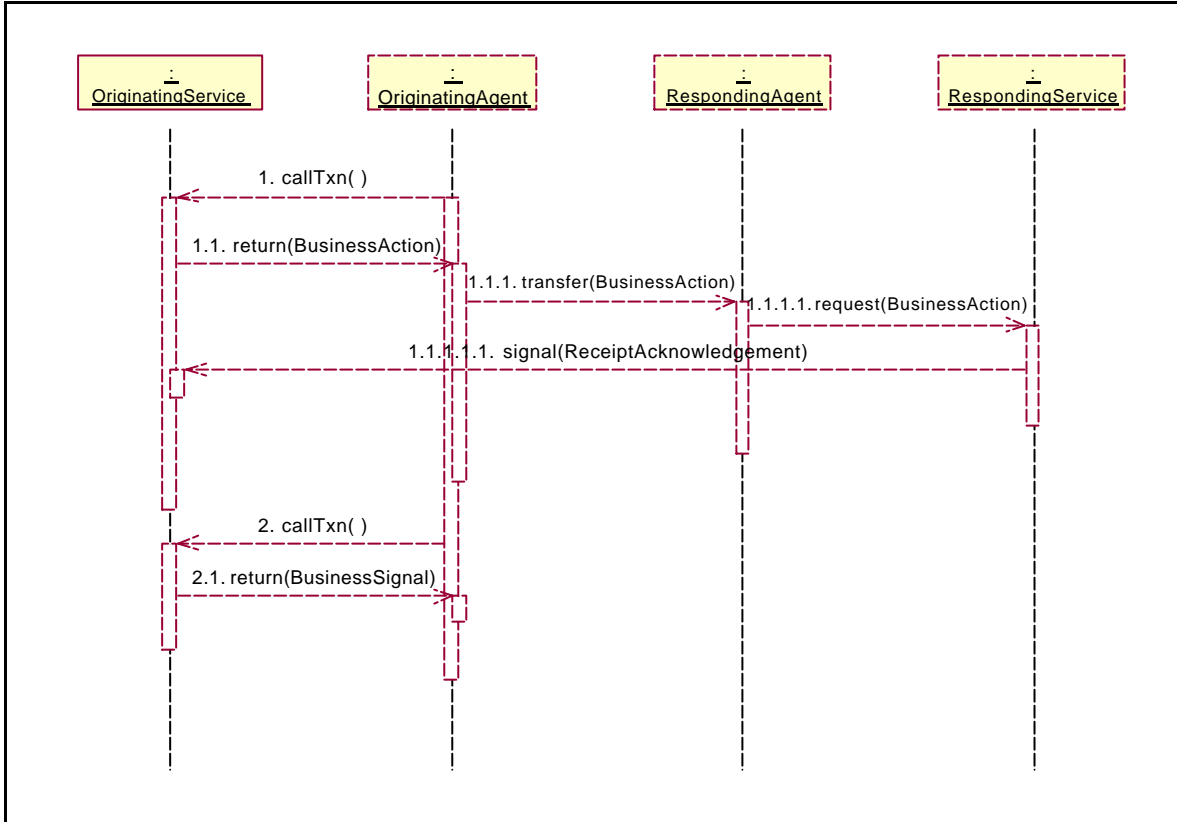3 Figure 9-49 illustrates Interaction Pattern E.

4

5

7 **Figure 9-49 Service-Agent-Service interaction Pattern E**

1
2
3 <u>Information Distribution and Notification Activities</u>
4
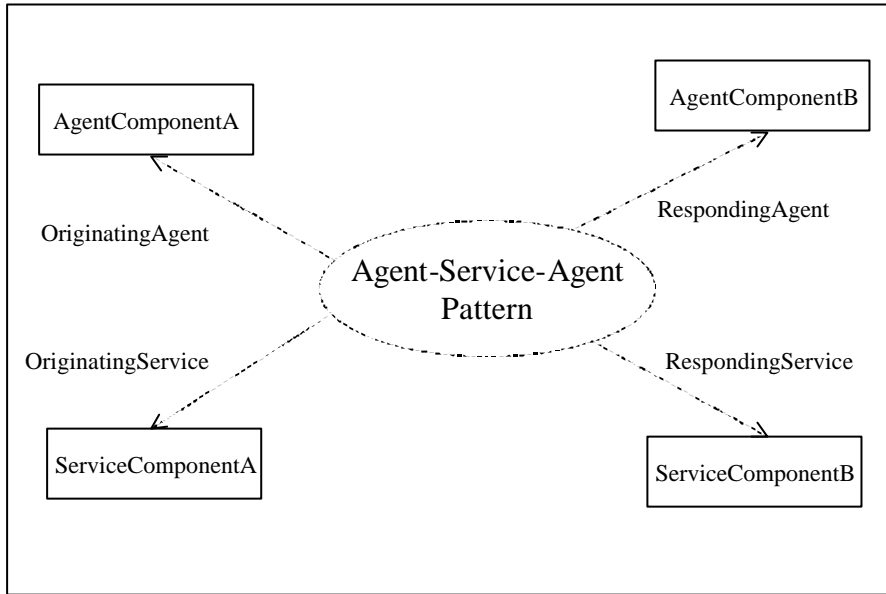5 Figure 9-50 illustrates Interaction Pattern F.

6

8
9

1
2 **_9.4.5   Agent-Service-Agent_**
3
4   Figure 9-51 illustrates the Agent-Service-Agent business service interaction
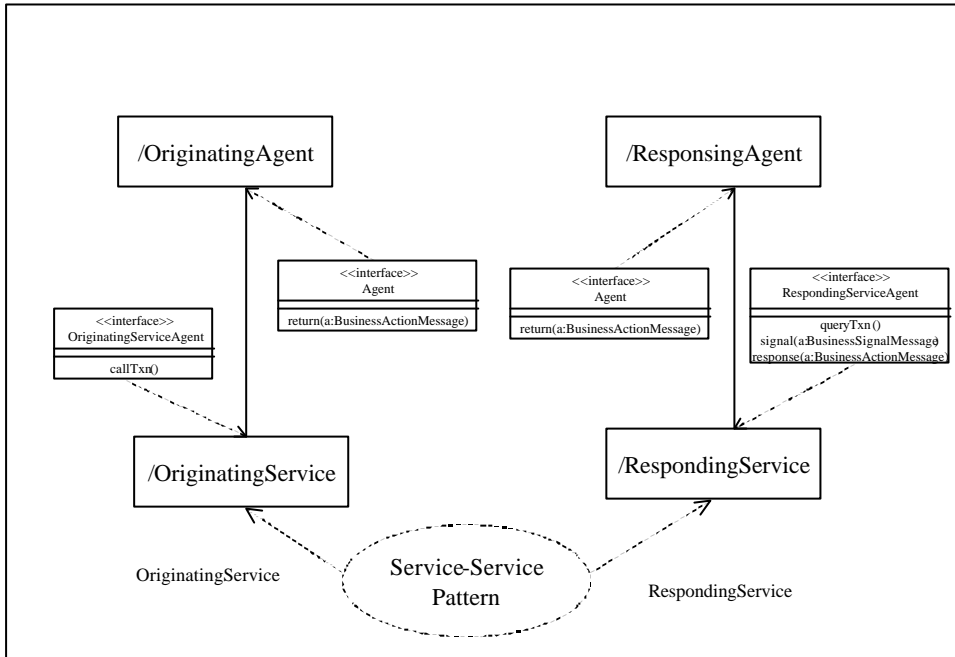5   pattern used in the business transaction patterns of Section 8.3.
6
7

8

9   **Figure 9-51   Agent-Service-Agent pattern**

1
2   Figure 9-52 shows the class diagram for the Agent-Service-Agent pattern, and
3   Figure 9-53 shows the class diagram for the Agent-Service-Agent unfolded from
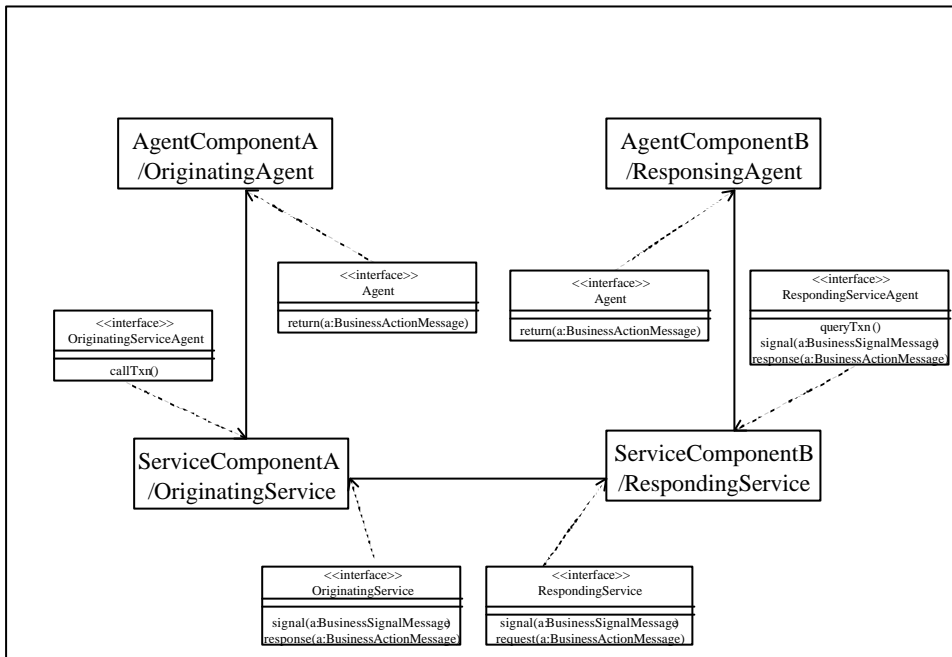4   the base pattern.
5

6

7   **Figure 9-52   Agent-Service-Agent class diagram**

5

Figure 9-53   Agent-Service-Agent class diagram unfolded

7

There are five variations within the Agent-Service-Agent pattern:
1. Interaction Pattern A applies to the Business Transaction Pattern where time to perform equals time to acknowledge acceptance and there is no responding business document.
2. Interaction Pattern B also applies to the Business Transaction Pattern where time to perform equals time to acknowledge acceptance and a responding business document.
3. Interaction Pattern C also applies to the Business Transaction Pattern where time to perform is greater than time to acknowledge acceptance.
4. Interaction Pattern D applies to the Query/Response, Request/Response, and Request/Confirm Patterns.
5. Interaction Pattern E applies to the Information Distribution and Notification Patterns.
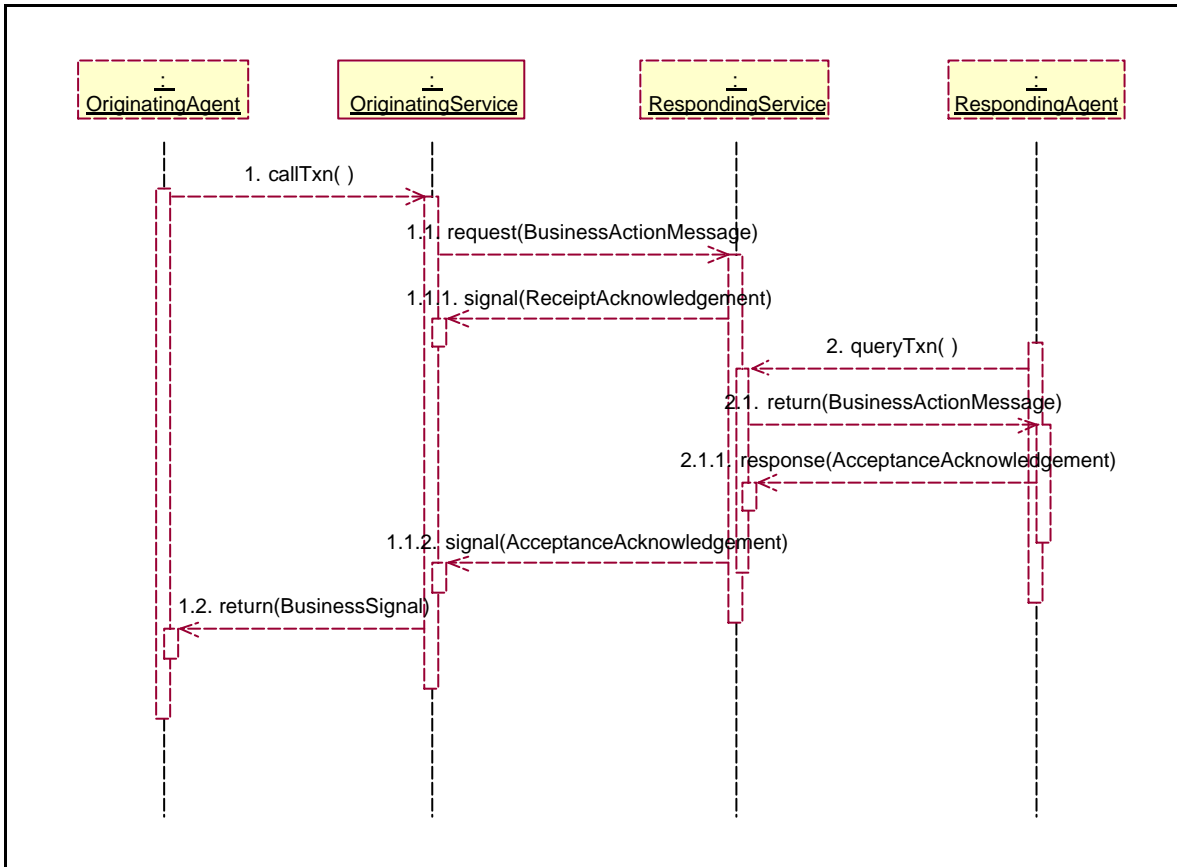
1
2
3
4   <u>Business Transaction Activity</u>
5
6   Figure 9-54, Figure 9-55, and Figure 9-56 illustrate Interaction Patterns A, B,
7   and C respectively.

8

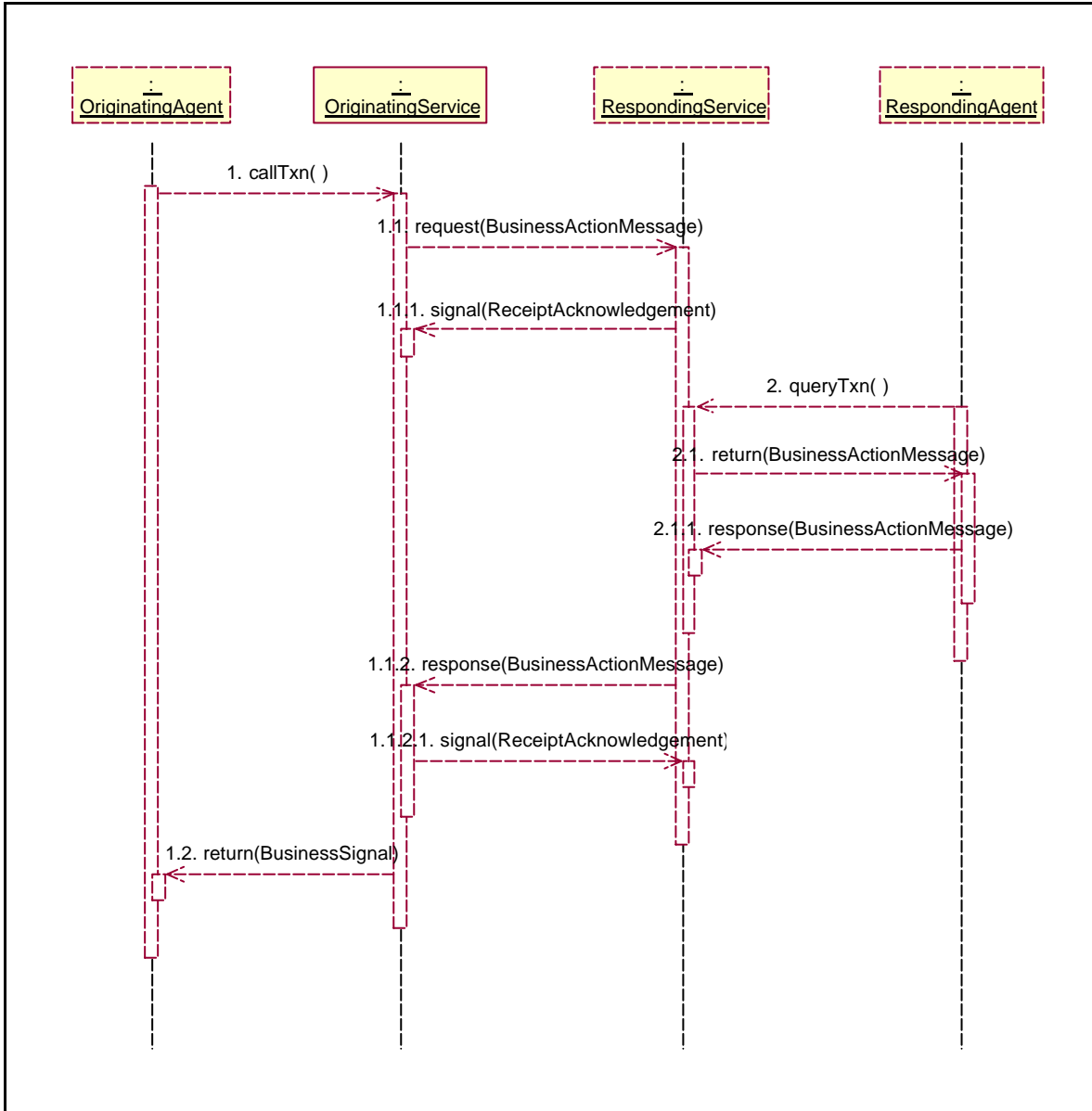9   **Figure 9-54   Agent-Service-Agent Interaction Pattern A**

1
2

Figure 9-55   Agent-Service-Agent Interaction Pattern B

3

4

5
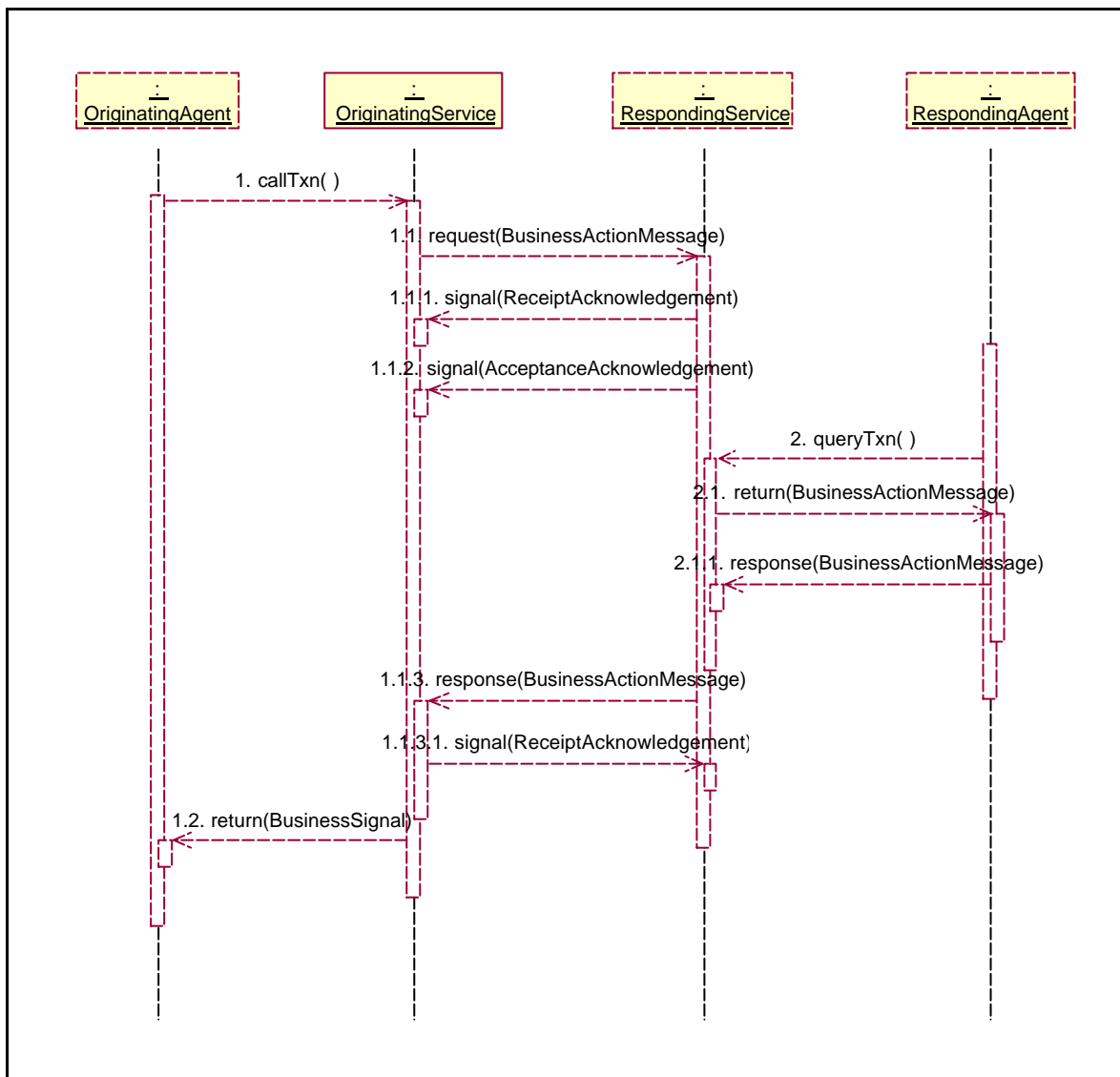
Figure 9-56   Agent-Service-Agent Interaction Pattern C

1

2

3   <u>Query/Response, Request/Response, and Request/Confirm Activities</u>

4

5   Figure 9-57 illustrates Interaction Pattern D.

6
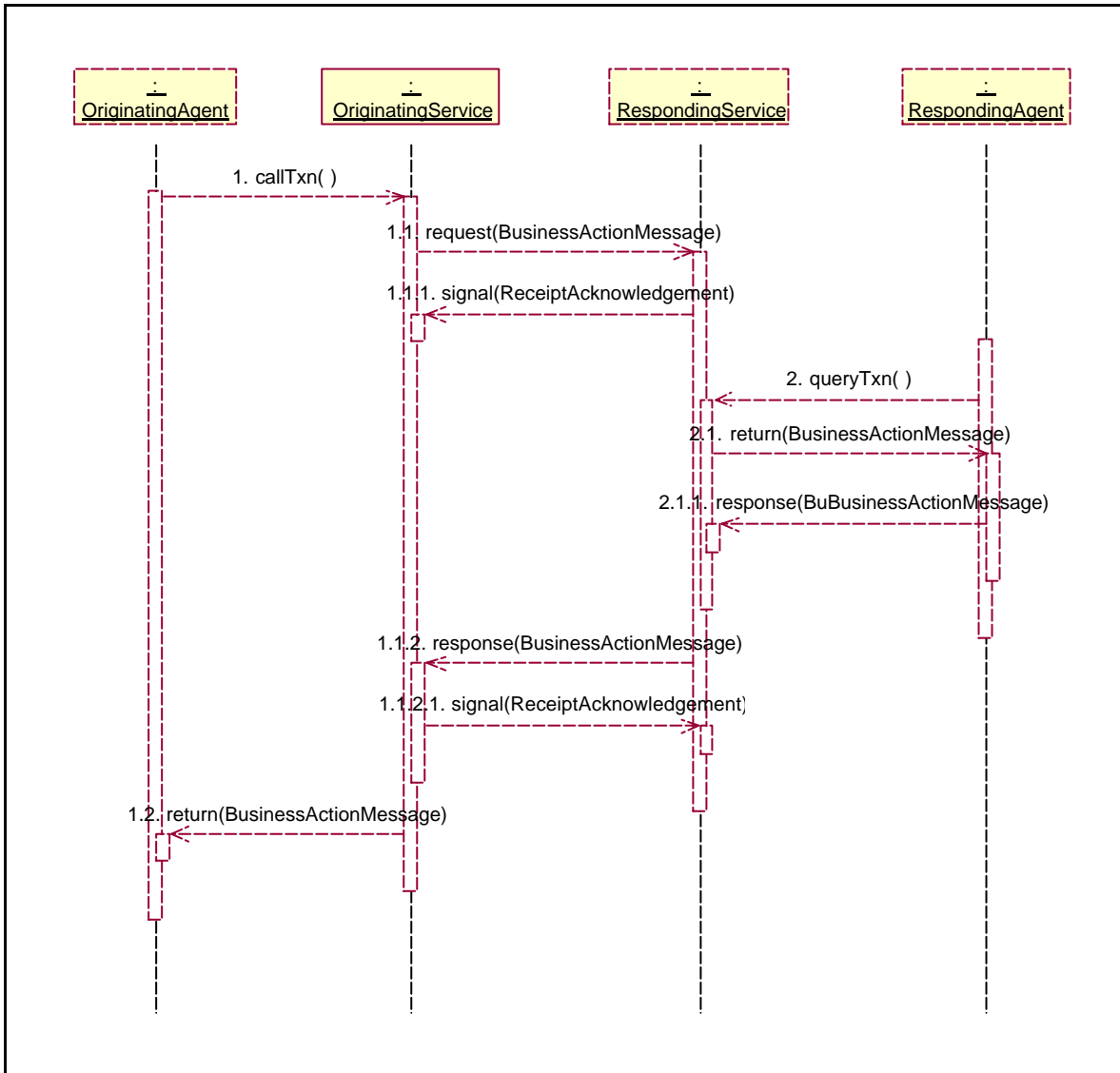
8   **Figure 9-57   Agent-Service-Agent Interaction Pattern D**

1

2

3   Information Distribution and Notification Activities

4

5   Figure 9-58 illustrates Interaction Pattern E.

6

7

8   **Figure 9-58   Agent-Service-Agent Interaction Pattern E**

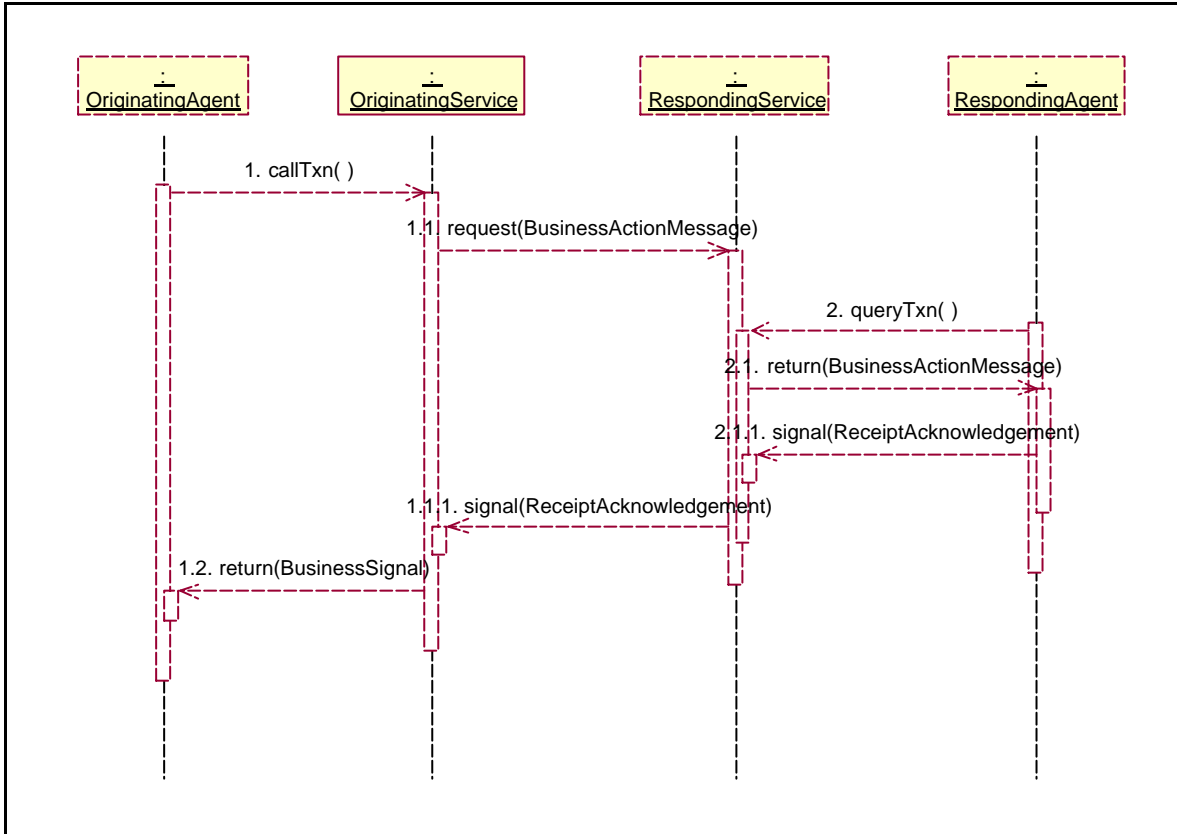## 9.5 Business Information Structure Design Patterns

9.5.1 The Reference Design Pattern

Business entity containers can reference themselves and other entities by explicitly modeling the reference association as an entity with association properties. As shown in Figure 9-59, the reference association (*SubComponent*) should minimally contain cardinality properties and a name that has a semantic definition specifying the relationship between the related entities. This design pattern is useful for reusing common sub-entity representations between multiple entity containers.
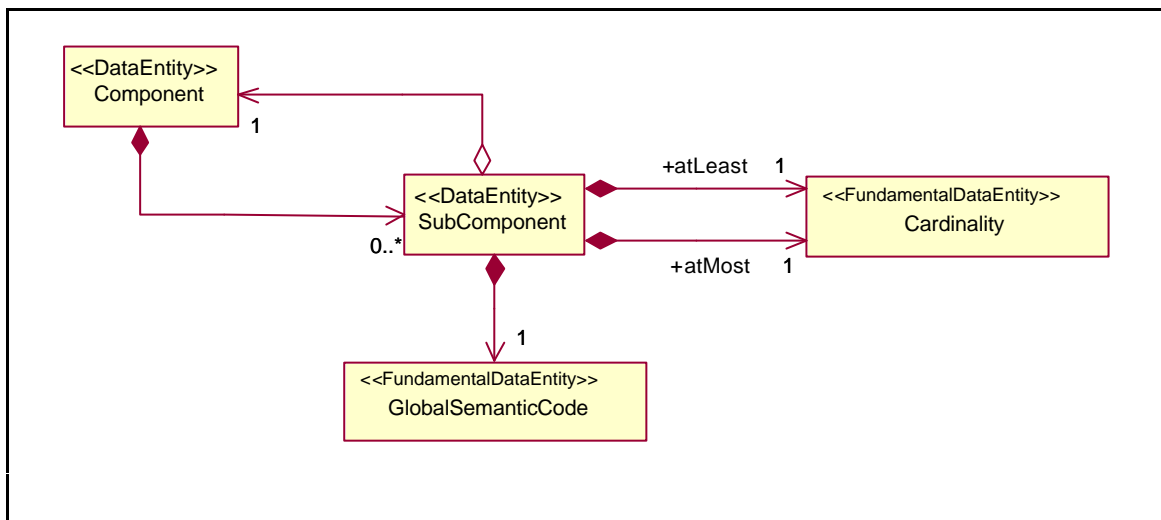
Figure 9-59   A Reference Relationship between Entities

Figure 9-59 shows a *Component* entity containing zero or more *SubComponent* entities that contain a reference to the same *Component* entity. Entities cannot be self-referencing via a UML association directly i.e. the client and supplier of a UML association cannot be the same. The UML association between the *SubComponent* and *Composite* entities must be unidirectional.
Figure 9-60 illustrates the use of parenthesis in a message guideline document to specify a reference from one entity to another. The supplier of the UML association is enclosed in parenthesis.

```
                ┌
            ┌───  Component
  0..*      │         ┌───  SubComponent
  1         │         │         ┌───────  (Component)
  1         │         │         ├───────  atLeast.Cardinality
  1         │         │         ├───────  atMost.Cardinality
  1         │         │         └───────  GlobalSemanticCode
```
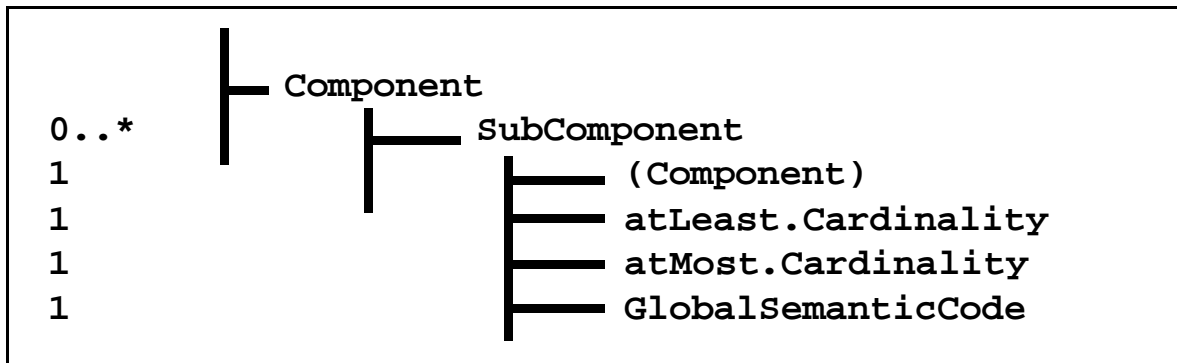
Figure 9-60 Illustration Showing Referenced Entity in Parenthesis

9.5.2 Query/Response Business Document Design Pattern

The query/response design pattern is useful for both querying business information and for specifying the structure of the response to the query. There are a number of approaches to designing query/response business documents.

1. The query and response are modeled as individual documents with fixed, independent structure.
2. The query is modeled as a constraint on a fixed structure that is used to return the response.
3. The query can be modeled as a constrained 'template' that must be 'completed' by a responding business partner.

The first approach is typical of Electronic Data Interchange (EDI) query/response message specifications. The second approach is typical of Structured Query Language (SQL) message specifications and the third approach is typical of symbolic programming languages such as Lisp or Prolog that implement unification. The UMM provides a design pattern for the third approach to query/response messages, as it is the most flexible approach to query/response message design where the query and response messages permit unlimited canonical data structures. The first two do not require a design pattern, as they are no different from standard business document specifications and are thus do not need a pattern.

Figure 9-61 illustrates a query/response data entity model. A product information query comprises zero or more query constraints and one product description. A product information response comprises zero (no results in query) or more product descriptions that match the query. A query constraint is an Object Constraint Language (OCL) expression that constraints the results returned in the query.

Specifying a template for the query results and placing constraints on the template by either filling in some of the template content or by constraining the content of the template using query constraints produces a product information query. Filling in the template in accordance with the already specified content and the constraints produces a product information response.
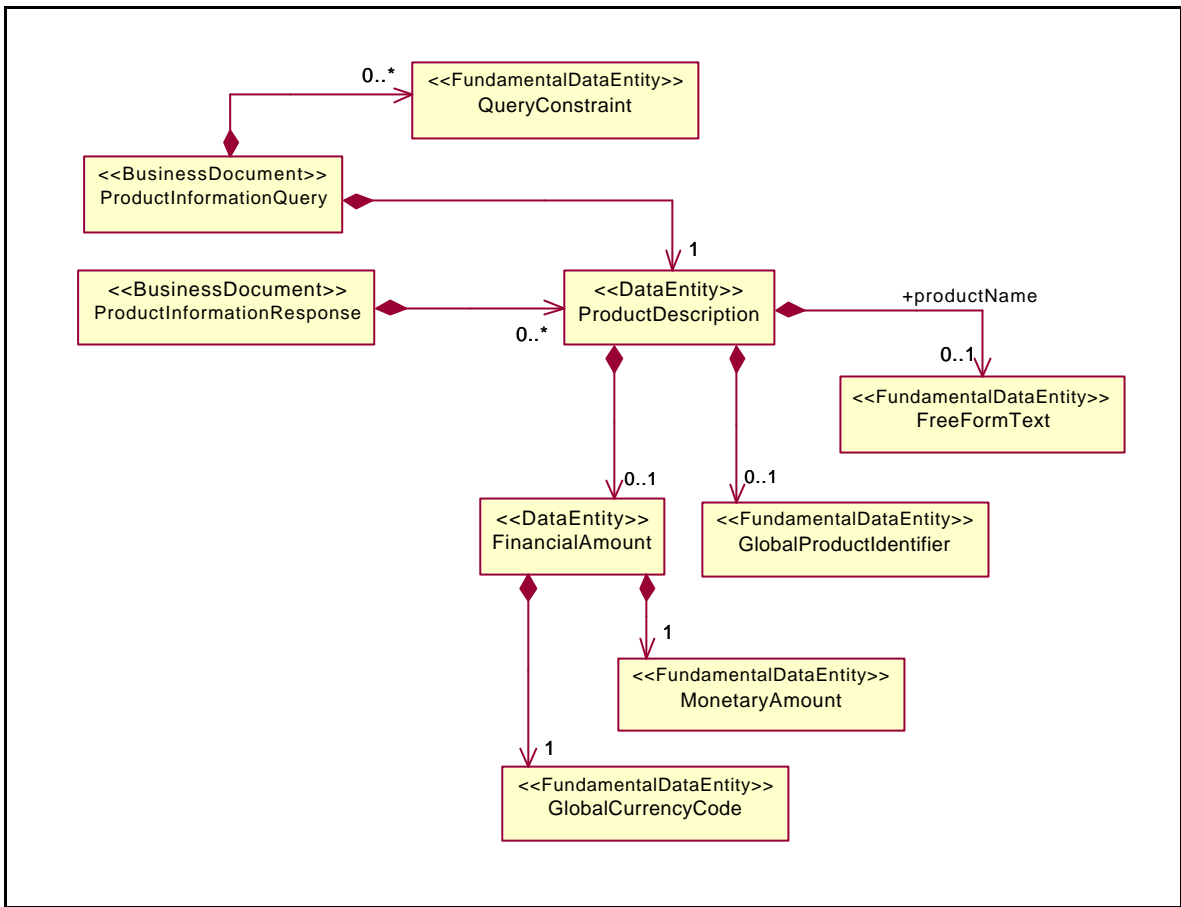
Figure 9-61 Query/Response Data Entity Model

9.5.3 Disjunction Design Pattern

The disjunction design pattern is useful for representing business information entities that contain one or more of a number of disjunctive entities (the pattern is also useful to inherit common data properties). This pattern is not necessary for representations of zero or more of a number of disjunctive entities. Figure 9-62 illustrates a model that employs a disjunctive design pattern.
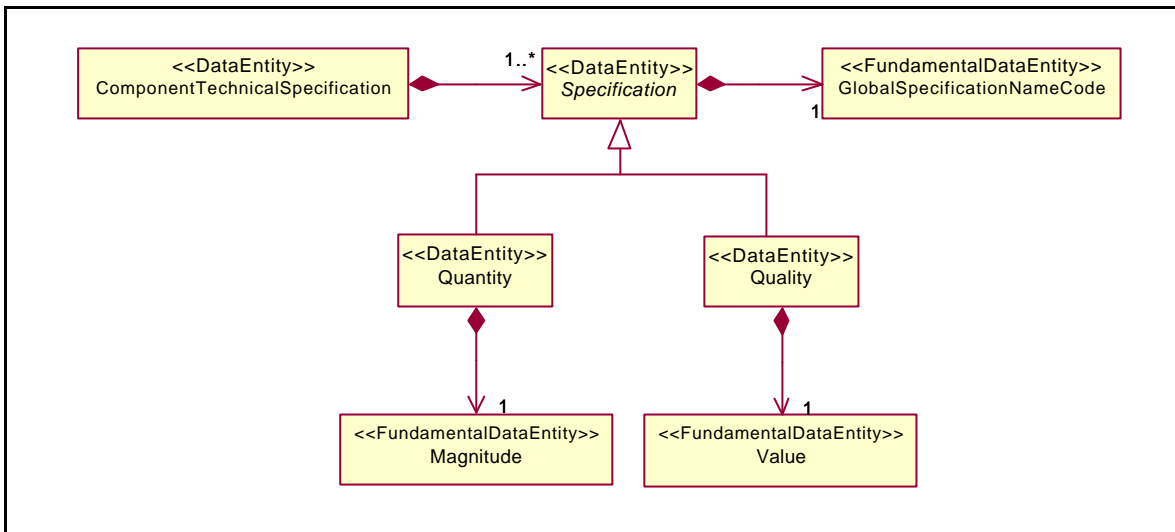
Figure 9-62 Disjunctive Data Entity Model

A component technical specification contains one or more specifications that are either quantities or qualities. Other representations of this specification allow either zero or more or two or more specification properties; now of which are meet the requirements of one or more specifications. Note that the specification data entity in Figure 9-62 is abstract (italicized class name). This prevents the data entity from being used as an object.

Figure 9-63 illustrates how the representation is shown in a message guideline document. The *Choice* node in the hierarchy shows the cardinality of one or more and the choice (disjunctive) nodes do not show any cardinality. The inherited *GlobalSpecificationNameCode* is repeated for each concrete class in the data entity model.

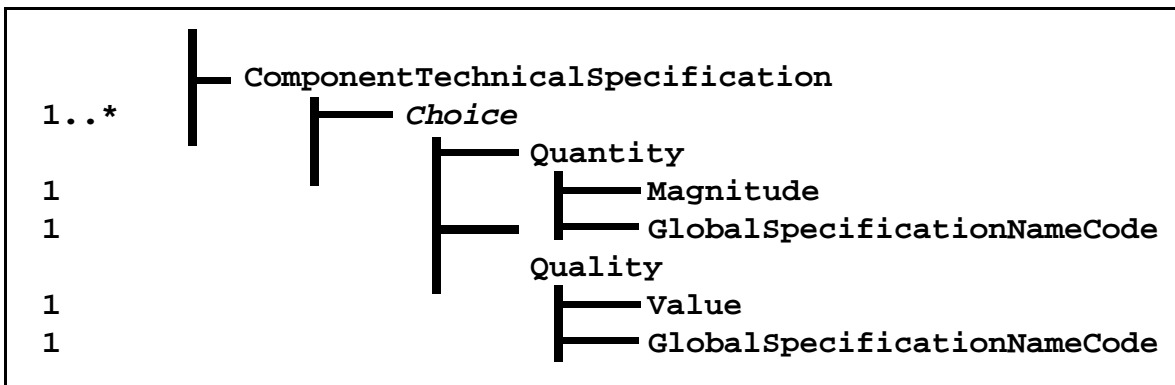**Figure 9-63 Disjunction Illustrated in a Message Guideline**

9.5.4 Reification Design Pattern

The reification design pattern is useful for representing common business information entities that share a common design pattern but are verbose in their representation. Figure 9-64 illustrates an entity model for representing a

1   manufacturer name and a product name.
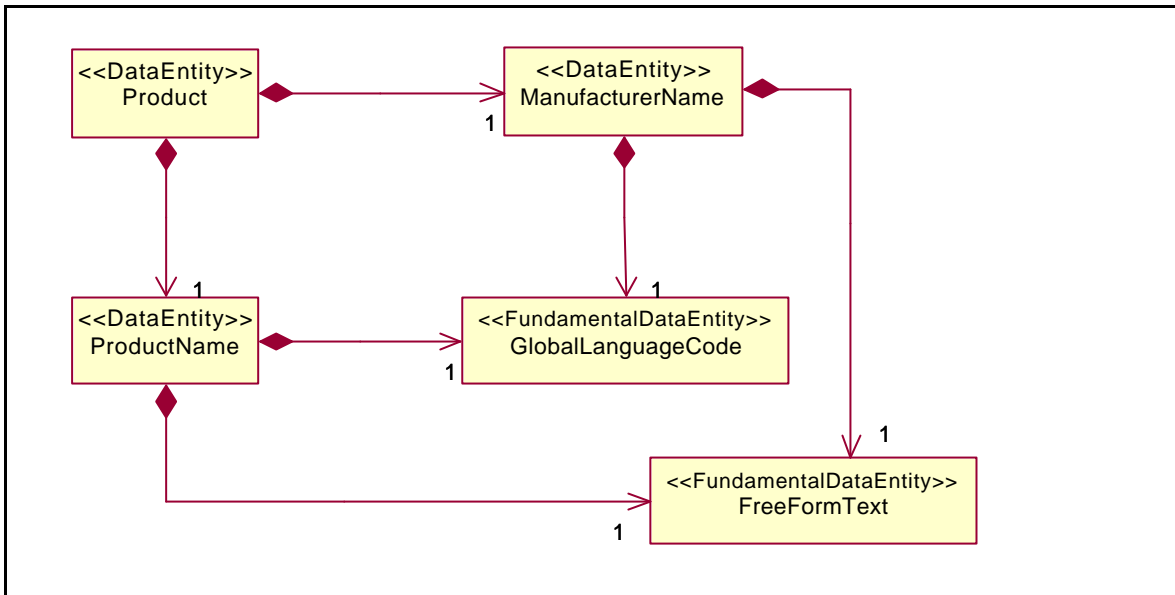
2

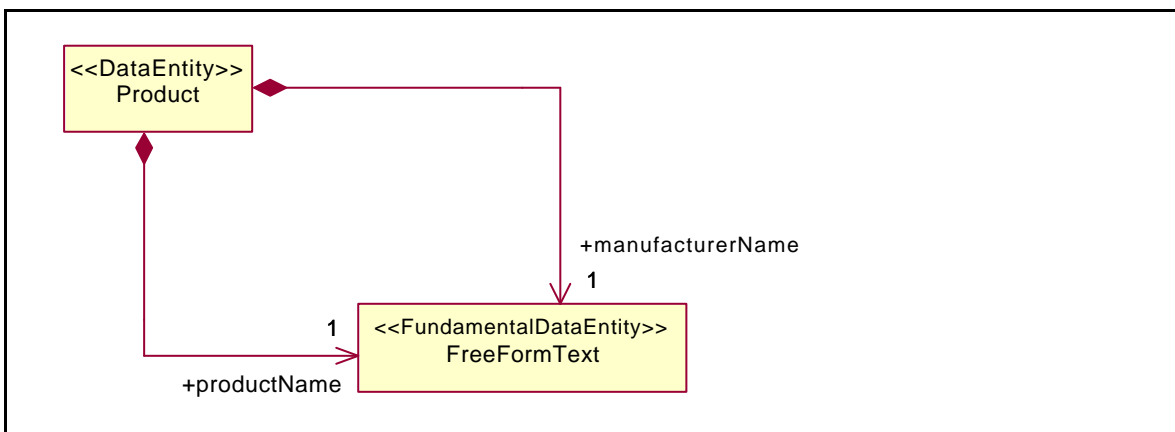3   **Figure 9-64 Illustration of a Free Form Text Entity**

4   Each 'name' entity contains a free form text entity and a global language code.
5   It is very verbose to specify these entities and relationships for each 'name'
6   entity in a large entity model. Figure 9-65 illustrates how the *ManufacturerName*
7   and the *ProductName* entities can be reified to property names if a design
8   pattern always emits a global language code requirement for each free form text
9   requirement.

10

11   **Figure 9-65 Illustration of Reified Data Entities**

12   The UMM uses this design pattern to reify the language code for free form text
13   and the physical unit of measure code for each quantitative data entity.
14
15   ***9.5.5 UML Translation Design Pattern***
16
17   The design pattern for translating from the UML specification to the specific TBL
18   (Tagged Value Length, such as XML) syntax document is expected.

1 Figure 9-66illustrates an example data entity model where a *Document* entity
2 comprises a *fromBusiness* and *toBusiness* declaration and a *Business*
3 comprises zero or one *Address* entity.

<<DataEntity>>
Document

+toBusiness   1

<<DataEntity>>
BusinessDescription

<<DataEntity>>
Address

+fromBusiness   1

0..1

4
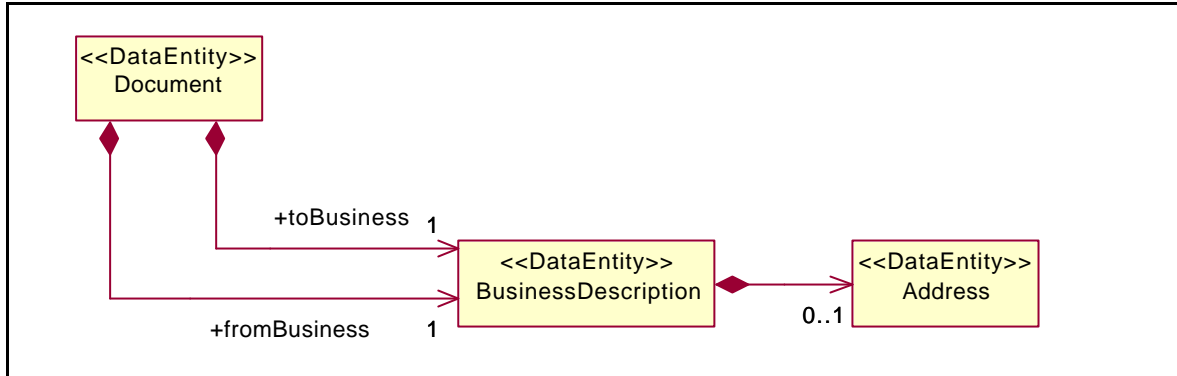
Figure 9-66 Illustration of a Data Entity Model

6 The UML model in Figure 9-66 is a 'network' model in that nodes in the network
7 are interrelated in a network of associations. A message guideline, however, is
8 a canonical hierarchy where each node in unique even though it is prototyped
9 on a node in the UML network model. The algorithm to convert the network to a
10 canonical hierarchy produces a graph shown in Figure 9-67 where each node in
11 the graph is dependant on a prototypical node in the network.
12 The graph is a guideline that is modified to accurately represent the business
13 data requirements. For example, Figure 9-67 illustrates that the *toBusiness*
14 declaration of a *BusinessDescription* is not required to contain an *Address* (it
15 needs to contain at least one Fundamental Data Entity but for the purposes of
16 this illustration it is not necessary to show this). The *fromBusiness* declaration
17 of a *BusinessDescription* is, however, required to contain an *Address*.
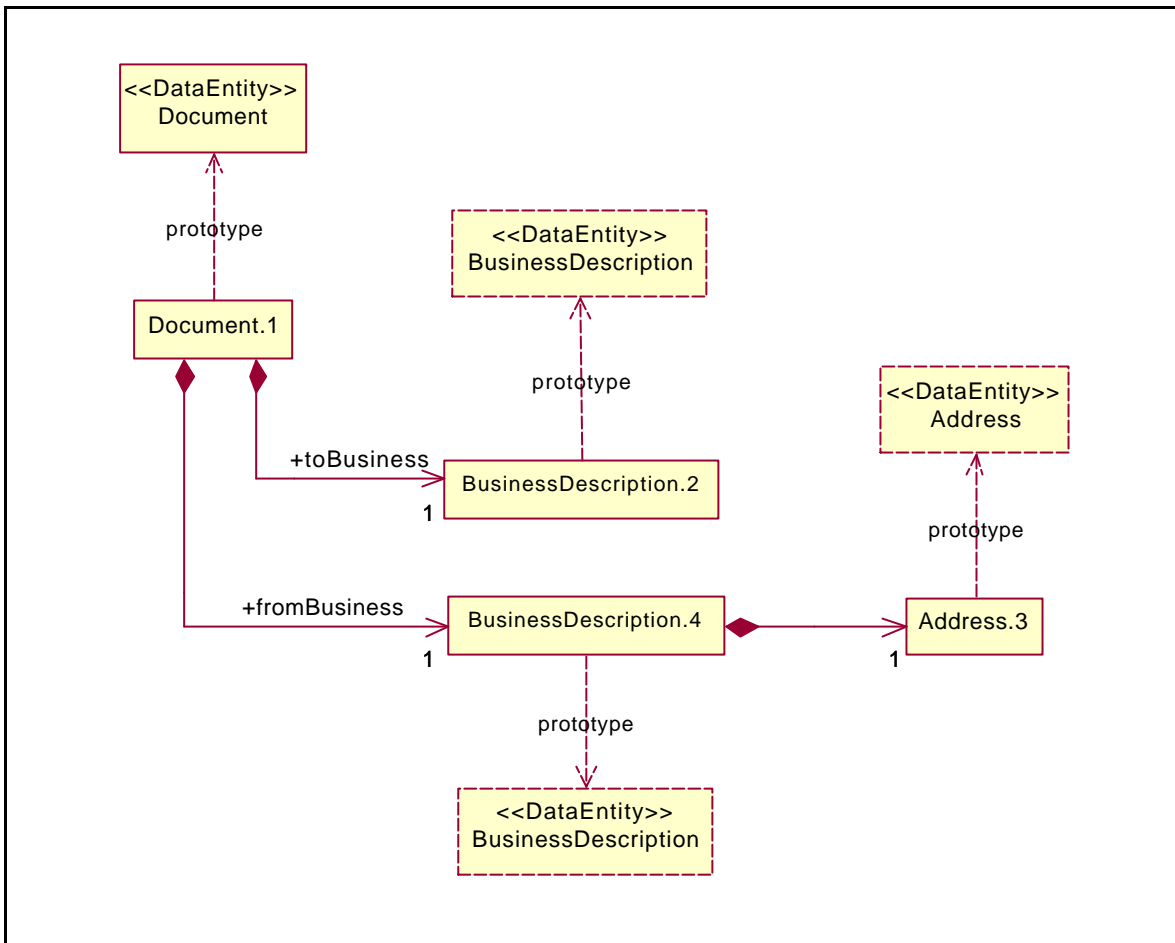
Figure 9-67 Illustration of a Canonical Hierarchy

9.5.6 Business Document Design Pattern

The following information is required in all business documents. Each business document must contain information that identifies the role, partner and business that is sending the business document. Each business document must also contain information that identifies the role, partner and business description that the document is going to. This information is similar to the information contained in the letterhead of a business document. Only the business identifier needs to be in the document as the identifier is the electronic equivalent of an address. Figure 9-68 illustrates the role descriptions in a business document.

```
1          From Role. Partner Role Description
1                    |-- Global Partner Role Classification Code
1                    |-- Partner Description
1                    |         |-- Global Partner Classification Code
1                    |         |-- Business Description
1                    |         |        |-- Global Business Identifier
1          To Role. Partner Role Description
1                    |-- Global Partner Role Classification Code
1                    |-- Partner Description
1                    |         |-- Global Partner Classification Code
1                    |         |-- Business Description
1                    |         |        |-- Global Business Identifier
```

Figure 9-68 Role Specification in a Business Document

The contact information of the initiating role must be included into the business document. The responding partner will be obligated to contact the initiating partner if there are errors in the received business document and a response (business signal or business document) cannot be delivered to the initiating partner, or there is no response specified. Figure 9-69 illustrates the contact information in a business document.

```
1          From Role. Partner Role Description
1                    |-- Contact Information
1                    |         |-- Email Address
1                    |         |-- Telephone Number. Communications Number
1                    |         |-- Contact Name. Free Form Text
```

Figure 9-69 Contact Information in a Business Document

The partner type, role type and supply chain code must be included as most conditional composition constraints are predicated on this information. Figure 9-70 illustrates supply chain specification in a business document.

```
1          From Role. Partner Role Description
1                   |-- Global Partner Role Classification Code
1                   |-- Partner Description
1                   |         |-- Business Description
1                   |         |        |-- Global Supply Chain Code
1          To Role. Partner Role Description
1                   |-- Global Partner Role Classification Code
1                   |-- Partner Description
1                   |         |-- Global Partner Classification Code
1                   |         |-- Business Description
1                   |         |        |-- Global Supply Chain Code
```

Figure 9-70 Supply Chain Specification in a Supply Chain

Each document has an identifier. Each responding document must include the identifier of a requesting document. This allows documents to be tracked and reconciled. Figure 9-71 illustrates the specification of a document identifier in a business document.

```
1          This Document Identifier. Proprietary Document Identifier
1                   |-- Administered By. Business Description
1                   |-- Document Identifier. Free Form Text
0..1   Requesting Document Identifier. Proprietary Document Identifier
1                   |-- Administered By. Business Description
1                   |-- Document Identifier. Free Form Text
```

Figure 9-71 Document Identifier in a Business Document

Each document must have a time and date stamp for auditing control. The date and time stamp is also used for legal purposes. Figure 9-72 illustrates the specification of a data and time stamp in a business document.

```
1          Document Generation Date Time. Date Time Stamp
1                   |-- Time Stamp
1                   |-- Date Stamp
```

Figure 9-72 Data and Time Stamp in a Business Document

9.5.7 Request/Response Business Document Design Pattern

The request/response design pattern is useful for requesting a business partner to perform a business action and to return a response that meets given constraints. This design pattern differs from the query/response design pattern in two respects:
1. Semantically, a query/response transaction specifies an initiator's request

for information that the responder has. A request/response transaction, however, asks the responder to perform an action and return a result of the action. This is an algorithmic response base on a prescriptive request.

2. Syntactically, a "Request" business document design pattern can comprise business rules that apply to the aggregation of the results in a response. Business applications responding to a request need to perform an additional processing step to apply these business rules to *all* the results of a query and not to *each* result of a query.

Figure 9-73 illustrates a request/response data entity model. A product availability request comprises zero or more query constraints, one or more business constraints and zero ore more product descriptions. The query constraints are constraints that must be met by each result returned in the response. The business constraints are the constraints that must be met by the entire response. Consider, for example, an initiator's product availability request for a maximum of 100 products of a particular type. The request for 100 products is a business constraint as the sum of all the product availability results must not be greater than 100. The type of product is a query constraint as each result must be the availability for the particular product type. A responding business partner may have less than 100 products and the partner may have more than 100 products in each of a number of locations. They therefore are required to perform a business action that reasons about how they will respond to such a request for availability. This may require some planning or optimisation algorithm to provide the response.

A product availability response comprises zero or more product availability results that match both the query constraint and the business constraint. A query constraint is an Object Constraint Language (OCL) expression that constraints each result returned in the response. A business constraint is an Object Constraint Language (OCL) expression that constraints the response.

Specifying a template for the response results and placing constraints on the template by either filling in some of the template content or by constraining the content of the template using query constraints produces a product availability query. Filling in the template in accordance with the already specified content, the query constraints and the business constraints produces a product availability response.

1
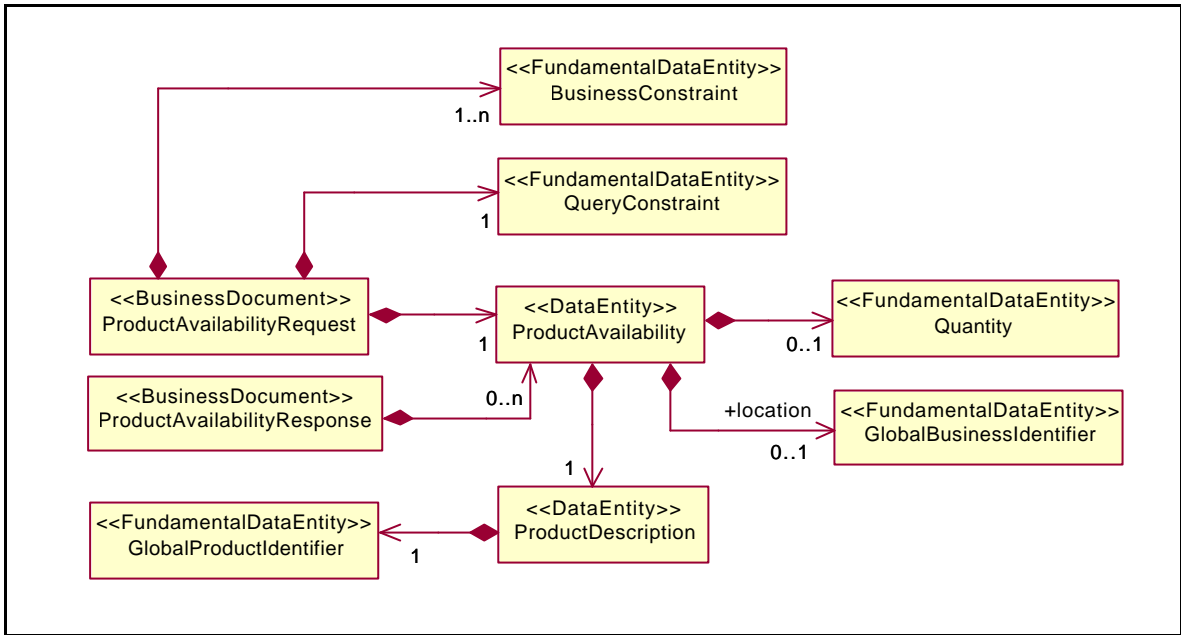
Figure 9-73 Request/Response Data Entity Model

3
4
5
6